MODELLING EMERGENT BEHAVIOUR OF ORGANISATIONS

TIME-AWARE, UML AND AGENT BASED APPROACH

RAUL SAVIMAA

2005

Faculty of Information Technology
Department of Computer Control
TALLINN UNIVERSITY OF TECNOLOGY

Dissertation is accepted for the commencement of the degree of Doctor of Philosophy in Engineering

Supervisor: Prof Leo Mõtus, D. Sc., Academician,
Faculty of Information Technology

Opponents:

1. Univ.-Prof. Dr. Dr. h.c. mult. Wilfried Brauer, Institut für Informatik (I7), Technische Universität München
2. Merik Meriste, cand tech, Institute of Technology, Tartu University

Commencement: August 30, 2005

Declaration: Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for any degree or examination.

Raul Savimaa

ISSN
ISBN

ILMNEVA KÄITUMISE MODELLEERIMINE ORGANISATSIOONIDES

AJATUNDLIK, UML JA AGENDIPÕHINE LÄHENEMINE

RAUL SAVIMAA

2005

Infotehnoloogia teaduskond
Automaatikainstituut
TALLINNA TEHNIKAÜLIKOOL

Väitekiri on vastu võetud tehnikateaduste doktori kraadi kaitsmiseks

Juhendaja: Prof., akad. Leo Mõtus, Infotehnoloogia teaduskond

Oponendid:

1. Univ.-Prof. Dr. Dr. h.c. mult. Wilfried Brauer, Institut für Informatik (I7), Technische Universität München
2. Tehnikakandidaat Merik Meriste, Tehnoloogiainstituut, Tartu Ülikool

Kaitsmine: 30. august 2005

Deklaratsioon: Deklareerin, et käesolev doktoritöö, mis on minu iseseisva töö tulemus, on esitatud Tallinna Tehnikaülikooli doktorikraadi taotlemiseks ja selle alusel ei ole varem taotletud akadeemilist kraadi.

Raul Savimaa

ISSN
ISBN

ABSTRACT

The thesis focuses on the modelling of emergent time-sensitive operational behaviour in multi-functional human organisations. The research concentrates on organisations that operate in a dynamically changing environment where decisions should be made in minutes or seconds and the impact of the organisation's activities is time-dependent (e.g. dynamic teams, organisations ensuring public safety like the police and rescue service and several other organisations).

An organisation, as seen in the thesis, comprises a set of autonomously functioning and proactively collaborating actors that interact by exchanging time-sensitive messages. A message may represent information and/or materials. The organisation has goal functions that are used for planning and assessing the efficiency of the concerted actions of departments. The functioning of an organisation is modified and/or tuned by adjusting goal functions of departments. Each actor has its own priorities and preferences that may not be fully compliant with organisational goals. Actors have significant responsibility and the decisions of single actors have great influence on organisational performance.

Currently, as will be shown in the thesis, the existing solutions (that are often concentrated on the modelling of existing or planned processes and workflows only) are not sufficient for modelling emergent behaviour. A suitable approach should combine time-aware modelling of processes with the simulation of actors' priorities, choices and behaviour.

As one possible solution, the thesis introduces a novel modelling methodology for studying emergent behaviour in organisations and for evaluating and assessing the influence of planned modifications of the organisational structure and the processes prior to their implementation. It is based on UML approach that enables to implement theoretical results related to the interaction-centred model of computation. UML use case, activity and sequence diagrams are used for describing processes. Other essential features of an organisation are encaptulated by different UML profiles (e.g. universal UML, RT UML and Agent UML). For analysing the timing correctness of interactions the Q-model (a candidate for the UML model processor) is used. Subjective human factors within a department and inter-departmental interactions are described and analysed as behaviour of a multi-agent system. The implementation of the methodology is illustrated by three case studies.

The results of the research are described in four different articles – three in the framework of international conferences (Savimaa, 2002a; Savimaa, 2004a; and Savimaa, 2004b) and one in scientific journal (Savimaa, 2005).

Key words: modelling of organisations, emergent behaviour, model design, the UML, agent technology, multi-agent systems, the Q-model, modelling of modifications.

KOKKUVÕTE

Käesolev doktoritöö käsitleb ilmneva käitumise (*emergent behaviour*) modelleerimist multifunktsionaalsetes organisatsioonides. Töö keskendub dünaamiliselt muutuvas keskkonnas toimivatele organisatsioonidele, millised peavad reageerima minutite ja sekundite jooksul (näiteks dünaamilised meeskonnad, päästeteenistused ja politsei). Kuigi töötajate käitumise täpne ennustamine ei ole kuidagi võimalik, on piisava detailsusega prognoosimine kasulik olemasolevate protsesside efektiivsuse hindamiseks ja muudatuste kavandamise toetamiseks. Usaldusväärne prognoosimine ning muudatus-ettepanekute kvaliteetne koostamine on võimalik vaid sobivat (pool)formaalset lähenemist kasutades.

Organisatsiooni on käesolevas töös käsitletud autonoomselt funktsioneerivate ja proaktiivselt koostööd tegevate tegijate (*actors*) struktureeritud kogumina ja nendevahelise keskkonnana. Tegijad suhtlevad omavahel ajatundlikke sõnumeid vahetades. Sellised tegijad moodustavad organisatsioonis struktuuriüksusi (näiteks osakondi). Üksusi suunatakse organisatsiooni sihifunktsioonidest tulenevate ülesannete ja hindamis-kriteeriumite kindlaksmääramisega, mis üldjoontes peaks määrama iga töötaja/tegija tegevused. Organisatsioonis on kehtestatud mittetäielik kogum norme (näiteks seadused, sisemised direktiivid, juhendid, käitumiskoodeks), mis loovad lubatud tegevuste ning ülesannete ja probleemide lahendamise ruumi ja viisid. Samas ei pruugi tegijate isiklikud eelistused ja prioriteedid organisatsiooni omadega kokku langeda, mis mõjutab organisatsiooni tegelikku käitumist tuntavalt, kuna organisatsioonis tegelikult ilmnev käitumine on tegijate (töötajate) erinevatest mõjuallikatest ja põhjustest kujundatud tegevuste kogum.

Doktoritöös on näidatud, et kasutatavad organisatsiooni protsesside mudelid ei ole piisavad ilmneva käitumise esitamiseks ja vajalik on täiendavalt arvestada tegevuste ajakitsendusi ning tegijate endi võimalikke prioriteete ja eelistusi. Töös on analüüsitud planeeritud ja ilmneva käitumise modelleerimisele esitatavaid nõudeid ning antud sobivate lähenemiste lühiülevaade.

Ühe sobiva lahendusvariandi esitamiseks on doktoritöö raames välja töötatud organisatsiooni mudel ning esitatud uudne metoodika sellise mudeli koostamiseks. Uurimuse konkreetse tulemusena on näidatud, et organisatsioonis ilmneva käitumise esitamiseks on vajalik protsesside ajatundlik modelleerimine ühitada tegijate käitumise modelleerimisega.

Töös soovitatud modelleerimismetoodika baseerub UML-il (*Unified Modelling Language*), mis võimaldab rakendada saadavad teoreetilised tulemused vastastikusele suhtlusele (interaktsioonidele) orienteeritud (*interaction-centered*) arvutusmudelitel. Protsessid on kirjeldatud UML-i kasutusjuhtude, tegevus- ja jadadiagrammidega. Organisatsiooni spetsiifilised omadused on hõlmatud UML-i erinevates profiilides (näiteks tava-UML, reaalaja-UML, agent-UML). Interaktsioonide ajalise korrektsuse analüüsiks on

kasutatud UML-i mudeliprotsessori kandidaati Q-mudelit ja selle rakendust LIMITS. Üksusesisesed subjektiivsed inimfaktorid ja üksustevahelised (näiteks osakondade vahelised) interaktsioonid on kirjeldatud ja analüüsitud multiagentsüsteemi käitumisena. Multiagent-lähenemine ja Q-mudel toetavad süsteemide visuaalset simuleerimist ning võimaldavad peita rakendatavat formalismi lõppkasutaja eest. Pakutav metoodika toetab protsessides kavandatavate muudatuste eeldatava mõju hindamist enne nende tegelikku rakendamist organisatsioonis.

Esitatud metoodikat on rakendatud tegelike tööprotsesside analüüsil ja ettepanekute tegemiseks. Metoodika rakendamise kohta on doktoritöös esitatud kolm näidet. Uurimuse tulemused on ette kantud ja avaldatud erinevates artiklites kolmel rahvusvahelisel konverentsil (Savimaa, 2002a, Savimaa 2004a ja Savimaa 2004b) ning ühes artiklis erialases teadusajakirjas (Savimaa, 2005).

Võtmesõnad: organisatsioonide modelleerimine, ilmnev käitumine, mudeli disain, UML, agenttehnoloogia, multi-agent süsteem, Q-mudel, muudatuste modelleerimine.

CONTENTS

1 INTRODUCTION

Modelling is a well-known activity of engineering. The system and its actual behaviour are often too complex to understand and therefore models are used to encapsulate the essential characteristics of an actual system and to allow simulating and analysing the behaviour of the system. The same is also true for the modelling of human organisations.

A human organisation is a collection of actors that are performing together to reach a common goal[1]. But often the outcome does not satisfy the desires of stakeholders. This can be caused by impropriate tasks, insufficient skills or inadequate actions of employees. Employees in a human organisation are autonomous subjects with free will, beliefs, desires, intentions and motivations. Therefore actually emerging behaviour of an organisation, as a result of the behaviours of its actors, depends on their current preferences and choices.

To predict possible behaviour of an organisation in different emerging circumstances, one needs suitable models that describe processes, actors and their interactions and to some extent also grounds for behaviour. A model that allows us to analyse various operational situations and to evaluate the results of such decisions is extremely useful: a tool enabling to forecast the effects of modifications would maximise the benefit of organisational performance with minimal re-design and at the same time would reduce the potential harmful side effects of organisational modifications. Such a tool would be even more important on grounds that restructuring is carried out "on-line", i.e. without suspending everyday activities.

There is no general solution for modelling emergent behaviour – the actually emerging behaviour is always caused by a number of different aspects and an observer does not have enough information for having a complete behavioural model. There is especially lack of applicable methodologies, suitable for modelling the emergent behaviour of multi-functional human organisations that operate in time-critical dynamic environment and have to react, on an operational level, in minutes or seconds (for example rescue services, the police and military organisations).

Modelling of expected emergent behaviour in those organisations enables to simulate different operational situations, in this way helping to improve the performance of actors. The topic is important since wrong operational and strategic decisions can have a serious direct impact for the organisation and its clients in the described application domain. On the other hand, well-specified response criteria, soft real-time characteristics and the explicit outcome of decisions enable exact and resultant modelling.

The present thesis focuses on the modelling of emergent time-sensitive operational behaviour and aims to support process modifications (change

[1] Definitions of terms are given in section 1.1

management). The suggested solution can later be generalised to some extent in order to model emergent behaviour also in human multi-functional organisations in other domains.

1.1 GENERAL BACKGROUND

This section describes the background of the current research – it reviews the characteristics of human emergent organisations, gives definitions of terms and notions and reviews goals for organisation modelling.

1.1.1 Terms and notions

A *human organisation*, as seen in the thesis, comprises a set of autonomously functioning and proactively collaborating human and artificial *actors* that interact by exchanging time-sensitive messages (a message may represent information and/or materials). An *actor* denotes a human person or a piece of software, data or equipment that is seen as an entity. From the modelling perspective, an actor possesses *attributes* that characterise the actor and a set of *methods* – description of its actions, given in some notation. Employees, database engines and information systems are examples of actors in the organisation.

Human actors are organised into *structural units* of the organisation – formal entities that have a common goal and a grouping purpose (e.g. departments or divisions). An organisation is usually hierarchical – its structural units may also consist of *sub-units* (e.g. sections in a department). Also a sub-unit may consist of smaller sub-units until at the elementary level the organisation consists of actors. *Organisational structure* is a description of structural units of the organisation together with their placement in hierarchy and inter-connections within the organisation (e.g. similarly to the definition of the structure of the system; MGHD,1984). The term *architecture of an organisation* is often used in the same meaning. *Hierarchy* is seen as a way of classifying structural units of an organisation into superiority levels. The hierarchy of structural units in an organisation is illustrated in Figure 1.1 on the left.

An organisation is inseparably situated in its *environment*. In the present thesis *Environment* is considered as the activity sphere of actors. It is used in two meanings:

1. *Environment of actors* is an integrated system of objects and artificial actors (pieces of software, data or equipment) inside the organisation – it comprises the information system, communication channels, organisational hierarchy and structure. The task for actors' environment is to enable actors' activities (e.g. interaction). Actors are situated in

their environment – in a real human organisation – and must consider all the constraints that the environment implements.
2. *Environment of an organisation* is a set of actors outside the organisation, e.g. clients (customers of services provided by an organisation), partner organisations (friends, competitors or enemies), the society. This is similar to the definition of actors' environment if to consider an organisation as a single actor. Dynamic environment means that tasks, requests from clients and applicable activities often change and the organisation must react in time.

The term *activity domain* denotes the operation area (or field) of the organisation. Outside observers often determine the domain as a result of the categorisation of the environment. The activity domains can be, for example commerce, banking, military affairs, law enforcement, education, etc. Within the activity domain several *activity areas* can be defined where the organisation performs its activities. Examples of the activity areas within domains include e-commerce, e-broking, e-based learning. An organisation may divide its activity area into smaller entities, for example *activity lines*. *Work processes* (often also called *business processes*) are series of activities of different actors that will be performed in order to reach goals in some activity line or in some activity area. The work processes may be hierarchical, consisting in their turn also of work processes. The smallest entities in this hierarchy are *elementary work processes* – work processes that are not further refined in the process hierarchy. The terms given in this paragraph are illustrated in Figure 1.1 on the right.

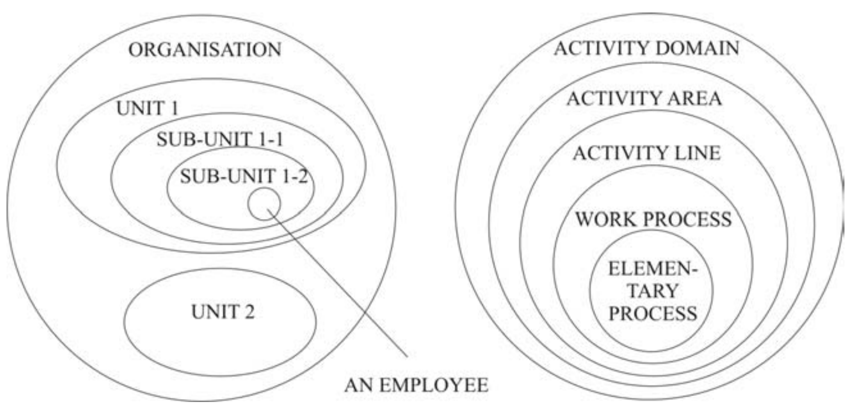

Figure 1.1 Relations between the terms used for structural units (left) and for the description of activity domain (right)

Actions are actor's activities. Actors execute processes by performing actions. In this way processes consist of the same activities, only seen from the perspectives of tasks and processes. Series of actions are seen as series of elementary processes that form together a composite process.

Multi-functionality of an organisation means that there are multiple goals to achieve (often in different activity areas) and tasks to solve at the same time. This means that in reality a number of functions and processes may require the same common *resources* (e.g. actors, equipment, knowledge, financial assets) at the same time and the same persons or resources may be fully or partially loaded by other functions (using time-sharing for example). As a rule, the existing resources in an organisation are not mapped directly to specified tasks and do not often correspond to the needs.

The *goals of the organisation* form the *goal function* – a pre-defined prioritised list of requirements and assessment criteria for the performance of the organisation. The goals usually describe the purpose and constraints of the organisation together with some long-term and short-term objectives. The overall goal may be given in very general terms. Planning of the organisational activities is essentially related to the decomposition of the goal into sub-goals. For actual implementation, the sub-goals are assigned to tasks and sub-tasks and attached to structural units of the organisation. *Task allocation*, i.e. assigning the organisation's activity domain to structural units of the organisation, can be done in many ways, depending on assessment criteria. Therefore no fixed suggestions exist for mapping goals, organisation structure and processes (Scott 1992). At the same time, proper correspondence of the goals of structural units as well as personal goals and motivations of the employees to the overall organisation's goal is a key component for the efficiency of the organisation.

Decision-making in an organisation may vary from a fully centralised process to a fully decentralised process. The possibility of lower-level units to participate in the decision-making in strategic, tactical, or operative issues, both on a local and global level, plays an important role in the performance. *Delegation* mechanisms can be different and, in principle, depend on the organisational structure. In practice delegation is usually organised according to the subjective visions of managers and it depends on the subjective assessment of efficiency and trustworthiness of units and actors.

Co-operation – mutual co-ordination of activities of different actors in order to achieve a common goal – determines the performance of the organisation. Good co-operation compensates the existing gaps in the normative base and plans. Actual co-operation does not necessarily correspond to planned (prescribed) co-operation; in fact it is often much more complex, depending on both formal and informal relationships between employees. As collective behaviour formal norms and informal relations determine also the behaviour detectable between structural units as multi-agent systems.

Human organisations (e.g. authorities and companies) are designed for the fulfilment of certain goals and tasks in the environment (the society). Applicable behaviour is often standardised, the suitability of each action depends upon a particular situation. The goal will be reached (or the task will be fulfilled) when certain pre-described work routines have been completed. In addition, the behaviour of an organisation should meet the requirements elaborated by the environment (the society). Such requirements (that can be named as *activity boundaries*) can be categorised into three groups, as illustrated in Figure 1.2: juridical frames for the organisation's activity, goal functions and intra-organisation documents. Arrows in the figure illustrate the influence between different categories of rules, documents and actions.

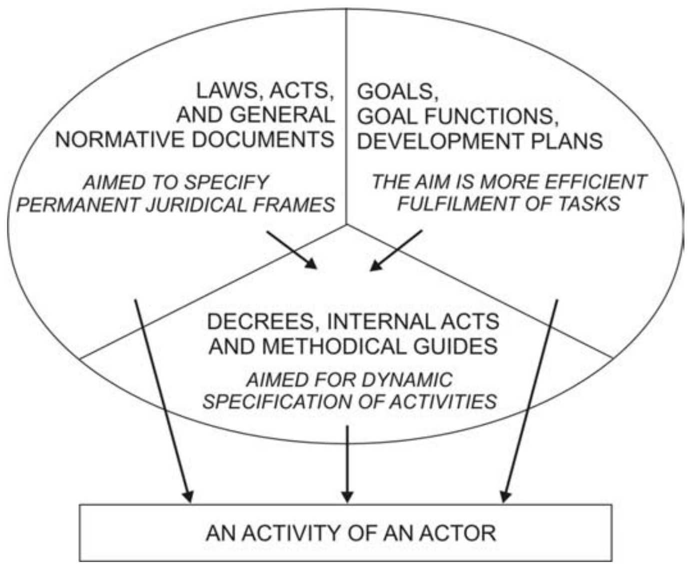

Figure 1.2 The illustration of different categories of the normative basis

A conventional human organisation relies on a set of documents (e.g. basic regulation, laws and acts, given to organisation from higher authorities) that describe its aim, responsibility and power and regulate the activity domain. Those documents can be considered as more or less static permanent juridical frames and it is quite difficult or sometimes even impossible for the organisation itself to make those rules to be changed.

The second category of documents is guiding documents (given form a higher authority or elaborated inside the organisation) as goals, goal functions and development plans for determining priorities and guiding how to fulfil tasks in the best way. Normally such documents are valid from three to five years for development plans or one year for annual goals, action plans and more precise work plans. For performance assessment, goal documents should also include measurable evaluation criteria for goals, tasks and subtasks of its structural units.

Finally, documents in the third group describe work organisation, methods and correct internal organisational procedures. Those documents are dynamically worked out inside the organisation. Process descriptions tend to be more generic than specific. The desired outcome of activities is difficult to represent explicitly in terms of detailed work processes. The expected work processes are are described in more detail in financial directives, guidelines for military and public service and for decision-making procedures.

Ideally, all those documents, falling into three categories of activity boundaries, should compose whole one non-contradictory entity that covers all work situations[2].

The planned behaviour is developed stepwise: *planned strategic behaviour* departs from the goals, general tasks and strategic plans of the organisation; *planned tactical behaviour* (as a more specific plan) refines the initially planned strategic behaviour into specific activities, taking also into consideration the goal functions of units, bonuses and specific features of the organisational structure and its information system.

In classical human organisations applicable behaviour is often also standardised. In a usual human organisation expected behaviour is usually described as a set of work processes that has to be carried out. The goal is believed to be reached (or the task will be fulfilled) when certain pre-described work routines have been completed. In multi-functional organisations the targeting of the overall goal means simultaneous achieving of multiple sub-goals. In most human organisations (like governmental organisations, foundations or non-emergency service companies) temporal criteria of processes are not crucial and are considered purely as deadlines for responding to environmental situations (e.g. incoming messages).

A structural unit in the organisation is controlled by fixing its tasks and goal functions (within the overall goal function for the organisation), which indirectly guides the activities of each employee in this unit. The organisation's structure is fixed, although dynamic groups are created for everyday operational tasks. Each actor has its own priorities that may not be fully compliant with the priorities of the organisation. The Functioning of an organisation is modified and/or tuned by selecting appropriate departments and adjusting their goal functions so as to satisfy the overall goal functions of the organisation.

[2] In reality they cover only most common situations and are often contradictory. One of the reasons is that they are completed at different times and each time different goals / aims are considered. In addition, in reality there are also a number of non-written rules which may gain even higher priority.

1.1.2 Emergent behaviour

Behaviour is considered as a series of actions performed by an actor. *Emergent behaviour*, as considered in the thesis, is an operational behaviour of an actor that emerges as a result of the actor's intentions, environmental situations and actor's reactions to situations. Emergent behaviour of an organisation is a co-influence (cumulative composition) of emergent behaviours of its human and artificial actors.

For modelling purposes *planned (expected) behaviour of an organisation* is described as an ordered set of planned *work processes*. *Reactive behaviour* is a response to received information or a signal from the environment (or from another actor). *Proactive behaviour* is a series of activities performed by an actor that are designated for the fulfilment of the actor's goals and tasks and are planned to precede the potential non-desired activities of other actors and environment. In proactive behaviour the actor takes the initiative to better fulfil its tasks and plans.

The behaviour of an organisation depends on the environment in three aspects:

1. The aim of the organisation and its operating environment actually determine whether the modelled organisation has long deadlines and a strong possibility to pre-determine the processes or emergent operational behaviour with high dynamics. Organisations, considered in this thesis, can usually influence the environment on a small scale but cannot influence large-scale changes or major tendencies in the environment.
2. The way how an organisation behaves (its "language of actions") is largely determined by the environment – an organisation must "speak" the language that is understandable for the environment the organisation is designed for.
3. *Actual operational behaviour* of an organisation – a composition of planned activities and reactions to emerging environmental situations.

Planned and actually emerging behaviour never coincide, some plans will be fulfilled and some plans will be fulfilled partially or never implemented.

The actually emergent behaviour of an actor is a result of multiple factors – it is based on planned strategic behaviour that is refined into planned tactical behaviour. The behaviour is modified according to emerging situations, goals and interactions, including informal communication. The intended behaviour is finally composed according to an actor's beliefs, desires, motivations, intentions, preferences, existing knowledge and the ability to acquire new knowledge.

Actual behaviour of an actor is often not fully compliant with the ideal behaviour for the role that the actor fulfils in the system. Also, the real environment is too complex and non-predictable in details – in a real organisation there are hundreds or thousands human and artificial actors that interact with each other in order to implement organisational goals and to react to environmental dynamics.

As a result, the behaviour of employees (especially temporal characteristics of that) is ongoing, caused by the behaviour of multiple independently interacting actors and is truly emerging. Not all possible relevant decisions related to emerging situations can be objectively prescribed. Often employees have great decision-making power within a strongly specified freedom domain (for example a manager of a rescue unit in the place of event, a police patrolling officer, an investigator, an immigration officer, etc.). Therefore the chosen behaviour and decisions of each related employee directly determine the result of the activities and influences to a great deal the overall efficiency of the organisation.

As in principle we cannot describe all possible behaviours in advance, suitable models can encapsulate only some sub-sets of (most probable) behaviours. Expected emergent behaviour can be represented as prognosis (in a form of behavioural patterns) and only to some extent, when taking into consideration as much as possible different aspects that influence the behaviour. Suitable models that will be used for describing emergent behaviour of the organisation and its actors must consider multiple actors and be also able to represent time issues of the represented processes and activities.

Emergent organisation is an organisation that constantly tries to adapt to its changing environment and meet the evolving requirements (Alatalo et al., 2002). In the current thesis the behaviour of emergent organisation is considered as emergent behaviour. The organisational structure is often fixed (not dynamically emerging) but can be easily adjusted according to modification plans, if such exist. A suitable way should be found how and how often to modify the structure of the organisation in order to cope with the evolution of goals and changes in the environment.

Dynamic nature of operating environment means that proper reaction (decisions and corresponding activities) is time-sensitive, depending on a specific situation and required feedback time. An example can be brought from the domain of policing: if information on a stolen vehicle in a particular town is received almost immediately by a police authority and forwarded more or less instantly to all police units, border points and vehicle registration offices, it may be sufficient for detecting the vehicle. If the first information in a similar situation arrives several hours later, the vehicle may already be taken abroad and in this case it is also necessary to inform international law enforcement information channels.

Examples of human emergent organisations include the police, rescue service, several other public service authorities or organisations as well as military groups. Those organisations have a number of actors, acting independently and exchanging often voluminous information. Voluminous information exchange influences also the behaviour of its actors.

Model of an organisation is a description of the structure, goals, formal and informal relationships, processes, actors and *behavioural schemas* (specification of a set of behaviours) of an organisation. A general description of a human organisation is usually given in a semiformal (or even informal) form; other components of the model (mainly process specifications and behavioural components) are more formalised. *Modelling* of an organisation is a process of devising a model of an organisation.

Meta-model (also called *reference model*) is a general platform-independent description (meta-description) of any family of similar platform-dependent models, used for the same purpose (e.g. Workflow Reference Model, Enterprise Model).

Soft real time systems are systems where temporal requirements are determined in seconds (or slower) and late data arrival or somewhat unexpected differences in process execution may disturb the overall system; but it still does not have a catastrophic (fatal) impact on the system or its components. On the contrary, in *hard real time* temporal requirements are very strict (often also defined in milliseconds) and inadequate performance may have catastrophic impacts (e.g. in aeronautics, chemistry or power industry).

The present thesis takes into consideration only operational behaviour and issues related to its modelling. In principle, also actual tactical and strategic behaviour in human organisations is emergent: modifications of work processes and methodical guides are both reactive (as a result of changing laws and norms) and proactive (as considerations and activities of actors related to the improvement of organisational efficiency or operational behaviour). Such emergent tactical and strategic behaviour is even more complex and difficult to model.

1.1.3 Goals for the modelling of organisations

In this section we will review what the usual goals are when modelling human organisations and compare them with our modelling goal. The similarity of goals facilitates the careful use of some of the same models and tools also for our task.

Modelling of work processes in different human organisations (e.g. enterprises, companies, or authorities) is widely used. One of the main reasons for modelling an organisation (especially its work processes) has traditionally been the design or modification of information systems that support everyday operations. Therefore organisational models in this area are developed most. Oftentimes an information system should not only support the information flow and execution of processes related to information processing, but should also assist correct decision-making by providing a list of applicable activities for each particular situation. Modelling is only a single step during the development of information processing system. In order to really support the information system development process, a model of the organisation should be suitable for being incorporated into this process.

Methodologies implemented for the development of information systems specify an organisation thorough its processes. For modelling purposes the work processes are described accordingly to documents and interviews with employees. Implemented models, information systems development methods (e.g. Unified Process, Arlow and Neustadt, 2002) and description languages (e.g. the Unified Modelling Language; Booch, Rumbaugh and Jacobson, 1999) often concentrate only on planned or detected process logic (algorithm) (for example dataflow diagrams, state charts), input-output characteristics and data exchange (i.e. interactions in and with the environment). For more detailed notations also extended mark-up is used (e.g. Coloured or Timed Petri Nets, Mortensen, 2003; Wang, 1998).

Often the Unified modelling Language (UML) is used for specifying processes and interactions. UML is a graphical language for visualizing, specifying, constructing and documenting the artefacts of a software-intensive system (Booch, Rumbaugh and Jacobson, 1999). It is widely used as standard in industry and academic domains helping to describe, visualize and analyse models of various systems. Object Management Group (OMG) adopted the first version of the UML in 1997 and OMG Revision Task Force (see OMG, 2005) releases its versions now (the latest version is 1.5, see OMG, 2003a).

Another goal for modelling an organisation (especially its behaviour) has been efficiency analysis and performance assessment. Contemporary views on modelling extend from the traditional analysis of characteristics of planned process to a complex analysis and simulations of different aspects of organisational behaviour (e.g. correspondence of goals and everyday activities and existence of co-ordination routines). Corresponding models enable to elaborate more sophisticated information systems that could also be useful for assisting during emerging circumstances outside pre-planned activities.

In real life management and efficiency issues are generally handled in qualitative terms. In normative descriptions those issues are not directly mapped into everyday activities in the organisation. The managing of an organisation or a unit in non-production domains is normally considered only from the viewpoints of planning, communication and psychology, is subjective and very person-centred.

Conventional analysis methods for the assessment of effectiveness in human organisations concentrate mainly on issues related to structure, work methods and document handling. Most of the tools and methods that are used for describing processes in human organisations, e.g. Gantt chart (Clark, 1922), Critical Path Method (MindTools, 2003a), SWOT (MindTools, 2003b), also Enterprise modelling methods (e.g. EKD, Bubenko, Persson and Stirna, 2001), concentrate on causal relations in the sequence of work processes and on the fulfilment of input and output conditions.

Organisations specify their work processes and use modelling and simulation in different detail. Depending on the content of work processes and the level of their definition (i.e. how well an organisation pre-plans its work processes, handles exceptional situations and evaluates its operational efficiency and quality) organisations can be classified into maturity classes – e.g. capability maturity model P-CMM in software industry (SEI, 2001). To ensure the quality and correct timing criteria of supportive processes, standardisation and normative documentation is needed. Similarly to quality standards ISO 9001 and 14001 (see ISO, 2003) for production environments, a number of quality programs are also elaborated in non-production domains for ensuring the quality of results by guaranteeing the quality of everyday activities, e.g. EFQM (EFQM, 2005), the Best Value (Boyne, 2000) and Balanced Scorecard (Kaplan and Norton, 1996) for non-manufacturing environment.

In the present thesis the aim of modelling (as a combination of both conventional goals) is performance simulation of time-sensitive emergent behaviour in the organisation in order to forecast behaviour (e.g. choice making) after possible modifications and support the designing of information systems that could effectively assist decision-making and directing for choosing correct activities. This modelling task is not fully supported by the existing approaches. In order to suggest possible alternatives and find a solution to the given problem, in Chapter 2 the existing conventional approaches and tools will be reviewed for organisation modelling and their suitability for the modelling of time-aware emergent behaviour will be evaluated.

1.2 PROBLEM STATEMENT AND RESEARCH OBJECTIVE

Currently there is no known meta-model that can adequately represent expected emergent time-sensitive behaviour in multi-functional organisations. Even more, there is no generally accepted approach that would form the basis for such models.

A model that can be used for studying emergent behaviour in organisations would be useful for solving a variety of problems, such as:

- Modelling and assessing the response of the organisation to management decisions in a variety of operational situations by analysing and simulating how an organisation (or its actors) will possibly behave and why.
- Elaboration of a better information system for the organisation in two aspects: improve information processing (the model forms a backbone to the information processing system) and/or improve control over the actually emerging behaviour of the organisation (by providing more precise information and explicit feedback control).
- Assisting strategic decision-making by providing coherent information about the interactions of structural units of the organisation and about the consistency of their goals. The model can dynamically assess managing decisions, especially those leading to structural changes.

The existing multiple, sometimes very powerful methods assisting organisational modelling in different application domains are not aimed at nor suitable (as will be demonstrated in Chapter 2) for emergent organisations. For example, the used methods do not consider motivations and possible alternative behaviour of actors. Also most of the approaches do not regard time issues to be important and timing analysis is quite weak. As a result, detailed intra-organisation planning, model development and pre-simulation of possible behavioural patterns are not common and work processes in organisations are often modified without previous modelling. Adjustments are done intuitively, mistakes are not found quickly and it is difficult to determine their causes.

The research objective is to support the analysing and re-designing of time-sensitive work processes in emergent organisations by enabling to forecast (to a certain extent) the probable behaviour of an organisation after modifications. Its primary objective is to analyse the existing organisational models, determine in which criteria they are compliant with the current modelling requirements and to devise a model for representing, modelling and analysing emergent behaviour in a multi-functional human organisation that operates in a dynamic, time-critical environment. An additional objective is to elaborate and outline a methodology for devising suitable models and to illustrate how these models can be used for modelling process modifications.

In the present thesis the research problem is divided into the following steps:

1. The research started with the analysis of modelling domain, requirements and existing approaches.
2. Synthesis of a novel solution for the research problem:

In more detail the analysis comprises:

- Analysis of requirements for models of time-dependent emergent behaviour (the summary of requirements is given in Section 2.1 of the thesis).
- Review and comparison of the existing methods and tools for performance and process analysis for human organisations (based on literature available in scientific journals, conference proceedings, books and other papers, the review and comparison are presented in sections 2.2 – 2.4).
- Specification of what the components are for a suitable model and what technologies can be used for developing such components (analysed in Chapter 3 of the thesis, grounds for it are briefly reviewed in Section 1.3).

The synthesis part of the research is composed of:

1. Devising of a suitable model for representing time-aware emergent behaviour and for supporting change management by predicting organisational behaviour after process modifications. The research will indicate suitable components for those models, give rules for aggregating the concepts (i.e. determination or combining of their

structures), suggest a method for analysing time-sensitive behaviour on this model and outline a suitable methodology for composing the models (Chapter 4).
2. Illustrations on the implementation of the methodology in a particular organisation (case studies are presented in Chapter 5 of the thesis).
3. Observations and conclusions (presented in Chapter 6 of the thesis).

The thesis points out four hypotheses that it will aim to confirm:

1. Conventional approaches for the modelling of human organisations are not satisfactory for modelling emergent operational behaviour of multifunctional human organisations that operate in a dynamic environment
2. A suitable model should integrate the specification of work processes into modelling temporal requirements and behaviour of actors.
3. Experiences from soft real time and multi-agent research (as domains where time criteria and interactions of multiple actors are researched) are useful to consider for devising the model for an emergent organisation.
4. Such an integrated model is effective and useful to implement in planning the modifications of processes.

The research can be considered successful if suitable (existing or new) models for modelling emergent behaviour are found and suitable methodology for elaborating such models is compiled. The usefulness of the models will be evaluated according to their specific characteristics against initial research objective.

1.3 GROUNDS FOR THE CHOSEN APPROACH

An actual human organisation (as described in Section 1.1) consists of autonomous human actors who have free will, actual decision-making freedom and multiple personal characteristics. The expected emergent behaviour is actually a result of a number of different, concurrently and sometimes quite independently appearing factors and conditions (as described in Section 1.1.2).

We can never describe expected emergent behaviour entirely (i.e. all possible behavioural patterns in all emerging situations) and so far no such single method is known. Expected emergent behaviour cannot be derived from formal or non-formal specifications and is impossible to express it in analytical models for real cases. Expected emergent behaviour can be represented as a prognosis (in a form of behavioural patterns).

The estimated emergent behaviour of an organisation cannot be described by using conventional algorithmic approach. Therefore the modelling approach, presented in this thesis, is in principle different from conventional methodologies, as interaction-centred paradigm is suggested instead of algorithmic-based paradigm. *Interaction-based paradigm* (Wegner, 1998) concentrates on actions and their interactions – the behaviour of a system (output stream) depends on the previous input stream over the input history (Meriste and Motus, 2002). The formal basis for interactive systems is the interactive systems theory.

As a result, we should choose an integrated set of methods and tools that can capture the organisation and its behaviour from the most important aspects, including goals and processes, temporal conditions, actors' preferences, informal communication and, if possible, to some extent also possible grounds for behaviour. It is preferred that we should try to utilize and combine the existing suitable tools and approaches. Possible alternatives will be carefully investigated and, if suitable, some of the corresponding techniques can be implemented here to a certain extent. The grounds for the used approach are briefly described hereunder.

For the specification and modelling of processes (including their characteristics, sequence and interrelations) in this research the Unified modelling Language (UML; Booch Rumbaugh and Jacobson, 1999) is used. For the current approach the UML is considered as the most suitable modelling notation because of its characteristics and wide availability (described in more detail in Chapter 3): the UML allows to model different aspects and views of the system, it is logical and visual, is accepted by Object Management Group (see OMG, 2005), UML describes different parts of the model in a single language. If in some cases (e.g. formal verification or modelling very specific systems) specific languages are needed, this possibility is included in UML specifications, for instance UML profile for Performance, Schedulability and Time (OMG, 2003b). The UML represents object-oriented approach that is in principle interaction-centred: although grounds for objects' behaviour differ substantially from grounds for the behaviour of autonomous agents, object-oriented paradigm represents adequately the static structure of an organisation and the activity frame of an object, and object-oriented paradigm is conceptually adequate for representing a part of the organisation. Therefore the UML can easily model interactions. The UML is in more detail described in Section 3.1.

Time-sensitive operational behaviour with tight time criteria and interactions between multiple actors has certain similarities with the requirements and model characteristics for process models in *soft real time* applications, message passing or batch manufacturing. Therefore solutions for the specification and verification of timing criteria will be searched from corresponding approaches in soft-real-time systems and for real-time production domain.

There are different candidates for modelling time issues of processes and actors' behaviour. For example, time as a partially ordered sequence of events can be modelled and process synchronisations analysed by using Petri Nets (Zimmermann, 2005) or CSP – Communicating Sequential Processes (Hoare, 1985). The latter can be combined with role-activity diagrams (Abeysinghe and

Phalp, 1997). The use of Petri Nets can be seen as additional possibility in domains where metric time has no importance. Petri Nets are popular in various domains (e.g. DiCesare et al., 1993; Zhou, 1995; or examples of the usage of Petri Nets introduced by Mortensen and Rölke, 2003). The developed versions of Petri Nets, like coloured Petri Nets (Mortensen, 2003) or timed Petri Nets (Wang, 1998), had widened the possibilities of classical Petri Nets and overcome some of its disadvantages.

The Q-model (Motus and Rodd, 1994) enables a wider specification of time (each process may use its own time paradigm), it implements effectively the approach of interactive computing; and, in principle, allows verification of timing criteria for processes and interactions. The Q-model forms a basis for the UML profile for Performance, Schedulability and Time (OMG, 2003b) and in this regard it can be seen as a RT UML model processor (see Selic and Motus, 2003 for more details). UML models that correspond to the UML profile for Performance, Schedulability and Time can be converted without any loss of information or violation of requirements into the Q-model notation. As a result, the Q-model is an excellent candidate used together with UML for a model component for the representation and analysis of timing characteristics of processes. The Q-model is in more detail described in Section 3.2.

The main activator of processes in a human organisation is the human actor who is autonomous in its decisions – it has free will that may cause differences in planned process criteria. Therefore models should concentrate not only on process flows but also on actors' behaviour and interactions between actors. In the current approach the behaviour of each actor is taken as a basic entity for describing organisation's behaviour. Behaviour of an actor is seen as a cumulative combination of simple behavioural activities.

Such a multifunctional human organisation where different actors co-operate towards a common goal, but still perform their tasks independently from each other and synchronise their activities by using interactions (e.g. by message passing) is very similar to a multi-agent system.

A conventional multi-agent system consists of multiple proactive agents that interact with each other, e.g. communicate by sending messages (Ferber, 1999). An agent is seen as a (computational) entity that is situated in some environment and that is capable of flexible, autonomous activity (action and interaction), in order to meet its design objectives (Ferber, 1999).

Agent-based approach has similarities with interaction-based approach; for example, agents have stream input and their behaviour (output stream) depends on the overall input history, if designed so. There are multiple studies on models of interactive computations in the agents' community, e.g. Ferber (1999), Wooldridge (2002). The principles of agent behaviour are quite similar to these of human behaviour. Multi-agent systems are reviewed in Section 3.3 of the thesis.

Human behaviour in agent systems can be expressed by several human characteristics like beliefs, desires, etc (e.g. BDI-agents, Rao and Georgeff, 1995, or socially intelligent agents, Edmonds, 1997), but those characteristics are more difficult to express in conjunction with pragmatic modelling of business processes, as analysed in Section 3.4.

Based on the description of application domain and definitions, given hereinbefore, we can now re-specify the research objective by using more specific terms: this research develops an approach for the modelling of emergent operational behaviour in multi-functional organisations by applying interaction-based approach, some process control techniques for soft real time systems and a multi-agent approach.

1.4 ORIGINAL CONTRIBUTION

1.4.1 Relation to the existing work of other authors

The proposed solution is based on the ongoing research of a number of authors in multiple areas:

1. Elaboration of RT UML (OMG, 2003a, 2003b) and its model processor the Q-model (Quirk and Gilbert, 1977; Motus and Rodd, 1994; Selic and Motus, 2003).
2. Development of UML by OMG from current version 1.4 to version 2.0 for being efficiently able to represent business processes (OMG, 2003c), although, it is also possible even now.
3. Development of UML-like languages for multi-agent systems (e.g. Agent UML; Bauer, Mueller and Odell, 2000; or other, see Bauer and Müller, 2003).
4. Research and analysis on interaction-based models of computation (Wegner, 1998; Wegner and Eberbach, 2004) and representation of process temporal criteria for human organisations.
5. Progress in studying social phenomena and personal human characteristics with multi-agent systems (e.g. Ferber, 1999; Wooldridge, 2002).

Although those topics have been studied for several years already, no final positions in those research areas have been elaborated so far and many underlying and related theories (e.g. time and schedulability issues in UML) have only recently been formulated or they are not even fully formulated yet. The research literature was mainly studied in 2004 and therefore this thesis may omit more recent reseach of other authors.

The base organisation for case studies on the implementation of the introduced methodology has been the Estonian Police. The Estonian Police is a hierarchical and countrywide geographically distributed organisation with the size of approximately 5000 employees. This organisation performs emergent operational behaviour for executing its multiple tasks in law enforcement domain. Currently there is an ongoing quality management project (based on

Balanced Scorecard) for specifying refined goal functions and evaluating actual outcome of performance. There are also other multiple development projects in management and information systems area in the organisation that enable to observe and study different solutions for specifying and comparing planned and emergent behaviour.

1.4.2 Novelty and original contribution

Original contribution of the research comprises three topics:

1. The thesis gives a systematic analysis (in Chapter 2) of the existing organisational process models and their evaluation with respect to modelling timed and ad-hoc behaviour, especially decision-making. This review concludes that the existing conventional approaches are not in principle suitable for modelling the emergent operational behaviour of multi-functional human organisations.
2. The requirements for suitable modelling methodology are analysed and a new modelling methodology is introduced (in Chapter 4) as the main contribution of the thesis.
3. The introduced novel approach combines conventional approaches for the modelling of processes (UML and some Enterprise Modelling principles) with the precise modelling of timing characteristics of inter-process interactions (the Q-model) and simulates emergent behaviour as a result of interactions of multiple actors. In this way the research implements real-time and multi-agent approaches for modelling human organisations. Although multi-agent descriptions and simulations are widely used, they are not used in conjunction with detailed time-aware specification of planned business processes (at least no references found so far).

The original novel approach is more suitable for modelling emergent behaviour than conventional approaches since it solves the problems related to the representation of actors' freedom and choice making while still maintaining model usability from the process perspective.

The suggested approach for modelling emergent behaviour in organisations is based on three concepts:

1. *Organisation model* that describes the organisation at the moment of modelling.
2. *Change model* that encapsulates the expected behaviour of the organisation before and after process modifications, represents different organisational states during the modification process and thus assists in change management.
3. The modelling methodology – a suggested process of devising *organisation model*, *change model* and the implementation of modification suggestions in the organisation.

Organisation model encapsulates the organisation from four viewpoints:

1. General description of an organisation, given in a natural language.
2. Planned and actual work processes of an organisation and related actors are listed in a special form of *process table* and formally specified by using UML use case, activity and sequence diagrams (timing and interaction constraints are captured by RT UML).
3. Temporal criteria of processes (e.g. activation frequency, duration, data consumption and production times) and interactions between processes (e.g. synchronisation and data exchange) are formally analysed and more precisely simulated in the Q-model that is the theoretical basis for UML profile for Performance, Schedulability and Time. In this way the Q-model is used as a model processor for RT UML.
4. Selected proactive components, behaviour related to subjective human factors (e.g. actor's goals, personal interests and decision-making) and interactions between actors are specified by using agent class diagrams. In order to visualise expected emergent behaviour, a subset of the system is chosen for multi-agent simulation.

The overall modelling, model development and change management lifecycle in the current approach is an iterative process (cycle) where more specific results are reached stepwise. The current modelling methodology introduces four general stages:

1. Formulation of the *analysis task*, devising a general description of the organisation and specifying its work processes in UML.
2. Completing the *organisation model* by specifying temporal criteria of processes and proactive components of the organisation.
3. Experiments and simulations on the *organisation model* and composition of a *change model* and modification suggestions.
4. Actual change management – an implementation of organisational modifications in the actual organisation, monitoring and evaluation of implementation results.

The implemented approach is interdisciplinary and incorporates semiformal methods of systems analysis for investigating time-sensitive behaviour. In this way the suggested novel meta-model is not only a "reference" of organisation-specific models for the analysis of time-sensitivity in a specific system but also an investigation tool for studying time-sensitive behaviour of organisations.

This thesis also presents a case study on modelling emergent time-dependent behaviour in organisations. The proposed methodology was implemented in the Estonian Police in 2003 for analysing information exchange processes on stolen vehicles. That analysis revealed gaps and repetitions in work processes and in information exchange between authorities. On the basis of the suggestions the corresponding information system was later actually modified. The same methodology has been partially used again for some other case studies. So far the methodology is not implemented in everyday development cycle of any organisation and it needs further refinement and specification.

It is expected that the suggested modelling methodology will give a solution for the adequate modelling of emergent time-sensitive operational behaviour of organisations. The devised meta-model will satisfy the requirements described in Section 1.2: it describes the organisation from the viewpoints of process, time and actors, and enables the planning of process modifications. The model enables to visualize simulations and predict the behaviour after modifications. By using the model, the management level in the organisation could confirm or reject modification plans and performs more appropriate decisions.

Models, developed by using the suggested methodology, can also be used for several different simulations and a profounder research on the grounds of behaviour for design information systems that support decision-making.

1.4.3 Publication of the material contained in the dissertation

The basis and the required analysing power of suitable models are briefly analysed and the suggested novel methodology outlined in the 11th International Conference on Information Systems Development ISD 2002, Riga and published in Savimaa, 2002a.

The combination of how to integrate the UML, the Q-model and a multi-agent approach for the modelling of organisations is in more detail described at the International Conference on Cybernetics and Information Technologies, Systems and Applications CITSA 2004 / the 10th International Conference on Information Systems Analysis and Synthesis ISAS 2004, Orlando (Savimaa, 2004a).

As a continuation to the material presented there, the modelling of planned modifications and essentials of the *change model* is in more detail introduced and described in the 8th World Multi-Conference on Systems, Cybernetics and Informatics, SCI 2004, Orlando (Savimaa, 2004b).

The methodology is implemented for analysing information exchange in the Estonian Police in relation to vehicle theft (Savimaa, 2003a). This implementation is shortly described and representation of the organisation as a multi-agent system is discussed in the Proceedings of the Estonian Academy of Science (Savimaa, 2005).

Additionally, the methodology and modelling of an organisation as a multi-agent system was presented in the 7th Estonian Winter School in Computer Science (EWSCS) Palmse, 2002 (Savimaa 2002b) and the modelling of co-operation and personal interests in a hierarchical organisation (Savimaa, 2003b) was described in the poster session in 2nd Estonian Summer School on Computer and Systems Science (ESSCaSS'03) (see ESSCaSS, 2003). The overall methodology for the integration of the UML, the Q-model and agent technologies was presented in the poster session in 3rd Estonian Summer School on Computer and Systems Science (ESSCaSS'04).

The review of the PhD thesis, problem description and devised solution were presented during the Doctoral Consortium of the 16th International Conference on Advanced Information Systems Engineering CAiSE 2004, Riga and published in Savimaa, 2004c.

1.5 ACKNOWLEDGEMENTS

I would like to express my special gratitude to Prof Leo Mõtus, supervisor of my Doctoral thesis, for his ability to present in a very interesting way the problems that emerge from the integration of engineering activities and the real world, especially in the domains of embedded systems, real-time engineering and multi-agency. He has also supported the research of the thesis very much with his competent suggestions and knowledge about state-of-the-art in this research domain. I would also like to thank Prof Ennu Rüstern for creating a scientific and fruitful atmosphere in the Faculty of Information Technology and earlier in the Institute of Automatics. He always gives helpful answers to different questions that students always ask during their studies. I am also very grateful to the critics of my thesis, Mr Wilfried Brauer and Mr Merik Meriste, for their valuable comments and suggestions for improving the final version of the thesis.

Many thanks to colleagues from the Estonian Police for the inspiring work atmosphere, constructive opinions and remarks and for the possibility to use work tasks as examples for implementing the elaborated original methodology. Special thanks to Ms Margit Sepp for proofreading the thesis.

This research is partially financed by the Estonian Science Foundation (grants No 4860 and 6182) and the Estonian Ministry of Science and Education (Research project No 0142509s03). The participation in the 8th World Multiconference in Systemics, Cybernetics and Informatics (SCI 2004) as well as in the 10th International conference on Information Systems Analysis and Synthesis (ISAS 2004) jointly with the International Conference on Cybernetics, Information Technology, Systems and Applications (CITSA 2004) has been partially financed by the Estonian Information Technology Foundation under the Tiger University programme.

1.6 OUTLINE OF THE THESIS

The rest of the thesis consists of five chapters. The requirements set to models in the described criteria are surveyed in the next chapter (Chapter 2). Also conventional methods and tools used for analysing the organisational performance, modelling of processes and supporting of change management are reviewed and evaluated against the given requirements.

Chapter 3 reviews in general lines the suggested suitable grounds for the novel methodology. It analyses interactivity questions, gives a brief survey of UML and the Q-model and discusses the suitability of the Q-model and UML 2.0 for modelling all necessary aspects of time-sensitive processes. The chapter continues with the analysis of behavioural aspects of an organisation. Basic principles of agent technologies are introduced with a brief review on how different approaches handle related issues. Also aspects related to the modelling of human organisations as mufti-agent systems (for example modelling of processes, motivations, co-operation and interactions) are reviewed and illustrated.

The novel methodology as an original contribution to modelling emergent time-dependent operational behaviour in organisations and displaying possible modification suggestions is introduced in Chapter 4. The methodology with all its components is introduced stage by stage; also a suitable preliminary meta-model of an organisation is given.

Chapter 5 is an illustration of the usage of the methodology. Three case studies are presented; the main example is taken from the law enforcement domain. Organisational processes, use-cases and class models are specified and the results of simulations described.

The contribution to problem solving, evaluation of proposed approaches, conclusions and future directions are reviewed in Chapter 6. The thesis concludes with references.

2 CONVENTIONAL METHODS FOR ORGANISATION MODELLING

This chapter reviews the existing conventionally used methods and tools for describing organisations, evaluating their performance and modelling processes. In the first section necessary requirements for models are listed that are able to capture emergent operational behaviour. The review of conventional organisation modelling approaches in the following sections begins with a simple analysis and planning tools. The review continues with more complex tools for measuring efficiency and concludes with solid methods for modelling processes and change management. Methods are selected to illustrate capabilities and disadvantages of each group of tools. The chapter concludes with a discussion on the suitability of conventional approaches and suggests a possible combination of the existing tools for modelling and evaluating organisational processes.

2.1 MODEL REQUIREMENTS

In general, a suitable model must be (according to Selic, 2003):

- accurate (i.e. faithfully represent the actual modelled system),
- inexpensive (i.e. the model must be much cheaper to construct and study than to construct or modify the actual system),
- understandable (i.e. expressed in a form that is readable by different users),
- predictive (i.e. can be used to derive correct conclusions about the modelled system),
- must emphasize important aspects of the system while hiding irrelevant ones.

Models for organisations fail often in accuracy and predictability. In both cases expenses of wrong decisions are difficult to measure because of the implicit structure, insufficiently defined choices and partially defined assessment criteria. This has often led to the practice that every management decision can be later justified and/or “slightly modified” to avoid the most serious mishaps, hence predictive modelling before decision-making is neglected. This is the reason why predictive models are often considered “too expensive”. In reality the use of predictive models may help to avoid serious mistakes (potentially leading to economical losses and definitely harming the psychological climate). Those models also enable a later audit and evaluation of the quality of decision-making procedures as well as the indirect assessment of the efficiency of the organisation.

In most cases the organisation, which is to be modelled, is already operating and has its everyday tasks and ongoing activities. Therefore models should first be able to encapsulate the existing processes. A good organisation model supports the measuring of operational efficiency and investigating of reasonable modifications of the organisation in order to improve the management quality. Modelling is also important due to the fact that testing the model is much cheaper than possible errors occurring in the real world. The cost of modifications can be assessed and controlled if the model supports the analytical and simulation study of the modifications. Traditionally the relation of different work processes to a specific structural unit, characteristics of units and co-operation between different units in the organisation is not modelled, whereas proper associations of work processes and units also improves organisational efficiency to a great deal and in this way also the outcome of organisation's behaviour.

Models of work processes (together with their starting, interaction, information exchange and finishing conditions) are mostly used on the level of IT and development specialists for devising information systems. The management uses the collected information (i.e. results of modelling) for strategic decisions. They have to plan the necessary allocation of resources to allow maximum operational benefit in the framework of overall organisational goals. To be supported in decision-making, managers need more sophisticated models that describe the organisation and its dynamic behaviour adequately. It is essential for managers to comprehend such models. This means that a suitable software product has to be made for managers.

A model of an organisation should encapsulate and express all the important characteristics of an organisation. Models of organisations that describe planned behaviour and support ongoing decision-making have to meet the following general requirements:

1. Possess the ability to encapsulate the most important work processes and organisational aspects (environment, organisation's structure, structural hierarchy and processes).
2. Possess the ability to deal with the hierarchy of business processes (Carley and Prietula, 1998). Tasks and processes may consist of simpler tasks or business processes. This means that business processes will be specified in terms of other business processes. Models should handle a hierarchical approach to different levels of business processes.
3. Goal representation, task decomposition and allocation, decision-making and delegation, roles in the organisation and resource management have to be represented at the required level of details, chosen by a designer.
4. It should be possible to represent interactions (e.g. communication and information exchange) and co-operation mechanisms between different actors (within organisation, inter-organisations and with environment) with regard to work processes.

5. It should be possible to analyse the relation of the existing information systems to everyday life of an organisation (support to processes, realisation of workflow, data / knowledge modelling, man-machine interfaces, individual decision-making accessories).
6. Possess the ability to model processes for development activities (i.e. established routines for enterprise modelling, representation of development strategies, modelling of tactical planning, support of analysis, modelling of business integration, co-operation between organisations, clients, public relations and reuse of processes and its components).
7. Support a straightforward interpretation of modelling results into day-to-day activities. This also includes the ability to switch from the vocabulary of management and employees to the vocabulary of modelling and back – there must be the possibility to explain modelling results easily and quickly to persons involved (decision-makers etc).
8. Enable persistent adjustment of the model, in accordance with the changes in the organisation.

In addition, models that specify time-sensitive emergent behaviour in dynamic environment and give reasonable realistic information about the behaviour of different actors in an organisation have to meet the following specific requirements:

9. Enable to model all temporal criteria of processes and interactions.
10. Enable to compare both behaviours – the planned one and the real outcome (i.e. existing, actually emerging behaviour) – as it is known that planned and actually emerging behaviour in an organisation never coincide.
11. Are able to observe and check all the essential aspects that influence the organisational behaviour, including a generic model for analysing employees' behaviour (this is necessary since the performance of organisations depends heavily on employees).
12. Are able to represent formal communication related to the primary tasks, work processes and goals of the organisation as well as reasonable amount of informal communication related to employees' opinions, beliefs, intentions, interests, etc., has to be described (Ferber, 1999).

Methods, reviewed in the thesis will be evaluated against those twelve requirements. In reality all those characteristics cannot be encapsulated in a single model – it would be too time- and labour-consuming and difficult to use and update. It is reasonable to decompose models into a hierarchy of closely interacting sub-models. Each sub-model is used for describing and analysing certain aspects of the organisation, or its structural units. Each sub-model can be developed separately and they together form a network of models that describes the whole organisation (or part of the organisation, depending on the availability of sub-models). Such approach is also taken as the basis for the suggested novel model methodology described in Chapter 4.

Three critical aspects are pointed out that have not been sufficiently appreciated in the modelling of organisations:

- Organisational goals, tasks and capacities (resulting from the resources and the way they are used) do not usually match. When giving tasks to an organisation, details of work processes are not specified. Also the analysis is very seldom made on whether the goals, capacities and constraints contradict each other– i.e. in principle, the analysis on whether the tasks are achievable. Quite often time constraints are not taken into account in models, whereas the tasks leading to the achieving of goals are time-sensitive. Therefore, a model should allow a complex hierarchical process and resource analysis as well as the handling of sufficiently sophisticated time category.
- Many decisions during the performance of a task are left to structural units at lower levels of the organisation. Each structural unit in an organisation has its own goal and the employees have personalities of their own, with their own wishes, opinions and personal goals. The actually emerging behaviour is caused by a multitude of factors, not necessarily related only to work or the organisation. For the sake of efficiency, it is important to match the goals of the organisation, its structural units and these of its employees, at least to a certain extent. Models should allow the analysis of interaction on the levels of structural units as well as in each structural unit on the level of an individual employee or groups of employees.
- Organisations have to function continuously. Restructuring and structural adjustments are to be carried out “on-line”, without suspending everyday activities and quite often the modifications have to be done without increasing day-to-day operating costs. To guarantee permanent support from the models, they are also to be regularly updated and have to support and assess “online” changes in the organisation.

It is essential that a model of an organisation reflects the current state of the organisation (its structure and process description). This is possible if the model is persistently updated so that it can keep pace with the changes in the organisation. The hierarchy of interacting autonomous sub-models supports the activities – whenever a structural unit of an organisation is modified, the changes are entered into the corresponding sub-model (in the ideal case, it is recommended that changes are first implemented in a model or sub-model, the effects are thoroughly studied and only then implemented physically).

2.2 EXISTING METHODS AND TOOLS FOR PERFORMANCE ANALYSIS

In this section the existing tools and methods for analysing an organisation's efficiency and performance are reviewed and compared. The survey begins with simple analysis tools that are used for evaluating single processes, tasks or activities and proceeds with the methods of assessing the organisation's overall performance and efficiency in relation to the goals. More sophisticated methods for modelling complex processes and change management in the organisation are reviewed in the next section (2.3). All the reviewed approaches and tools are evaluated against the requirements given in Section 2.1 and compared with each other by using a common set of characteristics:

1. The aim of the method or tool,
2. The scope of the method or tool,
3. Necessity of a computer aided tool for implementing the method,
4. Known computer aided tools that support the method,
5. Strengths of the method or tool for the modelling of organisations,
6. Weaknesses of the method or tool,
7. Consideration of time issues in the method or tool,
8. Consideration of employees (motivations, attitudes, etc).

The comparison is given in Table 2.1, Table 2.3, Table 2.6, Table 3.1, Table 3.2 and Table 4.10. These tables are an original contribution of the thesis. The survey of conventional methods and tools concludes in Section 2.4 with the preliminary estimation on the resultativeness of the existing methods in respect to modelling emergent behaviour in the organisation.

2.2.1 Analysis tools

There exists a large number of different planning and assessment approaches, the task of which is to evaluate selected activities or capabilities of the organisation in relation to a specific problem. Those approaches can be divided into two classes, starting from simple approaches:

1. Tools for work process analysis, goal refinement and estimation of necessary activities related to the fulfilment of a specific task. Such tools enable to obtain information on how a specific goal can be achieved, how the processes are synchronized and what deliverables (i.e. information and/or materials) are elaborated at a certain stage in a project lifecycle. Known representatives in this class of tools are Gantt chart (Clark, 1922) and PERT (MindTools, 2003a).
2. Tools evaluation of possible choices, activities, potential solutions and the existing situation. Known approaches as SWOT (MindTools, 2003a) and PEST (NetMBA, 2003) belong to this group.

The **Gantt chart** (Clark, 1922) is one of the well-known approaches for goal refinement and the evaluation of progress in the fulfilment of sub-goals. In the Gantt chart approach a project is broken down into different tasks that are mapped onto the timeline of the project. Gantt charts give a clear illustration of the project status at any given instant of time. A Gantt chart can encapsulate the most important processes and environmental condition and it accepts a two-level hierarchy of business processes. It also satisfies the requirements related to the modelling of goal representation and task allocation and the charts can hold information about interactions between processes. The Gantt chart does not fulfil other general requirements for models presented in Section 2.1 or specific requirements given in the same section (except the ability to compare planned behaviour and real outcome in simple project charts).

Another similar tool, the **PERT** (Program Evaluation Review Technique) chart, is a project management tool used to schedule, organise and co-ordinate tasks within a project (MindTools, 2003a). The PERT is similar to the **CPM / CPA** (Critical Path Method / Critical Path Analysis) and they are often used as synonyms. A PERT chart presents a graphic illustration of a project as a network diagram. The PERT chart overcomes the problem of Gantt chart and indicates also inter-task dependencies. This means that on the PERT chart or in the CPA it is easy to describe what sub-tasks are to be completed in time for the whole project to be completed on time. It also indicates which sub-tasks can be delayed without jeopardising the whole project. For complex projects the PERT / CPA chart is more difficult to interpret. At a high abstract level and in case of simple projects the PERT / CPA fulfils the general requirements for models, presented in Section 2.1, but does not still satisfy most of the given specific requirements.

The **SWOT** (MindTools, 2003b) is a technique for analysing strengths and weaknesses of a project. Additional benefits (seen as opportunities) and risks (seen as threats) can be described as well. Strengths and weaknesses depart form the organisation, its environment, or from planned projects. Opportunities and threats are determined as positive or negative results that emerge from the interference with the environment. The SWOT can be used for analysing ongoing routines as well as the planned modifications of work processes or for choosing from multiple action plans. SWOT analysis can be extended by **PEST** analysis (see NetMBA, 2003) where environmental conditions (that cause opportunities and threats) are analysed from political, economical, social (sociological) and technological points of view. The SWOT is aimed at the assessment of a single problem, solution or capability. Therefore it does not meet most of the general or specific requirements for models, presented in Section 2.1. Most important, it does not support any detailed temporal specification and does not analyse actors' motivations.

The represented techniques can be combined (e.g. the PERT and the SWOT) to build larger modelling environments. In Table 2.1 the reviewed methods are compared form the viewpoint of necessary characteristics required for the modelling of organisations.

Table 2.1 Comparison of some analysis and modelling methods

No	Property	Gantt	PERT / CPA	SWOT
		1	2	3
1	The aim	Planning of activities as components of a project and their timeline	Planning of actions, their dependencies and required time in a project	Evaluation of strengths and weaknesses of a project or activity
2	The scope	A project (e.g. devising an information system)	A project	A project or choice of action plan
3	Necessity of a computer aided tool	No	No	No
4	Existing tools	Yes (e.g. MS Project)	Yes	Not known
5	Strengths	Simplicity	Task dependencies	Different aspects are analysed
6	Weaknesses	Used for a project only, pure list of activities on a timescale	Used for a project only, pure list of activities on a timescale	Description is based on subjective opinions
7	Consideration of time issues	Schedule for a single task only	Yes (sequence and interactions of tasks)	No
8	Consideration of employees	No	No	If analysed as a target

2.2.2 Practical evaluation of the performance

Methods for the practical evaluation of an organisation's performance often assess the overall behaviour of an organisation but they can also focus only on a specific sub-goal, activity line or structural unit. Such approaches can be categorised into two classes: tools for one-time evaluation of a quality level and continuous methods for ensuring process quality.

Normally a progress audit or quality evaluation is performed by outside actors (e.g. auditors), but to some extent can also be used for self-assessment. The **People Capability Maturity Model** (P-CMM), a member of the family of the Capability Maturity Models, is a framework that helps organisations to manage human resources and enables to analyse processes inside an organisation (SEI, 2001). The P-CMM classifies an organisation into stages of maturity, depending on which processes are implemented for planning and evaluating the organisation's performance. It gives recommendations for the general improvement of the organisation's performance, assuming that they operate in relatively stabile environment.

The examples of more specific audit and evaluation standards in information technology and security domain are Code of practice for information security management ISO/IEC 17799 (ISO, 2000) and a generally accepted standard for good IT security and control practices COBiT – Control Objectives in Information and related Technologies (ISACA, 2005). Those approaches, as a rule, describe only general, not detailed activities that should be implemented for preventing risks. Process specifications are not included nor considered during assessment and temporal criteria; informal communication and actors' choices for behaviour are not modelled.

Tools for one-time specific audit and evaluation are not capable for modelling emergent behaviour in organisations since the used model expresses the organisation only on a very general level, it does not consider time issues and viewpoints of multiple actors. Also, these tools do not allow to prognosticate future behaviour after possible modifications. In conjunction with other more specific tools that are able to model emergent behaviour, such methods (e.g. CMM) can be used as a framework for evaluating an organisation's efforts and progress.

The second group of evaluation methods are approaches for ensuring measurable, persistent and continuous process quality (e.g. Balanced Scorecard and Best Value). Those approaches stress that quality of service can be achieved only if organisational processes and activities of all actors are always performed at the required quality level.

Best Value (Boyne, 2000) is an approach used in the United Kingdom for assessment and ensuring persistent and continuous process quality and quality of service in local government organisations. Best Value uses a set of performance indicators to measure the level of service and organisational efficiency in local governments. The departments in the central government in the UK set indicators (Best Value, 2003b) in different domains (e.g. construction, criminal justice, education, information technology, social care, etc as given in the IPF, 2003) in all regions and recommend the establishing of targets for each performance indicator, if applicable, for every authority and activity domain. The assessment of performance is done by comparing the indicator with the target and with the average level of this indicator in other organisations. Best Value approach stresses that achieving quality service provision is a process rather than a one-time goal. The Best value approach can be used only as an additional measure for modelling emergent behaviour, since it does not support most of the general and specific requirements for models (e.g. interactions and temporal criteria) as specified in Table 2.2. This table compares the analysed methods against the necessary characteristics for models, given in Section 2.1.

The **Balanced Scorecard** (BSC; Kaplan and Norton, 1996) is an assessment method for the analysis and evaluation of organisational performance. The BSC measures the activity from four aspects: finances, clients, business processes and learning and development. Determined goals, indicators, targets (target values of indicators), action plans are used in each of those aspects. The BSC compares different indicators that describe activity of a certain (e.g. coming or previous) period with target indicators.

Table 2.2 The comparison of model requirements and characteristics of BV and BSC

No	Requirement	Best Value	BSC
		1	2
1	Capturing the most important work organisational aspects	Is possible, depending on a specific model	Yes
2	Dealing with hierarchy and different levels of business processes	Yes, still on a general level (detailed processes)	Yes
3	Modelling of goals, tasks, roles and resources	No	Yes, can be considered for a better model
4	Representing of interactions and co-operation mechanisms	No	Those can be considered as targets, but usually omitted
5	Relation of the existing information systems to everyday life of an organisation	No	Yes, if specially considered
6	Process modelling for development activities	Can be specially considered as a target	Yes, if specially considered
7	Straightforward interpretation of modelling results into day-to-day activities	No	Is possible
8	Possibility of a persistent adjustment of the model, in accordance with the changes in the organisation	Yes	Yes
9	Modelling temporal criteria	No	No, but it is possible to specify some simple criteria (e.g. duration) as a target
10	Modelling and comparing the planned behaviour and the real outcome	Yes	Yes
11	Observing and checking of all the essential aspects that influence the organisational behaviour, including a generic model for analysing employees' behaviour	No	No (unless specially built)
12	Description of formal and informal communication	No	No

Table 2.3 Comparison of process quality evaluation methods

No	Property	P-CMM	Best Value	BSC
		4	5	6
1	The aim	Evaluate and characterize the maturity of work practices, establish a program for continuous development	Evaluate the characteristics of the service level (standardised comparison to other participating organisations with similar duties)	Evaluate the organisation's performance (how well goals are supported, what the existing and necessary activities are)
2	The scope	Organisation's performance (human resources and processes)	Organisation's performance	Organisation's performance
3	Necessity of a computer aided tool	No	No	No
4	Existing tools	No (but are models)	Not known	No
5	Strengths	Reviews processes and human resources, gives estimation of the current level and framework for improvement	Common criteria for a set of participating organisations, comparison, placement	Integrated approach for all aspects of activity
6	Weaknesses	Too general	No analysis for an organisation	Too much management and financially oriented and little process-oriented
7	Conside-ration of time issues	No	Can be used as a performance indicator	No
8	Conside-ration of motivations, attitudes, etc	No	Can be monitored and evaluated against targets	No

The BSC can be used as a system for supporting management decisions. The overall strategic goal has to be de-composed into specific strategic tasks in all four aspects. Tailoring the BSC for an organisation is a systematic process during which a mission and strategy for the organisation (or a unit) is transformed into well-determined targets and indicators and suitable performance indicators are chosen. Often performance indicators are used, among others also satisfaction of employees, their loyalty and the efficiency of their activities. The BSC approach emphasizes the importance of collective participation (especially the participation of higher management) in choosing performance indicators and completing the implementation plan.

The BSC offers a detailed analysis of an organisation's performance and the efficiency of processes. It compares the existing work processes against the goals and suggests modifying them or introducing new ones, if needed. The BSC does not support a detailed analysis of work processes; it is more like an upper-level planning tool and does not ensure that the chosen work processes are better or more efficient. Therefore the BSC cannot still support the modelling of emergent behaviour, as indicated in Table 2.2. The BSC can be used for the initial set up of the modelling of an organisation and for merging general goals of the organisation with concrete parameters of detailed work processes. Table 2.3 compares similar characteristics of the analysis and evaluation methods presented in this section and the methods reviewed in Table 2.1.

2.3 METHODS FOR ORGANISATION MODELLING AND CHANGE MANAGEMENT

This section continues to review conventional organisation modelling methods with the approaches that model an organisation from multiple perspectives (i.e. processes, resources, actors and roles). The review concentrates on workflow management, information systems development and enterprise modelling together with change management methods and tools.

2.3.1 Workflow management

Workflow management (WFM) is the computerized facilitation or automation of a business process in whole or in part (WfMC, 1995). Its focus is on the automation of processes involving combinations of human and machine-based activities and interaction with IT applications and tools. Workflow management can take care of executing the hierarchy of tasks and problems related to resource allocation as well as supporting the modelling of complex, integrated workflows.

The most prevalent use of workflow management is within office environment in staff intensive operations such as insurance, banking, legal and general administration, but is increasingly exploited also in other domains. There are several products available that use workflow management technology. However, there are no standards defined yet, only a reference model offered by the WfMC (1995).

The purpose of the Workflow Reference Model (WF RM) is to provide a common framework to support the development of workflow management systems, identifying their characteristics, terminology and components. The detailed specifications will be developed separately, e.g. manufacturing companies define new process reference models for integrating known concepts of communication, knowledge and enterprise modelling, business process re-engineering and benchmarking into one framework.

From process perspective any activity of the organisation can be viewed as a collection of interdependent processes. These processes change as organisations evolve over time in response to their environments. In order to keep the efficiency of an organisation, process changes have to be achieved quickly and efficiently. Process modelling technologies and notations are evolving so that processes, their consequences and affected actors may be understood with greater ease and increased visibility (Castano et al., 1999; Abeysinghe and Phalp, 1997; Montangero et al., 1999; Yu and Mylopoulos, 1993; Thrampoulidis, Goumopoulos and Housos, 1997).

Process modelling technologies are ranging from formal (mathematical) notations to more graphical (easier to understand) notations (Abeysinghe and Phalp, 1997). Formal notations may be executed as programs in order to study in detail the behaviour of processes but they are difficult to be presented to anyone other than an expert and therefore it is difficult to validate process scenarios with users, diagrammatic or graphical notations are excellent for process illustration but they do not often provide the benefits of process experimentation which can be gained with executable notations.

To some extent different approaches can be combined, e.g. CSP (communicating sequential processes) with role-activity diagrams (Abeysinghe and Phalp, 1997; Taveter, 1999): CSP defines a process in terms of event sequences in a diagrammic notation; an enactable process code is then automatically generated and used for modelling experiments, successful process scenarios are again presented to users with a diagrammic model. This approach is still not suitable for modelling emergent behaviour since the available mapping is time-consuming and only static models can be presented.

Another approach, COMIT (Kaplan and Carley, 1998) combines PERT chart information with the information of the organisational hierarchy and supports the measuring task accomplishing in a small organisation (one or two actors). COMIT allows user to model hierarchically decomposable tasks as series of stages, specify actions' duration, assign them to subtasks and specify several conditions (e.g. roles, experience, commitment, communication issues) for each actor. COMIT enables to simulate different scenarios on how an actor accomplishes a task. COMIT seeks both macro and micro-level views of the organisation and the emphasis is on co-operative work for very small organisations but it is currently not capable of simulating the complexities of large organisational projects.

The workflow management system (WFMS) defines, manages and executes workflows through the execution of software according to computer-represented workflow logic. Individual business processes may have a lifecycle ranging from minutes to days (or even months), depending upon its complexity and the duration of various activities. The WFMS, if designed correctly, can help employees to perform the necessary and most appropriate actions. In real life the WFMS does not cover the problem of overlapping nor does it fill the gaps between local (single) workflows, yet those issues are very important for the detailed analysis of estimated emergent behaviour in multifunctional organisations.

Table 2.4 The comparison of model requirements and WFM characteristics

No	Requirement	WFM
1	Capturing the most important organisational aspects.	Yes
2	Dealing with hierarchy and different levels of business processes.	Yes
3	Modelling of goals, tasks, roles and resources.	Partially: emphasis is not on roles but rather on processes
4	Representing interactions and co-operation mechanisms	In isolated workflows
5	Relation of the existing information systems to everyday life of an organisation.	Can be considered
6	Process modelling for development activities.	Must be specially considered
7	Straightforward interpretation of modelling results into day-to-day activities.	Yes
8	Possibility of persistent adjustment of the model, in accordance with the changes in the organisation.	Yes
9	Modelling temporal criteria.	Only sequence and duration
10	Modelling and comparing the planned behaviour and the real outcome.	Two models are needed
11	Observing and checking of all the essential aspects that influence the organisational behaviour, including a generic model for analysing employees' behaviour.	No
12	Description of formal and informal communication.	Yes, if specially considered and presented (as an workflow)

Workflow specifications can also be verified for ensuring the coherence between the model and actual system (Ortega and Soriano, 2001) and this to some extent covers the problem of overlapping and gaps inside a single workflow. Overlapping and gaps that emerge in real time as a result of interactions of multiple separately verified workflows are not detected (for example, if there is a requirement for the common resource). As a result, the workflow approach does not satisfy neither those nor some other requirements, given in Section 2.1. The model requirements and WFM characteristics are compared in Table 2.4.

2.3.2 Time and performance modelling

Temporal criteria are not so crucial for a conventional human organisation and therefore they are normally considered purely as deadlines for responding to the environmental situations (e.g. incoming messages), often given in days (for comparison – in technical systems response time is often in seconds or milliseconds). Timing characteristics of processes in the domain of non-production organisations usually lead to soft real-time type of constraints: each process is activated at a certain time, consumes necessary input data, performs required activities within certain duration and then terminates, producing the expected outcome. For a new task or for processing the next set of input data, the process is executed again.

The expected behaviour in this case can to a certain level of preciseness be described as a set of repeatedly activated processes that leads (or does not lead) to fulfilling the goal. Different problems will emerge if timing constraints and/or requirements are violated. For example, the result may become unreliable or outdated.

Synchronisation between processes can be described by using three types of relations:

1. Sequential execution – the next process is activated as soon as the previous is terminated.
2. Synchronous activation of two or more processes – the processes are activated simultaneously (e.g. main process and supporting or monitoring processes).
3. Asynchronous activation of two or more processes – the asynchronous processes are activated independently from each other, data is consumed when it becomes available and when the consuming process is able to use the data.

In general, such processes can be managed by using conventional management, control and psychological (both management and organisation psychology) approaches. Mapping of processes and related actors (units or employees) is usually done for the specification of information systems, but timing requirements for processes, definitions of time and resource conflicts as well as recommendations for solving the conflicts are usually not considered.

Additional difficulties, which cannot be solved by conventional approaches, emerge when specific timing constraints or real-time operations are required (e.g. resource usage duration, resource management in concurrent tasks, results of co-influence of different processes etc). Problems emerge if the environment surrounding the organisation (e.g. the society or other organisations) generates dynamic tasks and dynamically changing requirements to processes. To improve its performance in such an environment, the organisation should be pro-active, which may lead to the need for a deep analysis (mainly of causal relations, but often also of timing criteria) and to respective modifications in the structure and tactical behaviour of the organisation.

In this case the conventional management methods are insufficient since they do not take into consideration timing and proactivity-related issues. As a solution other (more specific) approaches (e.g. systems engineering approach) should be used for simulation and analysis for better modelling and planning results.

Systems engineering approach is typically used for modelling process behaviour in production domains (for example in chemical batch processes or control systems) where partially or fully automated processes are considered only from technological viewpoint. In non-production domain the systems engineering approach is seldom used and only on an abstract level for specifying process information flows.

2.3.3 Enterprise modelling

To satisfy social demands, enterprises need a new engineering paradigm ("life-cycle engineering") to assure process quality and safety. The paradigm has take into account not only processes of the organisation but also the background or environment questions (Aoyama, Batres and Naka, 2001). A suitable paradigm should support industrial information infrastructure, methodologies and various engineering views necessary to solve problems involved in all engineering processes from product design to its demolishment via manufacturing. Two of the most important aspects that the lifecycle engineering addresses for all organisations are:

1. Design strategies for manufacturing that emphasize to support the decision-making process rather than directly provide optimal solutions.
2. Maintenance of information models for upholding their interoperability and exchangeability over the lifecycle of products, processes and structural units.

Enterprise Modelling (EM) is a further elaboration of workflow management and is a structured technique for describing major aspects of an enterprise (Bubenko, Persson and Stirna, 2001). According to this view an enterprise is an organisation, consisting of multiple interacting workflows. During EM process an integrated and negotiated enterprise model is created, describing a specific enterprise (a private company, public authority, academic institution, or another organisation) from several different perspectives (e.g. processes, business rules, information, goals, actors and requirements, depending on the focus of a specific EM method). In general, the EM is a well-organised approach for modelling the current or future state in a large variety of organisations. The first EM methods and principles were introduced at the end of the 1960s. EM consists of two main components:

1. A meta-model, which defines constructs, syntax, semantics and graphical notation used to create an enterprise model. This forms the basis of the modelling language.
2. The development process for creating an enterprise model. This process suggests different paths for efforts to elicit, systematise and store the knowledge of business stakeholders or domain experts. Typical process paths are carried out in group-sessions and by taking and analysing interviews (Bubenko, Persson and Stirna, 2001).

The enterprise model consists of sub-models, such as goals model, business rule model, concepts model, business processes model, actors and resources model and technical components and resources model. Each of those inter-related sub-models focuses on one aspect of an enterprise.

The business rule model is used to define and maintain explicitly formulated business rules, consistent with goals model and the business process model for analysing the processes and flows of information and material in the enterprise.

In the actors and resources model human actors as well as non-human resources can have different roles associated with goals and business processes. EM focuses on planned work processes and activities and practically neglects real motivations and preferences of actors. There are two categories of actors – method providers and domain experts. The method provider is an actor who has knowledge with regard to the particular EM method, used for modelling in a specific case study. Domain experts are representatives of the modelled organisation who develop the organisation-related knowledge. Sandoe, 1998 defines organisational knowledge as three forms of memory: technological memory, structural memory (like norms, roles and conventions of behaviour, embedded into established routines and are occasionally formalised as procedure manuals), and mutual social memory that is retained in relationships between people and interactions between structural units of an organisation.

Different solutions can be applied on how meta-level thinking is distributed among the organization: some actors participate fully or almost everybody participates part-time. Since each of the involved persons has a different viewpoint, knowledge and ideas, a collaborative approach (Holsapple and Joshi, 2002) should be used that maximises the possible input form different information or knowledge sources.

Different persons from inside or outside the organisation can create organisational models. Business process engineering is a dynamic team-based endeavour that can lead to a mature process only through rapid process prototyping, incremental development, iterative refinement and the re-engineering of ad hoc process task instances and models (as, for example, used in modern IS design methods).

One important step in this process engineering life-cycle is process meta-modelling – constructing and refining process concept vocabulary and logic resource ontology for representing families of processes and process instances in terms of object classes, attributes, relations, constraints, control flow, rules and computational methods (Scacchi, 1998). Such computing environment for modelling, analysing and simulating may be knowledge-based (e.g. Scacchi, 1998 or "process handbook" by Malone et al., 1999). The "process compass" view in the process handbook defines four different relations for every specified activity: its sub-activities, "a part of", its specializations and its generalizations. Castano and Antonellis (1997) consider conceptual schemas defined according to the Entity-Relationship (ER) model in it extended version, including generalization and aggregation capabilities.

The notation used for graphical representation of enterprise model components is often simple but rich in details. The models used in the EM in its own notation can be converted to different diagrams in the UML notation without loss of information. For example, business process models can be represented by using activity and state diagrams in the UML and UML class and use case diagrams can represent actors and resources models. The EM process is typically iterative and the sub-models are developed in parallel threads. The stakeholders from the organisation and domain experts participate in varying degrees and their interests and knowledge need not be coherent.

2.3.4 Change management

Work processes are time to time revised in any organisation. The need for modification may arise from different reasons: environment (i.e. social environment or real life) change so that the existing rules and processes do not guarantee the fulfilment of goals and cause ill-performed behaviour with low efficiency. Rules and requirements (e.g. laws) are changed and respectively to this, the way of applicable behaviour (and guidelines for this) should be changed. Modifications can be also initiated from inside the organisation: internal goals are modified (and detailed processes for accomplishing the goals should be modified as well) or current efficiency does not satisfy expectations. Modifications of processes are in those cases unavoidable.

In a large organisation there are always some ongoing adjustments for ways to achieve the goals, or for goals themselves. These modifications are often done intuitively, the dynamics, possible modification of work processes and prognoses of results during day-to-day performance are usually not considered, thus the results depend heavily on subjective capabilities of leaders. In some cases global modifications (like transition of a monopolistic electrical supply company to a company, able to survive on open market) are pre-planned and supported by solid change management approaches like EKD-CMM (Barrios and Nurcan, 2004), but those are too voluminous, expensive and difficult to use for everyday changes.

It can be discussed whether design of an organisation is a one-time activity for establishing an especially designed structure in order to solve a certain task or whether organisational design is a continuous improvement effort with everyday changes in work processes and/or structure. The last approach in general is more accepted in he practice but organisational models do not often describe specific states in detail and redesign still begins from scratch.

As a rule, the organisation lacks the existing models of current processes and the work processes are to be re-specified prior to deciding what to modify in detail. Sometimes resulting process model and new or modified information system do not correspond to expectations – organisational efficiency is not improved. Inadequate modelling of work processes is often not the only reason. The main problem is that used process models do not reflect essential and important behavioural aspects: timing characteristics of processes and choices of autonomous actors in case of different emerging situations.

So and Durfee (1998) point out that organisational design should be an inherent part of a system, rather than something done by an external designer (i.e. higher authority). Organisational knowledge allows an actor to recognize its role and the roles of others in accomplishing collective goals.

Truly continuous improvement means that models are updated permanently (like Balanced Scorecard models are continuously used for general assessment). To achieve this, organisational modelling should be integrated into everyday activities and a change management procedure should be in place.

Change management is studied in different theories: many models of various types of organisations have been generated in organisational theory (Scott, 1992; Cohen, 1986; Fox, 1981); the contingency theory concentrates on organisational self-design (Lawrence and Lorsch, 1967). A school of thought in organisational theory is also the socio-technical systems perspective (Trist, 1981) that is influenced by human relations research. Related to coordination theory (Malone and Crowston, 1994), Malone et al. (1999) describe another approach to business process redesign and management tasks – an online "process handbook". Change management is also an important topic for many contemporary researches. It is based on Enterprise Modelling domain (for example EKD-CMM, Barrios and Nurcan, 2004) as well as other domains. Since the domain is evolving, there is no suitable modelling tool currently available that could persistently simulate the organisation at the same guaranteed depth and quality.

In recent years, reusability has become an important concern in the development lifecycle of software applications and of information systems. Business process re-engineering (BPR), for example, emphasizes re-use of information systems in accordance to modifications in processes (Pankovska and Sroka, 2002). In addition, several approaches have been illustrated in the literature to define and organize reusable components of information systems (software procedures). Different representation methods can be adopted for components in a library (e.g. F^3 methodology by Castano and Antonellis, 1997).

Resultant support to the organisation for modelling its operational behaviour and suggesting modification plans can be given by composing different scenarios that simulate the activities in a number of selected situations. Useful models should depart from and take into consideration different factors that cause the specific behaviour (goals, rules, actors' intentions and environment dynamics). Determination, which factors are crucial and which are not, depends in each case on the application domain and on specific issues of the modelled organisation: sometimes very qualified estimations of behaviour are needed and corresponding models should be more detailed, in other cases models should be simple and estimations can be more robust. It is important that a designer and the organisation should always be aware of what they want to accomplish by modelling, which models they therefore use and which models covers what.

From a number of **proprietary organisational change management methods** two models are considered here (that in turn are inspired and motivated by several methods used in the public sector): continuous planning and project management system of the Swedish Police (SNPB, 1999) and management cycle defined by the Scottish Police (SPC, 1996). The aim of those methods is to guide project management in an organisation (incl. define project stages, etc) and support the evaluation of the organisation's performance.

The Swedish Police use a continuous planning and project management system comprising seven stages (SNPB, 1999):

1. Problem analysis,
2. Planning,
3. Specification of the goal,
4. Composition of a guiding document (project or plan),
5. Implementation,
6. Checking and evaluation,
7. Dialogue with participated actors (briefing of results).

The management system can be used on all management levels – for strategic, tactical and operational planning and management.

The Scottish Police define the management action cycle as consisting of five stages (SPC, 1996):

1. Define objectives,
2. Plan activities,
3. Brief participating actors,
4. Monitor progress,
5. Evaluate results.

The cycle concludes with evaluation, which leads to re-defining of objectives and in this way closes the loop (an analogue to iterative spiral lifecycle in systems and software engineering).

The Swedish and Scottish police management approaches are in principle similar: both methods capture the most important work processes and organisational aspects – environment, organisation's structure, structural hierarchy and processes – and are able to deal with the hierarchy of business processes. Approaches allow the modelling of goal representation, task decomposition and allocation, decision-making, delegation and resource management. Also different roles in the organisation can be distinguished and initially planned interactions and co-operation mechanisms represented. The relation of the existing information systems to everyday life of an organisation can be considered by using the reviewed methods and process modelling for development activities is included naturally. The advantage of both approaches is that they consider the whole planning cycle from goal specification to the monitoring of implementation results. The methods allow easy straightforward interpretation of modelling results into day-to-day activities and support persistent adjustment of the model in accordance with the changes in the organisation. The disadvantage is that the methodologies are too descriptive. Computer-aided tools are not required for those methods and they do not exist.

The models support the observing of some aspects of employees' behaviour: they allow to consider employees' motivations and attitudes but they do not consider time issues, the planned behaviour and the real outcome are compared only afterwards and only formal communication is considered. In this way the approaches are similar to BSC and BV methods and have useful characteristics to take into account when elaborating new methodologies for organisation modelling. These methods concentrate on process change management but do not model processes at a detailed level nor consider time issues and interactions, they are not suitable for modelling emergent behaviour.

The Enterprise Knowledge Development (EKD; Bubenko, Persson and Stirna, 2001) is a change management EM method that prefers interactive approach to EM. EKD provides a systematic and controlled way of analysing, understanding, developing and documenting an enterprise and its components. The purpose of EKD is to provide a picture of how the enterprise functions currently, what the requirements and the reasons for change are, what alternatives could be devised to meet these requirements and what the criteria and arguments are for evaluating these alternatives.

The EKD approach is based on the premise that the key to successful change is knowledge shared by multiple stakeholders (in a broad sense) about the current and desired future states of the organisation and possible alternatives for the transformation process (Elektra, 2002). The EKD framework defines a set of models that describe the modelled enterprise (organisation), modelling project organisation as well as guidelines for project work in this modelling and for stakeholders' participation. The deliverables of the EKD process are a number of models that describe the organisation and its requirements from a number of interrelated perspectives.

Change management in EKD is comprised in using a business process model for the initial state, a business process model for future (desired) state and a change process model for modelling the change process. The modified version of EKD uses two types of design patterns: patterns dedicated to the modelling of the distribution of management and the human resource in the application domain (generic product patterns) and patterns tailored to change management of the distribution and the human resource management (generic change process patterns). Product patterns are goal patterns, business process patterns (actor-role, role-activity, object, rules) and information system patterns. Change process patterns consist, for example, of change intentions and intention resolution.

Table 2.5 The comparison of model requirements and characteristics of EKD

No	Requirement	EKD
1	Capturing the most important work processes and organisational aspects	Yes
2	Dealing with hierarchy and different levels of business processes	Yes
3	Modelling of goal representation, roles and resource management	Yes
4	Representing interactions and co-operation mechanisms	Yes
5	Relation of the existing information systems to everyday life of an organisation	Not explicitly, but can be considered in the description of processes
6	Process modelling for development activities	Yes
7	Straightforward interpretation of modelling results into day-to-day activities	Yes
8	Possibility of persistent adjustment of the model	Yes, time-consuming
9	Modelling temporal criteria	No
10	Modelling and comparing the planned behaviour and the real outcome	It is not stressed but can be implemented at the level of sub-models
11	Observing and checking of all the essential aspects that influence the organisational behaviour, including a generic model for analysing employees' behaviour	Cannot be specified explicitly on models but is considered implicitly – participants (e.g. domain experts) during the model development incorporate their ideas, beliefs and desires related to the expected result
12	Description of formal and informal communication	No

Table 2.6 Comparison of organisation analysis and modelling methods

No	Property	WFM	EKD
		8	9
1	The aim	Provide a common framework for the development of workflow management systems	Supportive method for preparing and implementing changes in an organisation by incorporating the knowledge of all participants
2	The scope	Organisations (any kind, mainly related to the integration with IT applications and tools)	Large project or organisation to be restructured
3	Necessity of a computer aided tool	No	No
4	Existing tools	Several	No (yes for other EM applications)
5	Strengths	Structured approach to business processes	Capturing of all important existing knowledge
6	Weaknesses	Purely descriptive, detailed specifications may have lot of other descriptions	Voluminous
7	Consideration of time issues	No (can be considered as usual resource)	If specially considered, no specific analysis model exists
8	Consideration of moti-vations, attitudes, etc	No (employees are considered as acting corresponding to the specification)	Included as opinions from different knowledge sources (domain experts, method providers)

The EKD design process consists of three stages: discovering (building the enterprise model), understanding (analysing the model and the needs for change) and designing (building the new model and implementing it). An example of the implementation of EKD is modification of organisations in electricity supply industry (ESI) sector (Elektra, 2002) for circumstances where the management change is not carried out in a controlled, structured manner but rather, the process is driven by previous experiences of consultants and is very much situation-dependent (Elektra, 2002). Such practice is typical for most real cases. The most important advantages of the EKD can be summarised as:

- The method considers different aspects of organisational change from the business objectives to the information system's functionality that support this change;
- Well-balanced attention is provided to the development of "products" as well as to the development "process"
- The availability of well-tested background support mechanisms (training material, case studies, experiences, CASE tools) makes EKD both teachable and applicable to industrial size applications.

The EKD approach is a remarkable elaboration of the EM approach. The most interesting and useful innovation is that the case is divided into "as is" and "to be" models plus a model of the change itself. An interesting feature is also the portability of stakeholders' knowledge. However, the EKD approach does not still satisfy all the requirements given in Section 2.1, (especially modelling temporal criteria and informal communication) as indicated in Table 2.5.

Table 2.6 continues the comparison given in Table 2.1 and Table 2.3 for Workflow system (WF MS) and EKD approach.

Inter-organisational services and business-to-business integration can be modelled by using similar approaches that are used for modelling processes and behaviour for a single organisation. Often models for inter-organisational services and business-to-business integration are more layered than intra-organisation models and the conceptual basis for business integration must be better defined (Wangler et al., 2001). Business-to-business integration will increase in the future since future organisations should find new ways of co-operation (e.g. agile virtual enterprises and dynamic alliances) in order to be more adaptable to customer needs and desires, co-operative, technology-centred and flexible to respond to changes in the environment (Xu et al., 2000).

2.4 SUITABILITY OF CONVENTIONAL APPROACHES: SUMMARY

The aims of the conventional modelling of human organisations can be summarised into two large categories:

1. Assessment of performance and quality measures in ongoing projects – to evaluate progress, the strengths and weaknesses of objects that are to be modified (e.g. organisations, human resources or technical solutions) or even inside an organisation as a whole (those methods were described in Section 2.2).
2. Modelling of planned work processes for modifying the existing processes in order to guarantee a more efficient fulfilment of goals (examples of such methods were described in Section 2.3). Process modelling approaches describe processes, their actors, sequence of processes and some additional parameters as well as devise suggested process schemas. In those methodologies the process of modifications and the implementation of modification inside an organisation is usually not described (except EKD that is strongly designed for change management).

As it was also briefly illustrated in Section 2.3, there are two usually quite isolated groups of approaches related to the modelling of an organisation:

1. Modellers and developers of organisations,

2. Developers of information systems and other computer applications,
3. Members of the organisation (actors).

According to these two different approaches the implemented methods are also different and developed separately: methods for describing organisations (examples were given in this chapter) and methods for developing computer applications (e.g. entity relationship diagrams, a myriad of object-oriented methods and, proceeded from this, UML ideology and tools). Both of those directions have influenced each other from time to time and the usage of methods overlaps partially. The aims are different in both modelling directions: for example dynamically changing goals, goal-influenced structure, non-deterministic characteristics due to the stronger influence of human and market forces, etc for the modelling of organisations. Methods for devising information systems and developing computer applications are evolving in the same direction that has already created similarities in the used methods – e.g. multi-agent systems, real-time systems, ubiquitous and pervasive computing systems, interactive and proactive computing. In some current methods there are already similarities with the approaches taken in some other group, for example EM and EKD-l share several very similar characteristics with the UML. This gives the idea to continue with the same trend for developing new methods.

The given review of the conventional modelling methods in this chapter has shown that although there are multiple different analysis and modelling techniques for describing and planning the performance and activity of an organisation, they are not suitable for modelling time-dependent emergent behaviour. The implemented methods do not consider the motivations and possible behaviour of employees and identify the behaviour of employees as a planned behaviour for the corresponding actor or role. Also most of the approaches do not regard time issues to be important and the timing analysis is quite weak. Multiple functions and processes inside an organisation may require the same collective resources (e.g. actors, equipment, knowledge, financial assets) simultaneously and the same persons or resources may be fully or partially loaded by other functions at the same time. During each modelling session normally one specific task or function domain is reviewed and the resources are considered as fully available for the tasks under the analysis. Therefore the outcome of the organisation cannot often be estimated correctly since modellers do not often consider those concurrently activated (and thus competitive) goals and tasks. Additionally, some actors (humans) may have their own motivations and necessary recreation time (that is used during the regular working hours). This is similar to lead-time required to readjust devices in flexible manufacturing and should be considered when the resource is used continuously for several processes or tasks (e.g. employee has multiple tasks).

Therefore the current research takes the aims of conventional modelling methods and tries to extend them in the following areas:

1. To model timing criteria of processes and describe in more detail how process interactions and timing is done.

2. To concentrate on how the choices are actually made by actors – the possibilities of the actor to choose another behaviour or to perform an action a little bit later, the following consequences (in terms of processes) in each of such cases.
3. To model interactions between actors in more detail.
4. How to visualise processes and behaviour so that they are understandable also to non-IT-specialists.
5. How to assist the planning of process modifications and present possible results of modifications.
6. How to assist during modifications by providing a roadmap with the estimation of the expected behaviour of the organisation.
7. How to integrate sub-models since a single model cannot encapsulate all organisational characteristics.

Some ideas for devising a suitable approach can be derived from the review and conclusions: to use the principles of EKD in conjunctions with more detailed (specified on the computer and better formalised) results from computer applications development domain (e.g. paradigms for description and tools for analysis). From the presented selection of techniques a hypothetical combination of methods can be suggested for integrated planning and analysing the performance of an organisation: the BSC can be used for defining the goals and primary activities of an organisation, the Best Value or a similar method can be implemented for evaluation and for the comparison of results with other organisations. A more detailed planning and designing of necessary changes can be done using any EM method, e.g. EKD. SWOT or other analysis techniques can be used for solving specific cases.

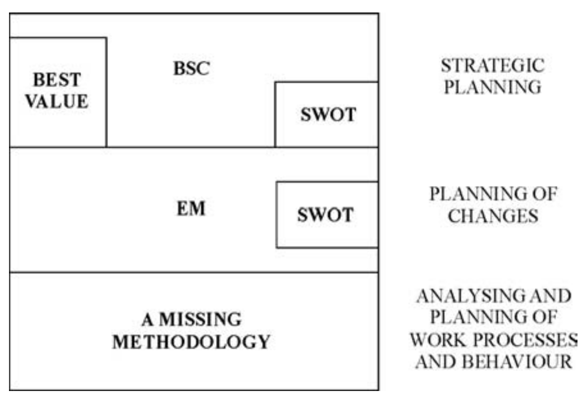

Figure 2.1 A hypothetical combination of existing and missing modelling methods for analysing and modelling of an organisation and modifications in its performance

Such a hypothetical combination of conventional modelling approaches still leaves a gap between the general planning of modifications and the concrete detailed analysis and simulations of work processes. Therefore such an integrated approach for process modelling and planning of modifications should be complemented by temporal analysis and modelling emergent behaviour of employees, as stated in Section 1.2, shown in the current chapter and illustrated in Figure 2.1. The grounds for missing modelling methodology will be reviewed in the next chapters hereinafter.

3 TOOLS FOR MODELLING EMERGENT BEHAVIOUR

The previous chapter surveyed some widely used methods for modelling organisations and concluded that they focused mostly on defining operational goals and on methods for assessing the level of achieving those goals. This is an appropriate starting point for the formalised study of organisations but it leaves the control and modification of processes pending upon the intuition and subjective beliefs of managers and their consultants.

The next step in the formal study of organisations should go deeper into the factors that influence the achievability of goals. As a natural start, one should pay more attention to the analysis of lower level processes within organisations. The essence of processes and how they influence the overall behaviour are not thoroughly studied in the context of organisations. However, a lot of relevant research has been done in the context of devising information systems and the distributed artificial intelligence based systems. Some of the results have been successfully applied in practice – business information systems, computer control of technological processes, etc. Some other results have invoked theoretical interest in revising the foundations of conventional computing (e.g. Wegner and Goldin, 2000; Wegner and Eberbach, 2004).

This chapter discusses the possibility of applying the developed and applied ideas in:

- The Unified Modelling Language (the UML; Booch, Rumbaugh and Jacobsson, 1999) that has some common features with Enterprise Modelling methods discussed in the previous chapter (e.g. multiple views to organisation),
- The Q-model (Motus and Rodd, 1994; Selic and Motus, 2003) that represents a model for multi-stream interaction machine, dedicated to time-sensitive computations,
- Research in agent technologies and multi-agent systems (Ferber, 1999; Wooldridge, 2002) that permits the modelling of behaviour and interactions of multiple actors,

in the study of organisations.

The functioning of a real organisation cannot be adequately described in algorithm-centred model of computation. At the time of modelling we do not have complete information as to the grounds for behaviour of each particular actor. As the environment is dynamic, we cannot either estimate all possible emerging situations. In real situations the activities of actors depend also on the fact when particular information has arrived or situation emerged. Consequently we are not able to describe algorithmically the expected sequence of actions of a human actor. As a result, we cannot describe the expected behaviour of the whole organisation as a single algorithm.

Actors' choices depend on their preferences, emerging situations and received information. This means that an actor is similar to interactive system – an output of an interactive system depends on its current state and previous input history. Organisations' activity is carried out in multiple simultaneous streams that interact freely, not following the non-interference principle conventionally assumed in concurrent processing (Apt and Olderog, 1997; Rakitin, 2001). It is not an exception that some concurrently performing activity streams stop occasionally their autonomous operation and consult (or ask advice) each other, or from an external source in the environment of the organisation.

In the current thesis the suitable approach for modelling emergent behaviour is the interactive systems approach. Actors are considered as agents and the organisation as a multi-agent system.

The behaviour of an interactive system leads to the so-called super-Turing computation (Wegner and Eberbach, 2004) – agents can be computationally analysed only by using a model of super-Turing computation. Models of interactive computing and agent technology in particular seem to be quite suitable for modelling organisations and studying their behaviour. Emerging behaviour is a characteristic phenomenon of interactive computing that also characterises behaviours observed in human organisations. Interaction-based models of computing exceed algorithmic models in formal description and analysis power (Wegner, 1997).

Besides emergent behaviour and multi-stream computation, interactive computation may possess and exhibit history-dependence in general and dependence on quantitative time constraints imposed upon the behaviour of computing agents in particular (Wegner and Eberbach, 2004). Stream input-output and history-dependent behaviour of the structural units that represent actors are important for modelling characteristic features in the actual behaviour of human organisations. Usually the communication in an organisation is on-going via multiple simultaneous communication channels and the decision about how to respond (what behaviour to choose) in a particular situation depends not only on the current input (the message last received) and on the present state but also on previously received messages, the context of this particular situation, the available knowledge about previous situations and on the belief about future developments.

No widely accepted models of interactive computations are available for the time being. However, several concepts and prototype models have been suggested. In general, it has been accepted that a building block (component) for building an interactive computing system is not an algorithm but rather a set of interacting, repeatedly activated algorithms (Motus and Rodd, 1994; Motus and Meriste, 2001).

Both, the UML and the Q-model are based on interaction-centred model of computations. In the context of organisation modelling it seems natural that the

UML and related methods capable of handling interactive computations are chosen as the departing point for describing and analysing the work-processes and emergent behaviour in organisations. The first section of this chapter (Section 3.1) discusses the UML as a widely used powerful and standard technique for modelling large applications and for generating the process code directly from the model. Section 3.2 describes the use of the Q-model for analysing time behaviour in systems based on interactive computation. The chapter proceeds with introducing multi-agent approach (Section 3.3) and reviewing how this approach can be implemented for modelling human organisations (Section 3.4). The chapter concludes with suggestions for implementing reviewed methods for modelling emergent behaviour in organisations.

3.1 THE UNIFIED MODELLING LANGUAGE

This thesis focuses on developing methods for building advanced models of organisations that support step-by-step modification of behaviour. A suitable method would help to take corresponding decisions by analytical or simulation studies, keeping in mind emergent behaviour and the multi-agency of an organisation. The Unified Modelling Language (UML; Booch, Rumbaugh and Jacobson, 1999) meets those requirements partly – it enables to describe the architecture, work processes and their interactions in an organisation but is not too supportive in prototyping and in formal behavioural analysis of the organisation as a whole. Therefore the UML is used as one of the three cornerstones of the methodology suggested in this thesis.

The UML is a graphical language for visualizing, specifying, constructing and documenting the artefacts of software-intensive systems (Booch, Rumbaugh and Jacobson, 1999). Although the UML is initially designed for complex software-intensive systems, it is also suitable for modelling processes in a wide spectre of organisations. It is a widely used standard in industry and academic communities helping to describe, model, visualize and analyse models of a large class of systems. Object Management Group (OMG) adopted the first version of the UML in 1997 and OMG Revision Task Force (see OMG, 2005) releases its new versions (currently the latest formal version in June 2005 is UML 1.5, see OMG, 2003a and the final adopted specification of UML 2.0 is released).

Modelling and creating a complex software-intensive system assumes that the system can be viewed from different perspectives since different persons – end users, analysts, developers and managers are concerned with different aspects of the system. The descriptions of different perspectives are combined in the UML by defining process view, communication view, design view, implementation view, deployment view and use-case view. The use-case view of a system encompasses the use-cases that describe the behaviour of the system as seen by its end users, analysts and testers.

There are structural and behavioural diagrams in the UML. An example of the structural diagram is a class diagram that shows a static structure of an organisation as a set of classes and interfaces, together with their collaboration structure and relationships. A class (a description of a set of objects that share the same attributes, operations, relationships and semantics) implements one or more interfaces – collections of operations that are used to specify a service of a class or a component. Each interface may represent a role that the object plays. Role defines the behaviour of an entity in a particular context or situation. For instance, an actor represents a role or a set of roles that a human or another component plays in a system when interacting with use-cases.

Most often implemented behavioural diagrams are use case diagram, sequence diagram and activity diagram. Use case diagram, especially important in organizing and modelling the behaviour of a system, shows a static view: a set of use-cases, actors (a special kind of class) and their relationships. Relationships in the UML can be represented as dependencies (relationships among classes), generalisations (link classes to their specialisations) and associations (structural relationships among objects). The dynamic aspects are captured in sequence diagrams, state-chart diagrams and activity diagrams. A sequence diagram is an interaction diagram that emphasizes the timed ordering of messages sent and received by different objects. An activity diagram shows the sequential or branching flow of control from activity to activity. Activity diagrams are especially important in modelling the function of a system.

The UML is visual, logical and it has a number of software environments that support the whole modelling process (e.g. Rational Rose, see Rational, 2003) or just the supporting design of diagrams (e.g. Visio, see Visio, 2003). The UML is also strong in (according to Selic, 2003) emphasizing semantics (as opposed to notation) – it is a model-based approach and has detailed semantic specifications for some parts of the model. In addition, it has higher-level abstractions for the most currently used object-oriented language technologies (e.g. state-charts and activity diagrams, support for specifying inter-object behaviour as interactions); it is extensible and also customisable. The UML-based software development environments (CASE environments) support to some extent also source code generation. Moreover, the UML allows to represent different parts of the model in one "language" (Booch, Rumbaugh and Jacobson, 1999). Sometimes it is still useful to represent some specific characteristics of the model or behaviour of the organisation in a formal theory, supported by modelling environments to emphasize and deeply analyse those aspects (see model processors, for instance in UML profile for Schedulability, Performance and Time Specifications, OMG, 2003b, Motus and Selic, 2003). It is often possible to find there how (automatically) to transform those descriptions into the UML. In this way the UML allows to analyse and to verify (to some extent) the models for the system's behaviour and properties before the code generation.

The UML is related to the unified software development process (the unified process, UP; Arlow and Neustadt, 2002). This is an iterative process that emphasises the creation and maintenance of models for the system under development. The development in the UP is architecture-centric and activities are use-case driven. Each model in the UP is object-oriented. The UP is a configurable process where different things can be adjusted according to the needs of a specific organisation.

In spite of some current weaknesses, the UML seems to be a promising tool for the modelling of organisations. Some further UML related developments that are particularly suitable in the context of this thesis are discussed hereunder.

3.1.1 Modelling of resources, time and timing related analysis

Specifying time constraints, schedulability and performance analysis are the focus of the UML Profile for Schedulability, Performance and Time (see OMG, 2003b). The general structure of the profile is modular. Three quantitative model analysis sub-profiles (resource analysis, performance analysis and schedulability analysis) are all based on the general resource-modelling framework (Selic and Motus 2003). The modular structure allows users to apply only a related sub-set of those sub-profiles.

Time has to be considered in several areas of modelling, especially in real-time systems, but also increasingly in studying the functioning of organisations. In the real-time UML profile metric time is considered (OMG, 2002). The profile distinguishes between continuous time (as a continuous and unbounded progression of time instants) and discrete simulated time that may be represented simultaneously in a multitude of forms, including those where time does not necessarily increase monotonically.

Time sensitive software and its models often require formal or semi-formal quantitative analysis. Automatic analysis of UML models becomes possible by using formal models that are not part of the UML environment but are derived directly from UML models as defined in the real-time UML profile. Model processors are the concept on which the performance and schedulability analysis tools have been built (OMG, 2005) and possibilities for building interaction analysis tool are being studied (Selic and Motus, 2003; Motus, Vingerhoeds and Meriste, 2005). A common characteristic feature of real-time systems and organisations is forced concurrency (Motus and Rodd, 1994), it is also called truly concurrency (Wegner, 1998) processing of information. Full timing analysis can be performed for conventional non-interactive concurrency (that can be described algorithmically) by applying a single, strictly increasing metric time for a set of concurrently executing programs. At the same time several timing properties in the case of forced concurrency assume a more sophisticated time model with multiple increasing metric times (see, for instance, Motus, 2003; Selic and Motus, 2003).

The time model adopted by OMG for the UML real-time profile is flexible enough to support both algorithmic and interactive computing paradigms. For instance, model processor for schedulability and model processor for performance are based on the conventional (single increasing time) time model, whereas model processor being developed for interaction analysis is based on multiple (independent) metric times and can analyse forced concurrency that is truly interactive (see Section 3.2 of this thesis for additional details).

The modelling of resources is also fundamental to real-time UML profile and used as the basis for most other packages. The general resource model defines a common terminology and conceptual framework for real-time model analysis methods. The quality of service characteristics is a part of the resource model. The model analysis problem is then reduced to comparing the demand (the required quality of service) against the offered quality of service of the resources. Causality modelling package captures the essentials of the cause-effect chains and is used as a basis for the dynamic modelling associated with the profile. A fundamental concept in the causality model is the notion of an event occurrence, corresponding to an instance of the UML event notion. In the more complicated cases where the causal relations are incompletely known, sophisticated time models are used to approximate incompletely known causal relations. In those cases the interaction analysis reveals the feasibility of the resource model (see again Section 3.2 of the thesis). A resource usage describes how clients use resources and their services. It corresponds closely to a use-case instance. A resource usage can be either static or dynamic. The dynamic usage model applies in situations that concern the order and time of the usage of resources. The usage is then represented by a scenario instance (i.e. an ordered series of steps called action executions).

3.1.2 Model driven development and the UML 2.0

The UML 2.0 is the first major revision to the standard since its inception in 1997. The UML 2.0 standard revises some aspects of UML infrastructure, superstructure, Object Constraint Language and UML Diagram Interchange. The UML 2.0 adds two fundamental capabilities for modelling software architectures (Selic, 2003): structured classes for modelling structural aspects and (complex) interactions for modelling behaviour. An interaction is a specification of how stimuli are sent between instances to perform a specific task (the interaction is defined in the context of collaboration) (OMG, 2003d). In UML 2.0 behaviour specifies the computation that generates the effects of the behavioural feature. The description of behaviour can take a number of forms, including interaction and procedure (a set of actions). Interaction is specified through ports. A scenario is a specific sequence of actions that illustrates behaviours, interactions or the execution of a use case instance (OMG, 2003d). The new standard is designed to support model-driven development paradigm, for instance Model-Driven Architecture (MDA; Kleppe, Warmer and Bast, 2003; Raistrick, Francis, Wright, 2004), as defined by the Object Management Group (OMG).

A system, its model and modelling notation can together be represented in a four-layer hierarchy (OMG, 2003d; Denno et al, 2001; Selic, 2003):

1. Real objects (e.g. computer memory, run-time environment) form the bottom level;
2. The application model level (i.e. models of the system being developed);
3. The meta-model (i.e. architecture of the application model, e.g. the UML);
4. The topmost level is the level of the meta-meta-model (i.e. architecture of the meta-model, e.g. Meta-Object Facility, MOF).

The previous versions of the UML concentrated more on the model level, while the UML 2.0 covers also the upper level in this hierarchy. The Model Driven Architecture (MDA) paradigm and Model Driven Development (MDD) approach assume and require that most of the design work is to take place at the model level. The development of an application system proceeds from platform independent models to platform specific models with interim (formal and informal) analysis and terminates with (automatic) code generation followed by testing and actual operation of the application system. The modifications resulting from the interim analysis, tests and actual operation are fed back to the platform specific models and platform independent model for additional analysis and/or future reuse in other applications.

The need for the formal analysis of the UML models increases remarkably in connection with the MDA and MDD application. Today software and systems engineers use widely the so-called model-checking method – based on a design or implementation, a model that describes it (typically in the form of a finite state machine) is developed. An approximate verification or validation or testing is carried out on this model and the resulting modifications and conclusions are assigned to the design and/or implementation that was under analysis. The concept of model processors introduced in the UML real-time profile (OMG, 2003b) extends and generalises slightly the model checking concept – by simplifying the build-up of those formal models by enhancing the presentation of the quality of service related information in the UML models, by allowing a wider class of formal models in order to cover also interactive computation and by improving the feedback of analysis results to the UML models.

A general term model processing encompasses the notion of a primary model (e.g. UML model) that is being analysed for specific properties not analysable within that model, a secondary model built on the information from the primary model, typically based on a different formalism that is suitable for analysing those specific properties and equipped with analysing tools (e.g. the Q-model, see Section 3.2) and the transformations (preferably automatic) from the primary model to the secondary model and back from the secondary model to the primary model. An example of model processor is a processor for schedulability analysis that accepts a UML model and analyses the schedulability of that model (analysing part) or generates appropriate modifications that enable schedulability (synthesising part) and feeds those modifications back to the UML model. Another example of a model processor is a processor for performance analysis (see OMG 2003b).

The goals set to MDA and MDD methodology and the toolset for its implementation are very close to the goals set to the organisation models – design, analyse, plan and assess the organisation's behaviour before the modifications are actually implemented. If the analysis of the design or implementation plans reveals discrepancies, the design or plans should be re-engineered before the actual changes in the organisation are introduced. Quite often minor modifications are made in an organisation without preliminary modelling – in that case those modifications should be introduced to the models (as part of the model re-engineering process applied in MDA and MDD).

3.2 THE Q-MODEL

Once established the organisations operate continuously until they exist. The models that describe the behaviour of organisations operate as non-terminating programs. Real-time systems and a rapidly increasing number of new computer applications operate also as non-terminating programs. The behaviour of non-terminating programs cannot be described within the theory of algorithms (based on Church –Turing thesis), such programs can be described and analysed by using interaction-centred model of computation (Wegner, 1997). As stated in Section 3.1, the UML is one of the models that is able to encapsulate interaction-centred computations, although it is not sufficiently formal for handling quantitative properties of behaviour. For that purpose the concept of model processors has been developed.

The Q-model is one of the few formalisms that handles time-sensitive interaction-centred computation (another is, for example, Caspi and Halbwachs, 1986). It has the description and analysis power of a multi-stream interaction machine and supports automatic prototype generation at the very early stages of systems' development.

The first published time-sensitive model of interactive computations was focused on timing analysis of real-time software (Quirk and Gilbert, 1977). The model has been named the Q-model and developed further (Motus and Rodd, 1994), implemented in CASE tool Limits and linked to OMT and UML based software engineering environments (see, for instance, Motus and Naks, 1998). The Q-model has been applied for timing analysis in an industrial diagnostic tool BRIDGE (Naks and Motus 2001). The underlying time model (Motus 2003) is based on the simultaneous use of multiple time concepts (fully reversible, strictly increasing and relative) and is in rather good concordance with the OMG adopted time model (OMG 2002a). The Q-model seems a promising candidate for describing and analysing (time-sensitive) processes in an organisation and for checking the coherence of time constraints imposed on inter-organisational and intra-organisational interactions and therefore is taken as a second of the three cornerstones of the methodology suggested in this thesis.

The Q-model is especially suited for modelling, animation and timing analysis of interaction at the early stages of system development – e.g. user requirements, specifications – but can also be applied at the design and implementation stages. Its building blocks are quite autonomous functional units (processes), interaction of those functional units is organised via a concept of intelligent interfaces (channels). Each process can be a stream processor – a repeatedly activated (by its own time-set, or by its interaction partners), terminating, time and history sensitive mapping of its input streams into output streams. Channels enable time selective communication – the consumer can subscribe to messages of certain age – and also provide means for coordinating the behaviour of interacting processes. This provides a sound mathematical basis for formal analysis of many behavioural properties and for automatic generation of an executable prototype for the experimental study of the model.

The Q-model is based on the idea that an activity of a system can be represented as a set of interconnected processes that are executed in certain time instances. In this way a behaviour of an organisation is seen not as one single process but as an aggregated whole of multiple, repeatedly activating and finishing processes. The representation of processes in the Q-model is normally (mostly) used for process control environments. Therefore this approach is mainly used for industrial domains e.g. chemical batch processes, water engineering, etc. This is especially popular because it has sound mathematical basis related to the approach and possibilities for visually well-describing representation. The Q-model can also be efficiently used for describing other processes and for other application domains. Although still not used, the Q-model is also suitable for the presentation and analysis of processes in modern, computerised human organisations where specified work processes and information exchange play an important role, even if the latter is not explicitly shown.

The Q-model is suitable to be used for modelling timing issues of processes and analysing resource conflicts since those characteristics are an essential part of the Q-model ideology. This is the most suitable notation to be used until the UML does not offer such possibilities. The Q-model may loose its importance after the UML 2.0 will be introduced since the UML 2.0 supports related issues (hopefully there will also be several real modelling software products available) and UML Profile for Schedulability, Performance and Time (as considered in Section 3.1.1) already incorporates many issues into the concept of modelling time and performance used in the Q-model.

The Q-model as the only modelling tool is not sufficient for multi-functional organisations. The model of real organisations in the presented circumstances may quickly become too complex to be handled and understood by humans. This reduces the efficiency of modelling the organisation’s actual behaviour.

The Q-model does either not support the modelling of different actors as personalities.

The Q-model analysis and simulation can be implemented in Limits PC CASE Tool (Limits, 1998). The tool can be used for composing Q-model diagrams and timing analyses of the organisation. All corresponding models are stored as Limits CASE Tool projects.

In Limits PC Case Tool the processes can be implemented as common processes or selector processes (input, output or input-output selector). Each process must have a name and a scope. Data are gathered from input port(s) and results are produced to output port(s) of the process. For a common process, the following parameters can be used: activation can be determined either by giving start period (in this case period length, deviation and simulation mode can be given) or an explicit time set for activations. An equivalence interval can be determined. A common process uses data from all its input ports and produces data to all its output ports. For input and/or output selector process there is also the possibility to determine scenarios for consuming or producing data from and/or to certain ports only. For this the combinations of ports and the probability of the activation of those ports can be determined.

Data and control signals between ports are transmitted by using channels. A channel has a name, scope, type and channel function. There are four types of channels used in the Q-model: a null channel, a synchronous channel, a semi-synchronous channel (also called petri channel) and an asynchronous channel. The channel function determines the earliest and latest data generations used in the process. Null channels do not have the channel function since no data are transmitted except the control signal. Synchronous channels are used to execute two or more processes simultaneously together with passing the initial data. In semi-synchronous channels the control signal and data are passed when the preceding process finishes. This is used to implement a sequential control between different processes. In asynchronous channels only data are passed and those cannot be used for determining the control signals. In practice semi-synchronous channels and asynchronous channels are used most.

According to the user manual (Limits, 1998) the Q-model diagram may be imported also from the file in text format. This allows some other software to produce its results into a file with a suitable format and it may therefore be an alternative to the drawing of diagrams by implementing the user interface.

A Q-model diagram is given as an example in Figure 3.1. This picture illustrates graphical notation used in the Q-model processor for analysing timing criteria. In the figure circles represent processes and arrows represent channels. Each process has a name and a set of input and output ports. Each channel has a channel type. Other parameters are implicitly described in the LIMITS, implementing a special user interface. A UML activity diagram is more suitable for human users to understand process characteristics.

The result of the Q-model analysis is a timing diagram. The timing diagram represents graphically the execution, data consumption and termination of processes. The timing diagram also indicates if some processes are executed parallelly in several copies or if the execution of some process is suppressed because of process equivelence interval. A timing diagram can be analysed by a human user in order to evaluate the results of process behaviour and detect collisions. For example, the Q-model diagram represented in Figure 3.1 is represented in the form of a timing diagram given in Figure 3.2.

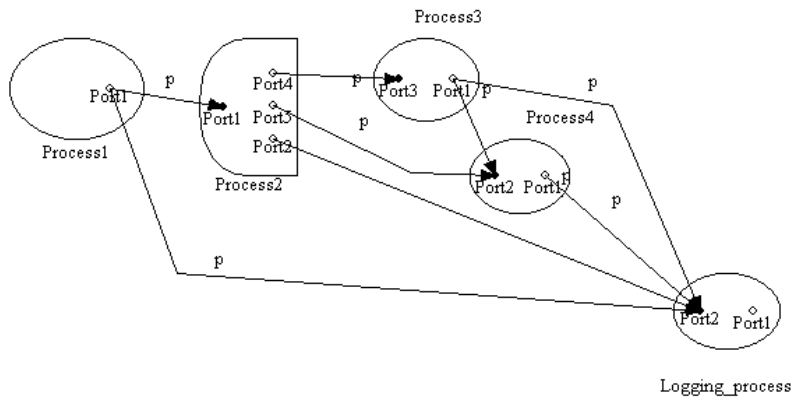

Figure 3.1 An example of the initial Q-model diagram

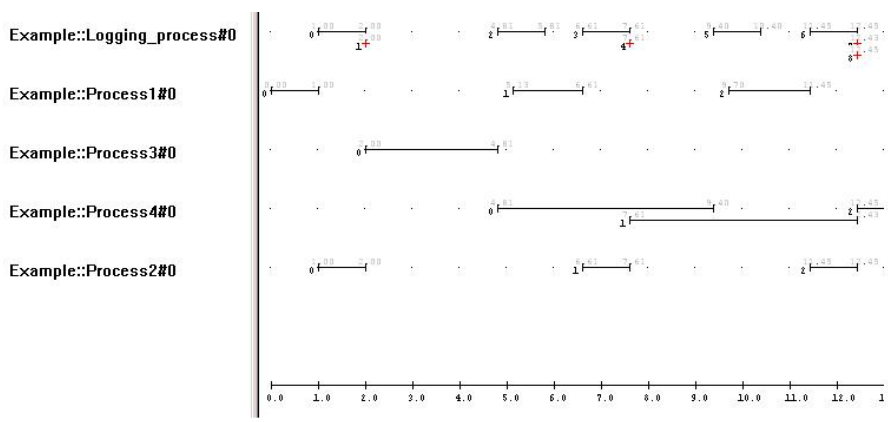

Figure 3.2 An example of timing diagram as a result of the Q-model analysis

3.3 MULTI-AGENT SYSTEMS

The methods and tools considered so far in the present thesis have focused on determining and/or modifying the goals, developing methods for assessing the efficiency and obtaining deterministic description on improving the work-processes in an organisation. Although important and extremely useful, those methods and tools neglect the subjective factors (introduced by humans), indeterminacy introduced by dynamic interactions between autonomous structural units and the influence of dynamically changing environment of an organisation. The human factors (e.g. intelligent decision-making) and dynamic interactions combined with the autonomy of structural units amplify the influence of emergent behaviour (i.e. the behaviour that cannot be fully determined by the static structure of an organisation) to the overall behavioural pattern of an organisation.

The progress made by distributed artificial intelligence in handling the cooperation of proactive, autonomous components that may have their own goals and can perceive their environment has been remarkable (Prietula et al., 1998; Wooldridge, 2002; Ferber, 1999). The corresponding methods – related to agents and multi-agent systems – deserve serious attention in studying the emergent behaviour of an organisation. This thesis combines agents with the Q-model and the UML-based methodology for the advanced modelling of organisations. The hypothesis is that such models enable to study traditional deterministic behaviour as well as emergent, time-sensitive behaviour of organisations.

This section presents basic requirements for socially intelligent agents, reviews some known agent applications and approaches (including the specification of multi-agent systems in the UML notation) and illustrates the modelling of human organisations as multi-agent systems with considering different behavioural aspects like task decomposition, resource allocation, agents' motivation, co-operation and communication.

3.3.1 Agents and social intelligence

The term *agent* usually means a specific piece of intelligent software or robotics– an agent is a persistent computational system, which (Ferber, 1999):

- must be able to work autonomously without human intervention;
- has goals, sensors and effectors (is capable of perceiving its environment to a limited extent and capable of acting in its environment);
- is able to communicate with the other agents of its own choice;
- possesses its own resources and skills and is able to offer services to the other agents of its own choice;
- is driven by tendencies and anticipation;
- and decides autonomously which actions to take in the current situation so as to maximize progress towards its (time-varying) goals (Maes, 1997).

An agent is proactive – is able to initiate its behaviour according to its own state and goals, as seen from the list above.

Agents can be categorised as software agents – computed systems to which one can delegate tasks (MIT SAG, 2001) – or physical agents. Software agents differ from conventional software in the way that they are semi-autonomous, proactive and often adaptive. Software agents are widely used in several software implementations (e.g. MIT SAG, 2001; Barber et al., 2004) like information filtering agents (e.g. Klush, 1998; Zou and Wang, 2000), agents as navigation guides (e.g. Kido, 2001), buying and selling agents, recommending agents and matchmaking agents. Physical agents are situated in or model the real world or its imitation (i.e. computer models of reality) and have to take into consideration the actual environment in which it operates. FIPA (see FIPA, 2005a) establishes common standards for physical agent architecture and interaction protocols.

Agents' internal architecture (this is the term generally employed to describe the internal organisation of an agent) often depends on agent type, design goals and approach (Ferber, 1999). Two types of agents can be distinguished in their way of acting and conceiving of the world. The *cognitive* agents use for reasoning the symbolic and explicit representation of the world, the *reactive agents*' behaviour is based on sub-symbolic level, for example, triggered by their sensory-motor system. Cognitive systems with modular horizontal architecture or blackboard architecture are used more in robotics in industrial manufacture (Giraud, 1984; Rencken et al., 1993) and by many authors from distributed artificial intelligence field (e.g. Barber et al, 2004; Fayek, Liscano and Karam, 1993; and Engelmore and Morgan, 1988). Reactive systems are more popular in autonomous mobile robotics (e.g. Brooks, 1986; Balkenius, 1995; Coderre, 1988) or in other applications with other architecture types (e.g. subsumption architecture, production rules architecture by Fox et al., 1998, dynamic system architecture based on stimuli-command relationship, architecture for the creation of cooperative autonomous robot by Steels, 1994 or connectionist architecture, also called neural networks). For software agents both cognitive and reactive architectures can be used.

By general definition, all agents are driven by tendencies. Some approaches consider that tendencies are coming from the environment, while others assume that tendencies are expressed within the agents. The behaviour of agents can thus be named as *reflex behaviour* and *teleonomic behaviour* respectively. As a result, four agent types can be determined, depending on their behaviour and methods for conceiving the world (Ferber, 1999). *Cognitive teleonomic* agents (also called intentional or rational) have their own explicit goals that motivate their actions. *Reflex cognitive* agents (or module-based agents) respond to questions or accomplish certain tasks that other agents command them to do. Most *reactive* agents are *teleonomic* (therefore called drive-based agents) and are usually directed by motivation mechanisms to accomplish an internal task, for example maintaining their energy level or any other goal defined by the designer. Another small group of agents, *reflex reactive* agents (tropistic agents), are made to respond only to stimuli form the environment and their behaviour depends on the local state of their environment (e.g. O-ANTS, The Omega Group, 2001 or Dorigo, Manezzo and Colorni, 1996).

Kido (2001) believes that future robots and software agents will need to interact with each other and with humans, using types of social expertise that may begin to match human social competence. Therefore the promising approach is to explore and understand the principles of human-style social intelligence and to utilize the findings for designing intelligent agents. The agents suitable for the modelling of human organisations should be socially intelligent, in other words the agent (in addition to general characteristics, given in the beginning of the section):

- achieves its objectives by interacting with other autonomous entities (Hogg and Jennings, 2001) and if necessary by taking the initiative (Nguyen, 1997);
- must be able to enlarge the specific instructions and to use symbolic abstraction for solving problems;
- is able change behaviour according to accumulated knowledge (Thompson and Coovert, 2001);
- possesses the ability to distinguish between potential partners and enemies (Edmonds, 1997);
- is able to operate in a resource-bounded manner (Hogg and Jennings, 2001);
- has to respond to dynamic changes in the environment in time (Amin and Ballard, 2000; Barber et al., 2004) and has to manage goal constraints and change focus for multiple goals (e.g. deadlines and priorities).

The theoretical background for socially intelligent agents is very much related to the organisational theory and co-ordination theory (Rognin, 1997). Much of the research is related to cognitive engineering (e.g. GRIC, 2005) and computer supported co-operative work (CSCW), where intelligent agents are expected to play an important role in the future.

Organisations can be modelled as populations of interacting agents in a given structure (e.g. Edmonds, 1997) or similarly to computer systems to which one can delegate tasks (e.g. MIT SAG, 2001). Families of agents can be used for evaluating the effectiveness of social behaviour (Fasli, 2003) and decision-making (e.g. Hogg and Jennings, 2001, Dugdale and Pavard, 2000; Parunac and Odell, 2001). When a real human organisation is modelled, cognitive agents with teleonomic behaviour seem to be the most suitable (Edmonds, 1997, Fougères, 2002). When models are simplified and only deterministic work processes are modelled, internal structure of an agent and its behaviour description can be simplified and other types of behaviour used. For example, agents used for the illustration in the present thesis (in Chapter 5) are combined from competitive task architecture (the choice for the behaviour is made on the message received) and modular horizontal architecture (modules inside behaviour) with reflex cognitive or reflex reactive behaviour. The properties of an agent that describes a unit or the whole organisation can be considered similarly to other agents and several agent types can be used.

3.3.2 *Multi-agent approaches*

A **multi-agent system** (MAS) consists of an assembly of agents and an assembly of relations. Ferber (1999) considers the environment also as a part of MAS but opinions of other authors do not agree with this. In the present thesis the environment is not considered as being part of MAS, but it is still emphasized that agents (MAS) are always situated in an environment. In this way a simple MAS consists of at least two agents working in the same environment, communicating with each other or otherwise influencing each other's performance. The compatibility of goals, sufficiency of resources, abilities of agents in relation to tasks and trustworthiness determine the type of interaction and relationship between agents (Ferber, 1999; Carley and Prietula, 1998). A sophisticated multi-agent system can model antagonistic relations between different teams where each team is a small member-friendly organisation (e.g. WebBots, Carley and Prietula, 1998). Most of such examples are related to the military domain since organisations in this domain have to deal with rapidly emerging events with high uncertainty and potentially catastrophic impacts (Kang, Waisel and Wallace, 1998). Complex multi-agent systems are also used in manufacturing (Sacile et al., 2000; Shen and Norrie, 1999) and in workflow management (WFMC, 1995).

Several modelling approaches are devised for specifying MAS (e.g. described in Wooldridge, 2002; Wille, Dumke and Stojanov, 2002; Wood and Deloach, 2000), like the AAII methodology (Kinny and Georgeff, 1997), the Gaia methodology (Wooldridge, Jennings and Kinny, 2000), MaSE methodology (Wood and DeLoach, 2000), the Cassiopeia methodology (Collinot et al., 1996; Drogoul and Zucker, 1998).

The AAII approach. Kinny and Georgeff (1997) describe a multi-agent systems architecture that is based on BDI (belief-desire-intention, Rao and Georgeff, 1995) architecture. The description of an agent system from the external viewpoint is encapsulated in two models: agent model and interaction model. An agent model has two components: an agent class model, which describes abstract and concrete agent classes and captures the inheritance and aggregation relationships between them and an agent instance model, which identifies agent instances and their properties. An interaction model describes the responsibilities of an agent classes, the services it provides, associated interactions and control relationships between agent classes.

From the internal viewpoint three models, specific to BDI architecture, specify each agent class: a belief model, a goal model and a plan model. A belief model describes the information about the environment and the internal state that an agent may be in and the actions it may perform. A goal model describes the goals that an agent may possibly adopt and the events to which it can respond. A plan model describes the plans that an agent may possibly employ to achieve its goals or respond to events it perceives. An agent inherits its belief model and plan model from its super classes. Execution properties of the architecture determine how events and goals influence intentions and how intentions lead to the action and revision of beliefs and goals. A logical agent is the encapsulation of state and behaviour, developed during the system design process. A physical agent is the actualisation of one or more logical agents.

The Cassiopeia method (Drogoul and Zucker, 1998) is an agent-oriented role-based method for designing the MAS. It relies on the concepts of role, agent, dependency and group. An agent is a set of roles. From the local (agent) viewpoint roles are divided into three levels: individual roles, relational roles (i.e. how they choose to interact with one another by enabling or disabling individual roles) and organisational roles that describe how agents can manage their interactions to become or stay organized (by enabling or disabling some relational roles). In addition, there are two layers of roles from the global (system's or organisation's) viewpoint: the dependencies layer and the groups' layer. The order in which those five layers are designed is not determined, but it usually is from the bottom to the top, starting from individual roles and finishing with organisational roles.

Cassiopeia has been used to design a number of teams of soccer-playing simulated robots. There are some difficulties that make this domain very challenging for solving in multi-agent approach:

1. The game is dynamic, meaning that the behaviour cannot be planned in advance.
2. The original game (human soccer playing) is decentralised decision-making and the only co-ordination is common goals and the general strategic plan for the game. This decentralised approach should be followed for maintaining the essence of the modelling task.
3. It is difficult to express at the agent level the behaviours that allow the collective achievement of the task.
4. The operations of the opposing team are by definition unpredictable and consequently require a high level of real-time adaptability.

The GAIA methodology (Wooldridge, Jennings and Kinny, 2000) models an agent or multi-agent organisation as a collection of roles. There are abstract and concrete concepts in GAIA: concrete concepts are agent types, services and acquaintances; system, roles, interactions, permissions, responsibilities, protocols, activities, liveness and safety properties are abstract concepts. In a concrete realisation of a system (e.g. an organisation) abstract roles will be dynamically instantiated to actual individuals. A role is defined by responsibilities (liveness and safety properties), permissions (rights), activities and protocols (way of interacting with other agents).

The design in GAIA moves form abstract to more concrete concepts. Roles model and interactions model will be completed during the analysis of requirements statement. The objective of the analysis stage is to develop an understanding of the system and its structure. In the design phase, Gaia is concerned with how a society of agents co-operates to realise the system-level goals and what is required for each individual agent in order to do this. The Gaia design process involves generating three models – agent model (implemented through a set of agent roles), services model (what main services are required to realise the agent's role) and acquaintance model (documents lines of communication between the agent types). Gaia borrows some technology and notation form the object-oriented analysis and design in order to provide the software engineer with an agent-specific set of concepts for understanding and modelling a complex system. The organisational structure of a system, modelled in GAIA, should be static, inter-agent relationships do not change at run-time.

In **Multi-agent Software Engineering (MaSE)** approach Wood and DeLoach (2000) focus on the construction of a multi-agent system through an entire software development lifecycle from problem description to implementation. MaSE is independent of particular agent architecture and programming language. It is assumed that the modelled system is closed and a special agent that models the environment encapsulates all external interfaces and participates in the system communication protocols. The analysis stage in MaSE comprises determining of goals on the basis of requirements, building of a goal hierarchy, applying of use cases and sequence diagrams and refining of roles and concurrent tasks. The design phase consists of creating agent classes, constructing conversations, assembling agent classes and the system design through deployment diagrams. The methodology is iterative across all phases with the intent that iterations will add details to the models. A sequence diagram is used to determine the minimum set of messages that must be passed between roles. If there are several possible scenarios, multiple sequence diagrams are generated. Goals are associated with each role. The role's behaviour and the accomplishing of tasks are described by the set of sequences of activities associated with each role. So far MaSE is used for the design of small MAS (up to ten agent classes). The methodology does not consider dynamic systems where agents can be created, destroyed or moved during execution. Inter-agent conversations are assumed to be one-to-one, as opposed to multicast.

Researches on using the **UML** in modelling **multi-agent systems** have been fruitful in recent years (Bauer and Müller, 2003; Bergenti and Poggi, 2000). Originally the UML does not support the important aspects of multi-agent systems, such as autonomy and pro-activity of each agent and proactively changing dynamic communication pattern of agents. Nevertheless, the UML and its modifications are gaining more and more popularity in agent modelling (e.g. goal-cases, reaction-cases and FIPA interface introduced by Flake, Geiger and Küster, 2001).

Using the UML for modelling MAS is guided concurrently by the UML and FIPA and is well covered on AUML web-page (FIPA, 2005b). This group follows a pragmatic philosophy – when it makes sense to reuse portions of the UML, do it and when it does not make sense to use the UML, use or create something else. The most widely accepted alternatives for representing agents and their interactions in the UML are:

- AUML (Agent UML) (Bauer, Mueller and Odell, 2000);
- AOR (Agent-Object-Relationship) based modelling language (Wagner, 2002);
- PASSI (Process for Agent Societies Specification and Implementation) that integrates portions of UML and AI based methods (Burrafato and Cossentino, 2002; Cossentino and Potts, 2002);
- Tropos (Mylopoulos et al., 2001) that exploits class diagrams, activity and sequence diagrams from the UML or sequence diagrams from the AUML.

The standard UML is being extended to a product called Agent UML (the AUML, see for instance Bauer, Mueller and Odell, 2000). The AUML is part of the initiative of FIPA Technical Committee on Modelling for the developing of vendor-neutral common semantics, a meta-model and abstract syntax for agent-based methodologies (FIPA TC, 2005). The TC on Modelling applies different sources for building up the AUML – in addition to the UML 2.0 also the best ideas from other approaches to studying the agent-object relationship, e.g. Tropos, Gaia, AOR and PASSI.

The AUML uses a three-layer representation for agent interaction protocols: templates and packages to represent the protocol as a whole; sequence and collaboration diagrams to encapsulate inter-agent dynamics; and activity diagrams together with state-charts to encapsulate both intra-agent and inter-agent dynamics. The AUML satisfies necessary requirements for the use of agents in industry: it is related to the nearest antecedent technology (object-oriented software development) and uses artefacts to support the development environment throughout the complete system's lifecycle (Odell, Parunak and Bauer, 2000). FIPA has also initiated its own study of time constraints to be used in time- sensitive agents and multi-agent systems, which unfortunately is not as advanced as the time model used by OMG and in the Q-model (FIPA, 2003). When the AUML will be "released", it is possible to evaluate the strengths and weaknesses on the FIPA AUML in comparison to the OMG UML 2.0 - not only from the modelling perspective but also from the viewpoint of its possible incorporation into a whole design methodology. Currently the AUML specification has not been finally approved yet and is under development.

Table 3.1 The comparison of multi-agent approaches

No	The property	AAII	Cassiopeia	Gaia	MaSE
		1	2	3	4
1	The aim	Modelling abstract behaviour of humans	Modelling of the roles of agents (actors) – individual, relational and organisational	To support an analyst to go systematically from requirements to design	Design of MAS thorough software development cycle
2	The scope	MAS	MAS	Organisation	MAS
3	Necessity of a computer aided tool	No	No	(Yes)	No
4	Existing tools	?	?	Yes	General specification tools can be used
5	Strengths for the modelling of organisations	A BDI architecture supports well the modelling of humans, complex system can be easily decomposed, good systematic approach	Actual practical implementation	Large-scale, systematic	Integration with software development lifecycle
6	Weaknesses for the modelling of organisations	Too much actor-oriented, no relation to process	Not so systematic approach, at least not so easily understandable	Significant computer resources considered for individual agents?	One-to-one agent conversations, closed system (? "-"), "static" system, <= 10 agents, no validation or verification
7	Consideration of time issues	No	Dynamics is considered	No?	Neutral (-)
8	Consideration of employees (motivations, attitudes, etc)	Quite human-suitable agent approach	Individual human and teamwork is modelled	Yes, but they are not the main target	Neutral
9	Agent type	BDI	Roles	Organisation is seen as a collection of roles	Goals + requirements hierarchy
10	Use of the UML	Class models	No	Yes (roles model, interactions model)	Yes (use cases and agent classes)

The given review touched only upon some of the multiple approaches for the MAS. Some additional approaches will be addressed hereunder as to their suitability for specifying emergent behaviour in an organisation. In Table 3.1 MAS approaches reviewed in this section are compared. The comparison is similar to the comparison of conventional modelling methods in Chapter 2.

When comparing the same approaches against the requirements given in Section 2.1, the following can be pointed out:

1. All reviewed MAS approaches encapsulate the most important work processes and organisational aspects and are able to deal with a different level of processes. Goal representation, task decomposition and allocation as well as interaction and co-operation mechanisms can be naturally represented in the MAS. The existing information systems cannot be represented easily, neither can process modelling for development activities (except when building suitable MAS). Modelling results can be interpreted to everyday activities since simulation is a visually powerful method for expressing behaviour. As known, the MAS model can naturally be persistently updated (although the overall organisation is too complicated to present and only a suitable part can be modelled).
2. From specific requirements, the MAS does not support a detailed analysis of temporal criteria, although soft real time can be modelled and visualised. Employees' behaviour, motivations, interests as well as formal and informal communication can be easily modelled, if such a description exists.

The comparison demonstrates that those approaches that are related to BDI-agents (e.g. AAII) are more actor-oriented and systematic. On the other hand, those approaches that start from the concept of role (e.g. Cassiopeia and Gaia) are more organisation-oriented but not so systematic nor well designed. Therefore for the modelling of emergent behaviour in an organisation a solution could be to compose an approach from the positive characteristics of both approaches.

It is seen that the existing multi-agent solutions do not solve the problem fully. Unfortunately none of them considers yet temporal constraints described in UML profile for Schedulability, Performance and Time. Therefore in the future the ideal approach should integrate positive aspects of UML, AUML and FIPA standards. In general, an agent should correspond to the recommended generic structure of an agent (Motus et al., 2002). At the same time, many of those approaches can still be used since minor disadvantages do not create significant problems. There may be strong justifications to choose one existing methodology instead of creating a new one. In this thesis the intention is to show what could be the best possible approach for solving current tasks. For that reason it is chosen to have a new approach that would maximize the progress in solving essential problems. In real applications actual choice of the multi-agent approach should be made by the designer, taking into consideration all important related issues. Therefore, as a result, own approach is considered here that emphasizes communication and interaction between agents as well as agents' own decision mechanism (reactive or cognitive, depending on the complexity of the modelling task).

Software agent development environments. The introduced methodology uses currently the JADE (Java Agent Development Environment; Bellifemine, Poggi and Rimassa, 1999) agent development software tool for specifying agents' behaviour. JADE implements the distributed agent platform in the Java language. JADE agent model, implemented in Java, allows good runtime efficiency, software reuse, agent mobility and the realization of different agent architectures. Agent development in JADE can be integrated with other known development methodologies, for example with the Gaia methodology (Moraitis et al., 2002). In particular, its communication architecture tries to offer flexible and efficient messaging, transparently choosing the best transport available and leveraging state-of-the-art distributed object technology embedded within Java runtime environment (Bellifemine et al., 2001). Message communication between agents in Jade is FIPA (see FIPA, 2005a) compliant.

Jade is not the only possible or the best environment for the modelling of agents, e.g. Java is used also for other agent projects outside JADE, e.g. Williams, 2003.

Another possible solution for supporting multi-agent systems engineering (MaSE) might be AgenTool (DeLoach and Wood, 2001) or Zeus (see Nwana et. al., 1999). MaSE guides the designer from the initial system specification to the implementation by guiding the designer through a set of inter-related graphically based system models. The underlying formal syntax and semantics ties them clearly and unambiguously together as envisioned by MaSE (DeLoach and Wood, 2001). AgenTool offers the possibility to compose the agent classes from UML-specifications. This is a remarkable advantage since it allows to ensure that models (in the UML) used for the specification of agents and their behaviour can be used directly in a semi-automated way for building an agent. Unfortunately AgenTool does not support FIPA-compliant message exchange between agents and offers its own solution instead.

The ZEUS Agent Building Toolkit contains tools that are oriented towards the development of collaborative agent systems where the overall system possesses qualities at the level that any single agent cannot reach. ZEUS is also implemented as a collection of Java classes. A ZEUS agent is based on forward-chaining production systems, which are similar to expert systems that have been developed in the artificial intelligence community.

The JAVA approach was preferred in the current research as a very comfortable agent development platform where interactions are FIPA-compliant.

3.3.3 Multi-agent organisations

Multi-agent systems can form different organisational forms (e.g. families of agents, holonic systems, confederations, federations), depending on the autonomy and co-operation levels of its members. As a result, there is a remarkable freedom in applying organisational constraints to a set of agents. The existing approaches view multi-agent organisations quite differently, depending on the modelling task. For example, a multi-agent organisation is seen as a set of constraints imposed upon the activities performed by agents (Fox et al., 1998). Such an approach is similar to that of Weber (1987) who views the process of bureaucratisation as a shift from a management based on self-interest and personalities to a management based on prefixed rules and procedures. According to Ferber (1999), an organisational structure is an abstract description for a class of potential organisations and can be used as a base for the instantiation of a multitude of specific organisations. Similar ideas have been discussed in MIT SAG (2001).

A MAS organisation has similarities with other artificial intelligence approaches, e.g. distributed heterogeneous multi-agent systems can be considered similar to heterogeneous distributed transaction processing systems (Bose and Burd, 1997) or to the distributed workflow management system (Wang and Zhong, 2000). Lee and Zhao (2002) propose a modular architecture of concurrently communicating and synchronised modules for real-time agents that are situated in a changeful, unpredictable and time-constrained environment. Chaib-draa (2002) introduces causal maps for the modelling of agents' subjective views, for qualitative decision-making, organisational dynamics and interactions. A knowledge-based approach for handling exceptions in workflow systems is offered by Klein and Dellarocas (2000). The proposed solution is based on the process handbook project (Malone et al., 1999).

In MAS organisations issues of different roles, task decomposition and delegation of decision-making power among several roles are widely analysed (e.g. Britanik and Marefat, 1999; Fox et al., 1998; Thompson and Coovert, 2001; Taveter and Wagner, 2001; Taveter, 1999). A role is seen as a temporary purpose for an actor to participate in an interaction. A representation (fulfilment) of a role is a set of actions of an actor in a particular activity (that can be represented, for example, in Role Activity Diagrams; Abeysinghe and Phalp, 1997). Roles are acted out in parallel and they communicate through interactions. Many agents can act a single instance of a role and a single agent may act many role instances. Roles can be collected into the hierarchy and terms generalisation of a role and specialisation of a role can be used (Fox et al., 1998; Malone et al., 1999).

Task allocation (or distribution) involves organisational mechanisms through which agents can combine their skills to perform collective work (Ferber, 1999). Barber et al. (2004) point out that agent-based system architectures offer a modular distribution of decision-making responsibilities.

Co-operative distributed problem solving in multi-agent systems is an imitation of real world (Smith and Davis 1980; in Wooldridge, 2002, p 192): task sharing takes place when a problem is decomposed to smaller sub-problems and allocated to different agents; the key problem is how tasks are to be allocated to individual agents, considering their capabilities and speciality. External and internal models of competency can be distinguished (Bonjour, Dulmet, Lhote, 2002): an external view considers competency as a black box which provides a qualified action to the current input, internally competence is a dynamic arrangement of different cognitive resources that leads to qualified activity. Team decision-making in MAS is, for example, analysed by Kang, Waisel and Wallace (1998) and Carley and Prietula (1998) (Team Soar environment). Modelling of individual decision-making is analysed by Iwazume, Kato and Kanai (2001).

In studying the multi-agent systems we are interested in three levels of interactions (Van Aeken and Demazeau, 1998; Ferber, 1999):

1. The micro-social level, where the focus is on the studying the behaviour of an agent with other agents. (These studies are mainly done in distributed artificial intelligence approach and conventional studies of organisations). As a rule, the role and goals of agents in a small group are well defined and the topology of interactions is not very dynamic (in many cases strictly predefined).
2. The social level where the focus is on studying the functioning of agent groups in a multi-agent system. The differentiation of roles, goals and capabilities of the groups, the emergence of structures as interactions between agents and the influence of coherence between goals and roles of interacting agents on the efficiency of operation are examples of the problems studied at this level.
3. The macro-social level of global societies (or populations) where research interest is mainly concentrated on studying interactions of a particular MAS with its environment (e.g. with other multi-agent systems in the environment). Research on learning and adaptive behaviour of organisations and on self-organising structures and a weak similarity with the artificial life studies are characteristic at this level.

3.4 HUMAN ORGANISATION AS A MULTI-AGENT SYSTEM

This section concentrates on the possibilities of how to describe proactive components of a human organisation as a multi-agent system. In more detail it concentrates on the modelling of personal interests, interactions and adaptation.

In most cases multi-agent systems (MAS) are developed for the research and simulation of the behaviour of single actors or informal groups (or even large open systems like human society), but also a number of implementations are done in well-determined hierarchical human organisations in the military domain (e.g. Kang, Waisel and Wallace, 1998 or Lin, 1998). Agent technologies and multi-agent approach are also widely used in different application domains in order to compose artificial assistants in the human world (for example, office, banking, commerce and social services) with many different approaches (as introduced in Wooldridge, 2002 and Prietula et al., 1998). Most of the approaches concentrate on emphasizing agent intelligence and communication, whereas environment (and necessary reactions to it) and time are of minor importance.

The MAS can be used for representing an organisation in the way that an agent represents a physical or organisational entity or structural unit of an organisation. Consequently, an actor's behaviour and interactions between actors are considered as the behaviour and interactions of agents. To model the behaviour of a unit in an organisation, a unit may, in its turn, be considered as a multi-agent system. A structural unit may also be represented as a single actor and in the MAS also as an agent.

Oftentimes each structural unit has its own aim, task and goal function (similarly to these for the whole organisation and for each employee) that have an influence on the general performance of the organisation. The influence of collective opinions and knowledge is important to consider during the general mission decomposition into units' missions if units have to co-operate as teams. This is easy to model since agent-based system architectures offer a modular distribution of decision-making responsibilities (Barber et al., 2004).

A common understanding is that the choice as to which parts (structural units) or aspects (views to behaviour or performance) of an organisation should be modelled as agents, depends purely on the goal of the model.

The levels of interest in a human organisation, similarly to MAS, can be considered as the following:

1. The micro-social level where the focus is on studying the behaviour of small structural units of a larger entity.
2. The social level where the focus is on studying the inner functioning of an organisation as a set of structural units.
3. The macro-social level of global societies (or populations) where research interest is mainly concentrated on studying interactions of a particular organisation with its environment (e.g. with other organisations in the environment).

One should often build a separate model of the environment for studying organisations with strict dependability requirements. The environment in relation to the organisation can also be considered hierarchical and the organisation is typically situated at the middle level of the environment's hierarchy. This means that the "upper levels" of the environment are capable of modifying goals and assign additional resources for the organisation. The interactions between an organisation and the environment are twofold -- "down" to fulfil the goals and "up" to find sufficient resources and to adjust the goals. In many practical cases, but not always, the environment is proactive – the better the goals are fulfilled the more resources are assigned.

Multi-agent systems implement many foundations of human organisations research. Those similarities can be used in applying the MAS for the representation of different kinds of human organisations (e.g. reviews of Ferber, 1999 and Wooldridge, 2002). In more detail this applies to the modelling of processes, task decomposition, task allocation and decision-making.

The organisational structure characterises the class of the organisation on an abstract level (Ferber, 1999). The specific organisation is one possible instantiation of this structure. The same organisational structure can act as a basis for defining the multitude of specific organisations. This is similar to the generalization and specialization of a role presented by MIT SAG (2001). In this way, the hierarchical agent model may start with describing single actors at the bottom level and the whole organisation at the topmost level.

To fulfil its tasks, organisations often perform a fixed, hierarchical, predefined structure where the members of the organisation are highly specialised. A dynamic environment makes an organisation to modify its structure from time to time and re-assign tasks to employees. There are essentially two ways of approaching dynamic structures in the MAS (Ferber, 1999): either to give the organisation predefined mechanisms so that it can modify its behaviour and structure when certain organisational parameters reach a critical level or to incorporate genetic algorithms into the organisation (or its agents) and the organisation then adopts the structure which appears the best in the current environmental conditions or adopts a structure that provides sufficient flexibility to the agents, allowing them to dynamically adapt to the new circumstances within the same organisational structure.

3.4.1 Motivations and personal interests

Each employee in an organisation is a person with free will and his/her own interests, beliefs, desires, habits, etc that influence the actually performed work processes of the actor. The current research does not consider general personal motivations that very much depend on individual characteristics, are related to the general attitude of a person towards work and obligations, are very difficult to model in conjunction with the organisational goal and are highly difficult to predict. It is presumed that members of the modelled organisation desire to work efficiently and correctly in order to fulfil their tasks in the best way. The problem is that even with serious work attitude there are multiple action choices that lead to different results. In this research motivations are considered as a choice to perform an action from a set of possible solutions (actions) in a specific task according to loosely specified priorities or to choose a task from multiple tasks which deadline is similar. The choice preferences and instant motivations influence the decision-making and execution speed of chosen actions and thus determine the emergent behaviour (as a result of emerging situations) of the organisation.

Skills, motivations and personal interests of the actors influence the actually performed work processes in daily tasks, e.g. information gathering, performing reviews or analyses, giving information, preference in choosing activities and setting up priorities. Encapsulating motivations in the model improves the understanding of organisational behaviour and may suggest changes in work processes (e.g. task allocation) or work regulations (e.g. system of bonuses for efficient work) in an organisation. The representation of motivations in multi-agent systems is quite similar to these of real organisations: motivations are the reason that pushes an agent into action. In multi-agent systems motivations can be grouped into four categories (Ferber, 1999): personal motivations (an employee tries to satisfy his/her needs or to discharge the commitments), social or deontic motivations (related to the rules of the organisation in connection with the obligations and limitations that it imposes on its members), environmental motivations (e.g. reflexes – immediate actions to an external stimulus) and relational motivations (tendencies of other agents can be seen as similar or related to deontic rules).

Motivations can be confronting (e.g. because goals are contradictory), therefore to choose the behaviour that satisfies the needs and corresponds to the most important motivations can be quite difficult for an actor and it is a challenging task to consider it in designing a MAS. Human behaviour is initially often a reflection on the environmental event and only later the behaviour is adjusted according to the thoughts. In the MAS this can be implemented as filtered connections from perceptions to actions and executors.

Beside motivations, a commitment is a key component of collective action in the cognitive MAS. In making a commitment, an agent binds itself. All human social systems are based on the relatively complex commitment structures (Ferber, 1999). Forms of commitments can be distinguished similarly to motivations (commitments to oneself, environmental commitments, commitments to the social group and commitments of organisations to their members and relational commitments). Commitments impose constraints also on the resources used to accomplish agents' tasks.

The involvement of employees plays a great role in organisation efficiency. The resultant way for increasing contribution from each employee is assigning sub-goals of the organisation's goal to persons as their goals (Fox et al., 1988) but there is still the problem of generating co-operation among individuals who are confronted with conflicting choices. Each individual has to choose in which amount to contribute to the production of common good and to which amount to satisfy its own interests according to personal motivations, attitudes, etc. The reason for this problem may also be that no individual can directly observe the effort of another employee (Huberman and Glance, 1998). A solution can be to set up closely working teams – sets of experts with different knowledge and skills who interact interdependently and adaptively toward a common goal (Kang, Waisel and Wallace, 1998).

Human behaviour in agent systems can often be expressed by several human characteristics like beliefs, desires, etc (e.g. BDI-agents, Rao and Georgeff, 1995 or socially intelligent agents, Edmonds, 1997) but those characteristics are more difficult to express in conjunction with the pragmatic modelling of business processes. Therefore motivation is considered here as the only characteristic of the internal states of an agent (an actor). The external characteristic of behaviour is at the level of co-operation between agents.

The motivations of agents can be expressed in class models of an agent (e.g. in some of the generalised classes, from which an agent class is inherited). The specific preferences can more easily be expressed in specific choices in agent behaviours, expressed as agent methods. Although the general structure for an agent, as suggested, should be BDI-agent, the specific mapping of motivations that are desired to express into actions and that the agent does express in reality is left to the designer and it has to be often performed manually according to the views of the designer. Nevertheless, model testing and performance of agents in the model in comparison to actual actors indicates whether the model is correct or not and whether beliefs, desires and intentions are adequately designed in the agent model.

3.4.2 Interactions

Interaction is a dynamic relationship consisting of a sequence (a set) of related actions. Compatibility of goals, sufficiency of resources and abilities of agents in relation to tasks determine the nature of an interaction (Ferber, 1999): simple collaboration, obstruction and co-ordinated collaboration express ways of **co-operation** and pure individual or collective competition and individual or collective conflicts over resources express negative co-operation (Ferber, 1999).

Resultant co-operation (as an interaction between agents) is mutual co-operation where all the co-operating parties (agents or their groups) act in the way that is useful for all group members. In the MAS co-operation can be expressed as the behaviour of agents that satisfies the objectives of the group to which the agent belongs (Ferber, 1999).

In a formal organisation, the co-operation means co-ordination of actions (i.e. actually performed work processes) and collaboration by sharing tasks and resources (e.g. Schmidt, 1994, Schmidt and Simone, 2000 and Wang and Wu, 2000). Co-ordination of actions is necessary for managing a group of agents in order to make them act together for the fulfilment of a task. Collaboration by sharing tasks and resources is also a common method of co-operation in human organisations where several actors work together on a common task. In the MAS that models human organisations, common methods for co-operation are communication, specialisation, collaboration by sharing tasks and resources, co-ordination of actions and conflict resolution by arbitration and negotiation.

The key problem in implementing actual co-operation in the MAS is co-ordination. Efficient co-ordination even improves the production of common good in a measurable way (Huberman and Glance, 1998). Different schemas can be used for co-ordination like co-ordination through partial global planning, co-ordination through joint intentions, co-ordination by mutual modelling (e.g. an agent places itself in a position of another agent) and co-ordination by norms and social laws. In a multi-agent approach the norms and social laws can be either offline (previously) designed or emerged form system behaviour.

Some useful practical examples and researches on co-operation can be pointed out. Carley and Prietula (1998) introduce an additional criterion, named 'trustworthiness' of an agent; each agent can be trustworthy or untrustworthy. Decker (1998) introduces a tool for building and testing computational theories of co-ordination by using task analysis, environment modelling and simulation (TAEMS), used for example in a post office, hospital scheduling, airport resource management, Internet information gathering. TAEMS provides duration, quality and type of task interrelationships (e.g. enables, facilitates, precedes, causes, shares-results, cancels, uses, requires-delay). Kang, Waisel and Wallace (1998) introduce an agent-based distributed artificial intelligence computational model of team decision making (Team Soar). Each member of the team is an independent goal-oriented problem solver with its own knowledge about the world.

In human organisations two kinds of interactions could be detected: communication (i.e. sending and receiving verbal or written messages) and actions. Under actions every activity of persons is considered here that may be related to the organisation and its task. In modelling an organisation as a MAS, the real nature of actions themselves will not be considered, it is instead considered that during the communication it will be agreed or not agreed upon co-operation and actions correspond to the will of an agent to co-operate or not.

Communication is a way for interaction between friendly, neutral, or non-friendly agents or groups. Communication is often an essential method for co-operation. Usually only official communication is specified and analysed during the modelling of the organisation, although informal communication has often major internal influence on task achievement and organisational performance and efficiency (e.g. negative informal attitudes cause choices in co-operation and may cause problems in fulfilling expected timelines). In the MAS communication expands the capacities of agents by allowing them to benefit from the information and know-how of other agents. Often communication is an essential method for co-operation. In each organisation, formal and informal communication can be distinguished. Formal communication is related to the goal and official work processes of the organisation. Informal communication, related to the members' (agents') internal opinions, intentions, interests, etc, has often major internal influence on organisational performance and efficiency, but usually it is not modelled. Communication can be observed and modelled at different levels in an organisation (Ferber, 1999):

1. At the micro-social level we are essentially interested in the interactions between actors as personalities (agents). At this level most of the

studies in distributed artificial intelligence have been undertaken. The interaction at this level may be aimed at the better achieving of an organisational goal (formal communication), but also other items can be discussed and internal sympathies and antipathies are shown (informal communication).

2. At the level of groups we study the differentiations of the roles and activities of the agents, the emergence of organisational structures between agents and the general problem of the aggregation of agents during the constitution of organisations. At the level of groups, interaction and co-operation between structural units to form the general performance of an organisation is considered as formal communication and interaction between informal groups with different interests is considered as informal communication.
3. The level of global societies (or populations) where interest is mainly concentrated on the dynamics of a large number of agents, together with the general structure of the system and its evolution. Research in relation to artificial life is quite frequently located at this level. At the level of global societies communication of the organisation with the environment, the public and stakeholders as well as pressure from different interest groups is to be considered. Informal communication, such as opinions and pressure from interest groups outside of the organisation, may sometimes have a fatal influence on the organisation ability to fulfil its tasks – e.g. when resources given to the organisation depend on external decision-makers.

In a real multi-functional organisation all those three communication levels are essential.

Organisational modelling aims at abstracting from the interactions between the agents of complex multi-agent systems and their fundamental and recurrent patterns (Ferber et al., 2000). Therefore the agent-group-role model can be used. This leads to some certain types of behavioural requirements within an organisation model: single role behaviour requirements, intra-group interaction and communication successfulness requirements (including social exchanges) and inter-group interaction requirements (Hannoun et al., 2000).

Communication in the MAS is normally expressed as sending or receiving a message from another agent. The behaviour of the agent is adjusted according to the received messages and internal decisions on those messages. In this way an interaction means a set of communicating acts between agents (Ferber, 1999), which have an influence on the future behaviour of the agents. A conversation can be described as a series of states linked by transitions. Co-operation between agents is implemented in the MAS by sending messages and performing co-operative actions. Some different notations can be used to express a conversation (e.g. finite-state automation or interaction diagrams). Since agents are sometimes involved into several conversations simultaneously and they have to manage those multiple conversations, it is then easier to describe those interactions by using the Petri nets. The Petri nets also allow not only to describe the sequence of an act but also to represent internal states of agents during the different phases of conversation or co-operative action.

Several agent communication languages (ACLs) exist (Wooldridge, 2002; Ferber, 1999), like KIF, KQML, or the FIPA agent communication languages. The FIPA communication language and the KQML are similar, e.g. the structure of messages is the same and the message attribute fields are also very similar. Comparison can be found in Wooldridge, 2002. The most important difference between the two languages is the collection of performatives they provide. The role of performatives is to determine the type of a message and to enable to react and understand the message in the same way by the sender and the receiver(s) (of course, if a system is designed so). Of course, any ACL is a narrowed set of a natural language, but FIPA performatives allow the grouping of a wide range of natural languages with multiple nuances into an understandable set of categories as performatives and let the nuances stay in the body of a message. This fixed set of performatives simplifies the modelling of organisational behaviour and makes the analyses more efficient. The most used performatives in the FIPA ACL are inform, propose, accept-proposal, reject-proposal, query-if and often used not-understood.

In most MAS that are elaborated today, agents act in behalf of human users. The design of proper and user-friendly interfaces has been an important topic of MAS-design community for years. The problem of trustworthiness may emerge also at communication level – an agent may use falsified messages or stolen (false) identity. The problem can be solved in different ways, for example by using a public key infrastructure (PKI) for distributed agent systems (Nikander, 1999).

3.4.3 Learning and adaptation

Learning and adaptation. Adaptability of the organisation means the ability of an organisation to change its behaviour, processes and structure in order to fulfil better the (possibly changed) goals or to adapt to a changing environment. Organisational learning, adaptation and seeking for partners can be in more detail presented as adaptation and learning of members or the organisation. This can be viewed as individual or collective knowledge.

Adaptation of employees is often in the following directions:

1. Better implementation of their work activities, knowledge and opinions.
2. Additional training for obtaining new work methods or for emphasizing the most correct ones.
3. Understanding one's own roles and the roles of colleagues, evaluation of one's own abilities and adjusting of one's own work methods according to this analysis.

Adaptation of the organisation consists of the following aspects:

1. Better determination of the role of the organisation, comparison and evaluation with colleagues and modification of its own behaviour.
2. Analysis of environmental (i.e. other organisations, work domain) situation and planning of additional training to employees.
3. Improving co-operation with other partners (organisations).

Learning and adaptation is difficult to model and is normally omitted. It can be modelled in agent methods either as more competent choice of corresponding behaviour or a slight dynamic modification of an existing behaviour. Learning or adaptation can be also implemented in agent methods by a designer as actual modification of the agent code. The methodology, presented in Chapter 4, allows modelling learning and adaptation, as outlined in Secion 4.4.

3.5 SUMMARY

The evolution of computer science and software engineering encourages the merge of earlier separately studied organisations' modelling issues with the general-purpose studies in developing computer systems. This is accompanied with the shift in goals for organisation modelling and in the aims of practical usage of those models.

The models for organisations are conventionally used to serve the off-line analysis role: determine the appropriate goals of an organisation, develop modification plans and define the measurable criteria for assessing the actually achieved goals. The feasibility of an organisation's structure, coherence of goals defined for structural units of an organisation with the overall goal of the organisation and satisfaction of time constraints imposed upon the daily operation of an organisation are not covered with the conventional modelling methods, as discussed in Chapter 2 of the present thesis.

This chapter suggested that the extension of the theoretical basis applied to modelling organisations – from algorithm-centred models of computation to interaction-centred models of computation – enables to include the above-listed, hitherto uncovered issues into the models of organisations. This has been demonstrated by the developments in the UML (e.g. UML 2.0 and its profiles, especially the profile on Schedulability, Performance and Time, OMG, 2003a) combined and enhanced with the recent results from distributed artificial intelligence, such as multi-agent systems. Such an evolution also suggests that models could be used as on-line support to decision making in strategic and tactical issues of running and/or modifying organisations.

In this thesis, the conventional methods of an organisation's modelling are extended with methods from the UML approach and timing analysis approach to real-time systems. The subjective and proactive features of organisations are modelled by applying agent-based methods. As indicated, models of interactive computing and agent technology in particular seem to be the most suitable for modelling emergent behaviour in an organisation. Unfortunately none of the introduced methods (the UML, Q-model, MAS) alone can model all the characteristics of emergent behaviour in human emergent organisations. Therefore the suggested methodology combines all three approaches – UML diagrams are used for describing work processes and actors, the Q-model is used for the analysis of temporal criteria of processes and interactions, and proactive components of the organisation and their behaviour are modelled as a multi-agent system. The UML, agent approach as well as the Q-model are essentially based on the interaction-centred paradigm. Whereas the conventional modelling dealt basically only with the designed behaviour, the interaction-centred and proactive modelling also covers emergent behaviour. The methods that form the ground of the proposed methodology are briefly compared in Table 3.2.

According to Tabel, the UML (especially use case and activity diagrams) is suitable for the modelling of work processes. Since the current version of the UML and available CASE tools do not fully support the simulation of timing criteria of processes, the Q-model will be used as an additional option for modelling the issues of time and performance. Interactions between agents, personal choice and freedom as well as additional human characteristics are more feasible to simulate by using an agent-oriented approach, for example represent the organisation or a part of it as a multi-agent system.

A multi-functional human organisation where different actors co-operate towards a common goal but still perform their tasks independently from each other and synchronise their activities by using interactions (message passing) is very similar to a multi-agent system. There are several strong aspects in favour of using multi-agent approach for systems architecture or modelling (Nguyen, 1997):

1. The application of the multi-agent technology has a positive impact in systems engineering because this technology supports the principles of modular design and implementation. Agents can be considered as independent modules and hence be designed separately. A system is then constructed by integrating those methods into a team.
2. Agents can be easily reused.
3. Multi-agent technology offers reliability through the application of the redundancy of agents.
4. Easy representation of the cumulative effects of emergent behaviour since the behaviour of an organisation is naturally seen from the viewpoint of multiple actors (if an agent represents an actor).

Validating and verifying the behaviour in MAS is problematic partly due to non-linearity of interactions and their non-deterministic nature, therefore this has to be set up and solved carefully (Dugdale and Pavard, 2000; Hogg and Jennings, 2001). It might be easier when agent-based systems are applied to finding proofs of behaviour (e.g. O-ANTS; The Omega group, 2001).

Table 3.2 A comparison of selected modelling and description methods

No	The property	UML	The Q-model	Multi-agent approach
		1	2	3
1	The aim of the implementation of the method	Description of processes, use cases, actors, activities and timing issues in a unified language	Modelling of timing parameters, common resources, exchange of data and control signals	Modelling of interactions, possibly also motivations and co-operation
2	Usual modelling scope	The whole organisation, all processes	All processes	In principle all actors and the environment
3	Necessity to have a computer aided tool for modelling	Recommended	Recommended	Yes (strongly recommended)
4	Existing tools	Multiple	Limits PC, maybe also other tools	Multiple
5	Strengths	Unified, multiple aspects	Timing	Flexibility (allows to model very different aspects, communication modelling)
6	Weaknesses	No timing aspects yet, no agent standard yet, multiple variation as “mights”	Complexity, not commonly used	No standard approach, one issue can be represented in different ways
7	Consideration of time issues	Dependencies are included, new profile about time is issued as an addition	Yes	Yes, if implemented
8	Consideration of employees (motivations, attitudes, etc)	On future standards on agents	Can be implemented as selector processes	Yes

Quality assurance issues in agent-based systems are not well solved (e.g. Wille et al., 2002). It is often very difficult to verify that the agent system behaves at the required level of service quality. This can be partly understood since behaviour in agent systems is emerging by its nature. Still the agent system should be designed so that it may represent some guarantees for the required level of service quality, if end users (owners) of the system demand this. Such a requirement is compulsory for allowing the agent system to be used in more serious implementation domains.

The development of complex systems requires an incremental and modular design process. Specifications are stepwise refined to implementations. Refinement comprises, among others, behavioural, state, interface, architectural, communication refinement. Correctness can be established by verification (that the more refined system corresponds to the properties of the more abstract one, etc the other way round) or synthesis (proves algorithmically). Mathematically it is not yet possible to prove complex systems. Modular design can be used as a design based on a notion of a component (i.e. composition of interface and behaviour). It is believed that correct design of units or components combined with correct placement and correct communication leads to a correctly developed global system (Dosch, 2003).

Theoretically and methodologically, there are not many known attempts of building time-aware proactive models for a human-centred organisation. The new approach – as described in the following chapters, has been applied and tested on separate fragments of a large non-profit organisation (the Estonian Police) and the results should be considered as a pilot project that is used for feasibility study of this approach.

4 SUGGESTED NOVEL METHODOLOGY

This chapter describes a novel methodology for modelling time-sensitive emergent behaviour in human organisations. This is a suggested solution to the research problem specified in Section 1.2 of the thesis. The suggested approach relies on the UML, multi-agent approach and the Q-model, introduced in Chapter 3.

The chapter begins with outlining the methodology and the overall change management life cycle. The chapter continues with the detailed description of each step in the proposed methodology. The given information is again summarised and the overall process illustrated at the end of the chapter. The case studies as examples on the implementation of the methodology are presented in Chapter 5.

4.1 REVIEW OF THE METHODOLOGY

The current approach for modelling emergent behaviour in organisations is based on three concepts:

1. *Organisation model* that describes the organisation at the moment of modelling.
2. *Change model* that encapsulates the expected behaviour of the organisation before and after process modifications, represents different organisational states during modification process and thus assists in change management.
3. Modelling methodology – a suggested process of devising *organisation model*, *change model* and the implementation of modification suggestions in the organisation.

Each of these concepts is briefly reviewed hereunder and in more detail in the following sections.

The *organisation model* (will be denoted here as *M*) is a *model of the organisation*,[3] used in the current methodology. The *organisation model* consists of interrelated components (as illustrated in Figure 4.1) and encapsulates the organisation from four viewpoints:

1. The organisational structure and the general description of organisation tasks, goals, existing activity sphere and its key actors are described in a natural language (this is in more detail described in Section 4.2.2 of the thesis).

[3] The *model of the organisation* is defined in section 1.1.2

2. Planned and actual work processes of an organisation and related actors are listed in a special form of the *process table* (introduced in Section 4.2.3) and formally specified by using UML use case, activity and sequence diagrams (as described in Section 4.2.4). Use case diagrams describe interactions between actors and activities. Activity diagrams are used for the modelling of activities and their sequence. Interaction diagrams can be used for a more detailed specification of interactions and communication between actors.
3. Temporal criteria of processes (e.g. activation frequency, duration, data consumption and production times) and interactions between processes (e.g. synchronisation and data exchange) are formally analysed and more precisely simulated in the Q-model, as described in Section 4.3 of the thesis.
4. Selected proactive components, behaviour related to subjective human factors (e.g. actor's goals, preferences, competition of the priorities of different tasks as well as competition between official and personal priorities, commitments, personal interests and decision mechanisms) and interactions between actors (agent activities and formal and informal communication) are specified, using agent class diagrams. The agent model is described in Section 4.4.

   A subset of processes and actors is chosen for multi-agent simulation. The composition of multi-agent simulation, its aims and results are in detail presented in Section 4.4.4 of the present thesis. The aim of multi-agent simulation is to concentrate on specific key issues of organisational behaviour (chosen by designer or specified in the modelling task) and visualise in detail the expected emergent behaviour in chosen situations.

Internal description of an agent (e.g. its methods) is presented in UML class and other diagrams and analysed in the Q-model. Multi-agent simulation is designed according to UML use case, activity, sequence and class diagrams. The UML diagrams of multi-agent simulation are a subset of the overall UML diagrams, composed for describing the organisation, its processes and behaviour, as illustrated in Figure 4.1.

The *change model* (will be denoted as *C*) is a partially ordered set of sub-models. Each sub-model is an *organisation model* (described above) that represents the organisation at a specific time of modelling. The *change model* consists of minimum two *organisation models* and a list of modification actions:

1. An *organisation model* that represents the current organisation at the initial moment of modelling.
2. An *organisation model* that represents the expected state of the organisation (i.e. its structure, actors, processes and behaviour) after planned modifications.

3. A list of activities on *organisation model* that transfers the model *M* from its initial state to the desired state.
4. The *change model* may consist of additional *organisation models* that represent the organisation at intermediate stages during the change process from the initial state to the desired state.

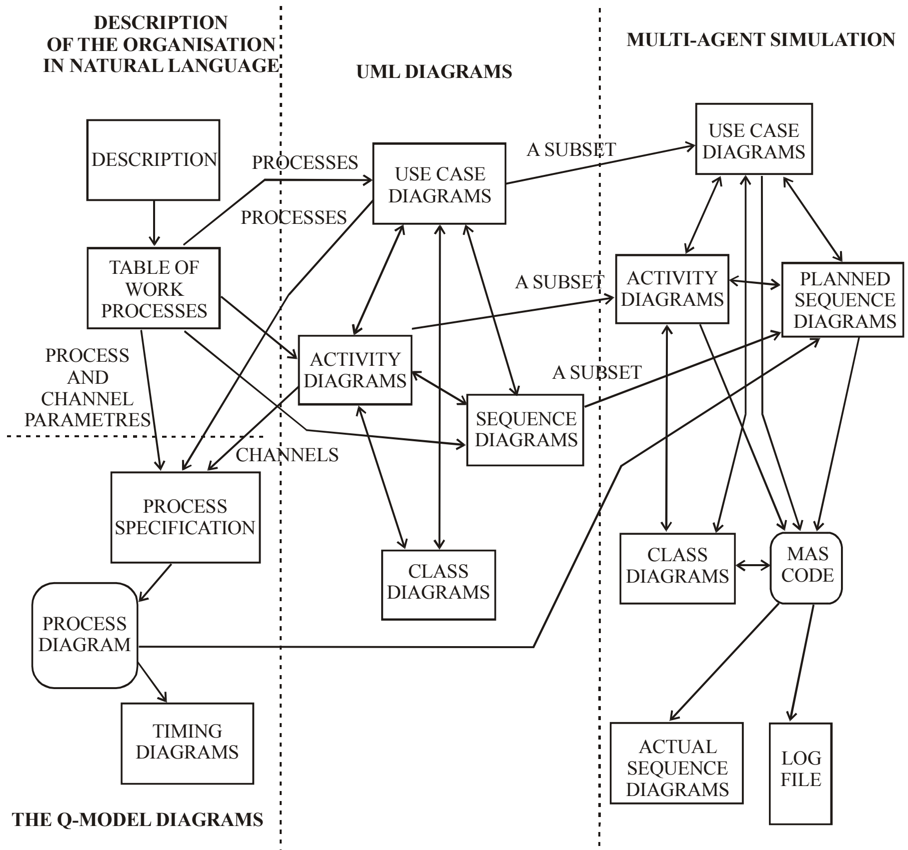

Figure 4.1 Integration of model components

Change model can be also considered as a *dynamic model of the organisation* since it represents dynamic behaviour of an organisation during its modification. This general idea is similar to EKD approach (Bubenko, Persson and Stirna, 2001) introduced in Section 2.3: an *enterprise model* consists of a business process model for the initial state, a business process model for the desired state and a change process model. Details in the models for the suggested methodology and in EKD are different because of different modelling aims, as described in Chapter 2 of the thesis. The *change model* is in more detail described in Section 4.6.

The overall modelling, model development and change management lifecycle in the current approach are an iterative process (cycle) where more specific results are reached stepwise (e.g. similarly to the Unified Software Development Process or shortly the Unified Process, Arlow and Neustadt, 2002). The current modelling methodology introduces four general stages as described hereunder and illustrated in Figure 4.2.

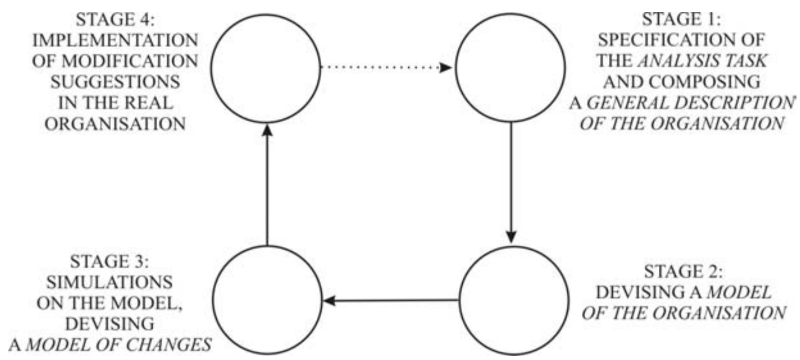

Figure 4.2 A general view of model development and change management lifecycle in the current methodology

The modelling begins with the formulation of the analysis problem and the modelling task (will be called as *analysis task*). The *analysis task* consists of prioritised sub-tasks and measurable criteria for the later evaluation of modelling efficiency.

The first stage of the methodology comprises also a general description of the organisation and specification of its work processes. In this stage the goal, tasks and inner structure of the organisation are reviewed and described in a natural language. The aim at this stage is to clearly state the purpose of the analysis and the expectations about modification results and to compose a description of the organisation by using natural language and UML use case and activity diagrams.

The second stage completes the (so called *static*) *organisation model* – a representation of the organisation at the moment of modelling in its initial state. The activities in this stage aim at the specification of the temporal criteria of processes and proactive components of the organisation – modelling from multiple actors' perspective. In this way this stage transforms the process table and UML description into the Q-model and multi-agent presentation, as described above. Currently LIMITS CASE tool (Limits, 1998) is used for analytical study and simulation of the Q-model. For the study of agents' behaviour JADE (see Bellifemine et al., 1999) tool is used. As a result of the second stage of the methodology, the model of an organisation integrates different views of the modelled organisation.

The third stage of the methodology is aimed at supporting change management. This stage comprises experiments and simulations on the *organisation model* and composition of a *change model*. The *organisation model* is used for modelling different scenarios in order to elaborate suggestions for the modification. The *change model* serves as a supportive material for modification suggestions and describes the expected emergent behaviour of the organisation during and after planned modifications.

Finally, the fourth stage of the methodology is the actual change management – the implementation of organisational modifications in the actual organisation, the monitoring and evaluation of implementation results. This stage is to a great extent not specific to the current methodology and conventional change management approaches for implementation can be used. The only difference is that the current methodology offers *change model* that can be used for assisting the planning of modifications and evaluating the outcome of implemented modifications.

The next sections of the chapter will describe the model components and the modelling process in more detail. The modelling process is summarised and the implemented components are compared at the end of the chapter (Section 4.8).

4.2 DESCRIPTION OF THE ORGANISATION AND ITS PROCESSES

4.2.1 Specification of the analysis task

Modelling of an organisation according to the proposed methodology begins with the determination on the goal of the current modelling activities. This can be written down as a short (about some sentences up to one paragraph) description of the aim of the analysis. Correct determination of the analysis goal is the precondition for successful analysis, modelling and possible modification of the organisation.

According to the modelling goal the *analysis task* is specified. *Analysis task* is a description of what to concentrate on during the analysis of the organisation. The *analysis task* consists of prioritised sub-tasks. Each sub-task consists of the following elements:

1. Priority of the sub-task (it is also the sequence number of the sub-task),
2. Name of the sub-problem,
3. Short description of the analysis sub-problem,
4. Existing situation,
5. Expected outcome,

6. Criteria for evaluation of how well the results are to be reached.

The *analysis task* is convenient to present in the form of a table as given in Table 4.1. The given example concentrates on the analysing of quality and speed in an information exchange between multiple processes and actors (illustrations in this chapter are in general taken from the first case study, presented in Chapter 5 of the thesis).

Table 4.1 An example of the analysis task (taken from Table 5.1)

Priority	Problem name	Short description of the problem	Existing situation	Desired result	Criteria for the evaluation of the result
1	Operativity of information exchange	Evaluate whether the existing information exchange routines are optimal enough	Not known, the existing situation is to be described during the analysis	Time criteria for information exchange are determined in all stages	Preciseness of the description, accuracy of the model and number of correctly specified temporal criteria
2	Duplicate processes	Evaluate whether there are duplicate processes during information exchange	The existence and justification of duplication is not known	The purpose and necessity of all related processes are known and evaluated	Preciseness of the model, correctness of necessity evaluation
3	Information exchange speed	Evaluate whether there are possibilities to speed up information exchange and if yes, give recommendations	Information exchange seems to be optimal but the analysis is not made	There exist time criteria for information exchange and estimation about additional resources	Preciseness of the composed review

The composition of the *analysis task* may require multiple iterations: initially it will be set up in general terms; elaboration of a more detailed analysis task is performed later in conjunction with the more specific description of the organisation and the exact determination of the modelling focus.

4.2.2 General description of the organisation

The *general description of the organisation* (will be noted as the component *D* of the model *M*) is presented in a written form in a natural language. This description should specify the most important characteristics of the organisation and give background information for the more detailed analysis of work processes and actors. If the modelling problem, described in the *analysis task*, is not related to the whole organisation, a more detailed specification may cover only the related units, actors and actions. This simplifies the model and reduces the amount of work needed.

The general description of the organisation consists of five components:

1. Goals and goal functions. This component can be considered as goals model in Enterprise Modelling (see Section 2.3 of the thesis).
2. Norms and other juridical frames. This component can also be seen as the business rule model in EM.
3. Methodical guides, rules and other intra-organisation documents. Depending on their content, documents listed in this group may belong either to the goals or business rules model in EM.
4. Structure of the organisation.
5. Relations with the environment – i.e. how the environment influences the organisation and what the activities of the organisation are.

The first three components correspond to *activity boundaries* that are specified in Section 1.1.1 and illustrated in Figure 1.2.

In the current methodology, guiding documents and goal functions are used for describing the expected behaviour of actors and structural units and for evaluating their actual performance compared with the desired performance (from the viewpoint of the organisation).

The results of the previous related surveys or quality (efficiency) measurement programs (e.g. Balanced Scorecard or Best Value, described in Chapter 2), if they exist, are welcome to be included in the description document. This enables a qualitative analysis of the organisational performance. Description in natural language will benefit from the support of the feasible description notations (e.g. diagrams used in the field, SWOT and other analyses, statistics etc).

4.2.3 The process table

The next step in devising the *organisation model* is a detailed specification of work processes and structural units. Process descriptions will be detailed until the level of structural units and specific employees (actors). This can be considered as a business process model in EM. Resources and personnel skills should also be described, if relevant. Information exchange is considered here as a very important component of work processes. Proper information exchange has crucial influence on the effectiveness of work processes; therefore information exchange should be described, as well.

Specification of work processes is normally given in a form of written document. It may be organised into several sub-sections, where each sub-section is designated for a specific actor or group of actors (employees) and consists of three parts:

1. A short description of the aims of the actor (roles) and its work processes.
2. The *process table* (will be noted also as *W*) that consists of descriptions of the most important activities, which will be used further in the modelling.
3. Further detailisation of chosen processes by using UML use case and activity diagrams.

When using some EM technique, actors and resources can be specified in the corresponding actors' model and resource model.

The current approach uses a special form of a table for the *process table* in order to simplify the analysis and its later description in other notations (i.e. in the UML and in the Q-model). Each row in Tabel *W* describes a work process W_i that has the next properties:

$$W_i = \{q; n; d; p; a; \mu; t; \varphi; \nu; \tau; r\}$$

where

q – process (sequence) number or ID – unique number for process identification during the whole analysis and modelling;
n – action name (string) – name of the activity (process) as used further in the documentation;
d – action description – description of the activity in the form of free text;
p – purpose of actions (free text) – short explanation of why this action is needed;
a – executor, actor who should complete this activity;
μ – input description (free text or a set of conditions) – description of where to find data for the action and when this data is used;
t – input time parameter -- invocation time, frequency or period (time) – a parameter that described when and under what conditions (reasons) the process is activated, how regularly this happens;
φ – duration of process (time);
ν – output description (free text or a set of conditions about the results of the process;
τ – estimation on the validity time (permanency) of output values;
r – additional remarks (free text).

An example of the form of Tabel *W* is given in Table 4.2. The order of columns is changed for better readability.

The processes that are described in Tabel *W* must be detailed until *elementary processes* are reached. An *elementary process* is the smallest action unit that will be used as a complex module for building activities, processes and behaviours. The level of *elementary processes* is chosen and specified by a designer in each modelling. In general, *elementary processes* are not described in more detail (however, use-case or Q-models can be used for the representation of internal components of those *elementary processes*). A work process for an employee can be represented then as a combination of those *elementary processes*.

Table 4.2 An example of the process table (*W*)

Process ID	Activity name	Purpose	Input		Action description	Duration	Actor	Output		Remarks
			Description	Time criteria				Description	Validity	
q	n	p	μ	t	d	φ	a	ν	τ	r
1	2	3	4	5	6	7	8	9	10	11
II	Receiving of calls and controlling of patrols									
J2.2.A	Receiving the call about the vehicle theft		Call		Answering the call, receiving information, making the decision	2..3’		Information to be used		
J2.4.A	Registering the information about vehicle theft		Call, info, decision		Registering in the information system, checking the data in registers	3’		Correctly entered information		

It is possible to have later analysis and optimisation phase of processes – i.e. internal optimisation of *elementary processes*, relation of the elementary process to other *elementary processes* and optimisation of work processes by optimising the existence and combinations of *elementary processes* in a process (e.g. changing execution parameters, data producing and consumption parameters and information exchange parameters).

4.2.4 Description of the work processes in the UML

It is important to devise a detailed specification with explicitly determined relations between processes. Therefore the Unified Modelling Language UML (Booch, Rumbaugh and Jacobson, 1999) is suggested for describing the processes of the organisation. Standard UML is used here because of its simplicity and popularity. Any UML-supporting description or analysis tool can be used for describing the system and its processes (e.g. Rational Rose, see Rational 2003; or Visio, see Visio, 2003).

Use case diagrams are used to describe the interactions of the actors and activities. The process table (W) can be interpreted as a description of the system use cases in the high format. The correspondence of columns in Tabel W is given in Table 4.3.

Table 4.3 Relation between the terms in use cases and in the *process table*

The meaning in UML use case notation	A column in the *process table (W)*
Use case	Activity (q, n)
Actors	Actor (a)
Purpose	Purpose (p)
Description	Action description (d)
Type	--

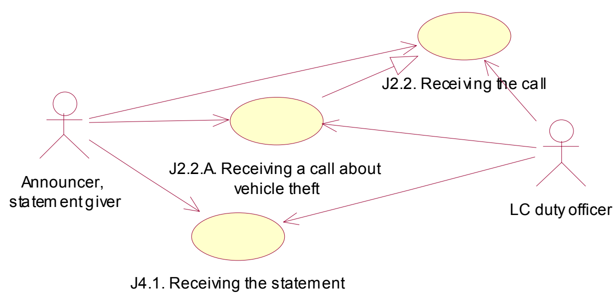

Figure 4.3 An example of a use case diagram

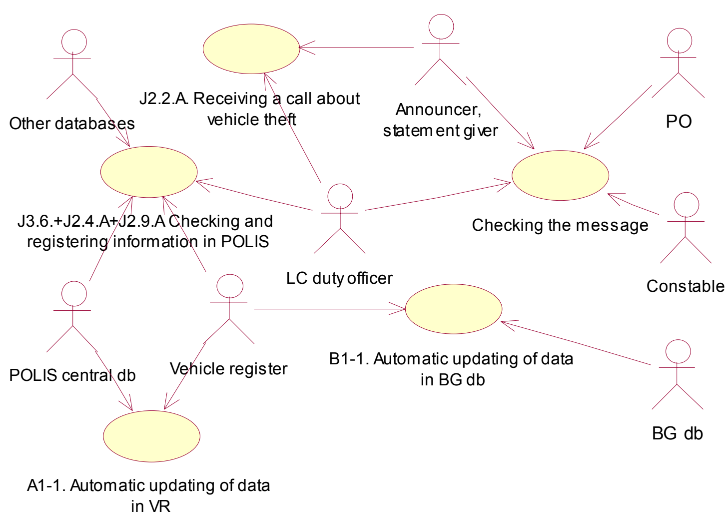

Figure 4.4 An example of combined use case diagram

The processes are represented in UML use case diagrams in accordance with Tabel *W*. Since no automated tools exist so far for converting Tabel *W* into UML diagrams, this work has to be performed manually by a designer. When describing use cases, the first single actions or actors are described. A simple example of a use case is given in Figure 4.3. Two actors (Announcer and LC duty officer) are presented in the figure. Activity J2.2 is a generalisation of the more specific activity J2.2.A. Simple use case diagrams can later be combined for more complex diagrams (as shown in Figure 4.4) in order to get a better understanding of the total system.

For modelling activities and their sequence, UML activity diagrams are implemented. The diagrams can be of different complexity. In the latter case complex activity diagrams contain activities of multiple actors or the whole interacting system, including some actors from the environment (as illustrated in Figure 4.5). Such complex systems that describe and analyse the overall system are the key aspect for the introduced approach. Often bottlenecks or performance problems do not emerge from single process sequences in a sub-system but as a result of interactions of several actors with their processes and their own temporal characteristics.

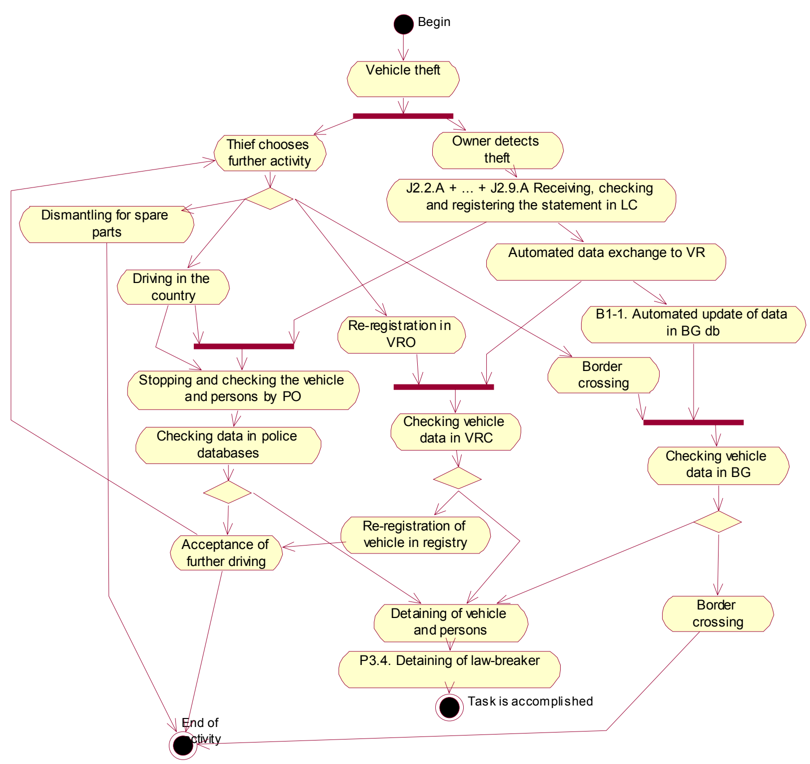

Figure 4.5 An activity diagram representing the whole system

Use case and activity diagrams are interrelated during the whole modelling, helping thus to analyse and compare differences. If it is necessary, a more detailed view on interactions can be given by using sequence diagrams. Use case, activity and sequence diagrams together form the UML-component *U* of the *organisation model*. The component *U* is normally stored as a model file in the format of modelling environment.

Section 4.2 can be summarised as the following. In this section the activities of the first stage of the methodology were described. The first stage of the methodology is a classical phase of problem definition and a description of the modelling domain. The aim of the stage was to set up a clear analysis and modelling task and to describe the organisation and its work processes. In more detail the modelling process at this stage of the methodology is divided into four steps:

1. Initial determination of the analysis goal and composition of *analysis task*.
2. Composition of the *general description of the organisation*.
3. Refinement and finalisation of the *analysis task*.
4. Specification of work processes (filling in the *process table* and composing use case and activity diagrams in the UML).

As a result, three components of the *organisation model* (the model will be noted here as *M*) were completed during this stage:

1. *General description of the organisation* in a natural language (component *D* of the *M*).
2. *Process table* (component *W* of the *M*).
3. Description of processes in UML use case, activity and sequence diagrams (component *U* of the *M*).

Each step can be implemented iteratively with more details until a satisfactory level of preciseness is reached.

4.3 SPECIFICATION OF TIMING CRITERIA BY USING THE Q-MODEL

This section describes the modelling of timing issues. The modelling of timing criteria is separated from the UML modelling since the standard UML does not support time-sensitivity and the real-time UML is not standardised nor supported by integrated tools. At the same time the UML description can be easily transformed into the Q-model and agent presentation, as shown hereinafter.

The specification of processes by using the Q-model (the Q-model itself was introduced in Section 3.2) implements the *general description of the organisation (D)*, the *process table (W)* and sequence and activity diagrams in the UML (component *U*). The process table in the Q-model notation (will be noted as component *Q* of the model *M*) describes the main modelling and simulation parameters. Tabel consists of the following fields:

- *q* as process id (used in use-cases and in the *W*);
- *n* as process name (will be used in the Q-model diagram);
- *T* as execution timeset (may be described over explicit time set or starting period length, deviation and simulation mode, or by incoming ports and channels);
- Θ as equivalence interval;
- for each input port of the process:
  - η_a as earliest data consumption;
  - η_b as latest data consumption;
- in the case of an input selector process:
  - μ_d as port combination;
  - μ_p as probability of this combination;
- α as duration of the process: shortest execution time;
- β as duration of the process: longest execution time;
- for each output port of the process:
  - α as shortest execution time;
  - β as longest execution time;
- in the case of an output selector process:
  - υ_d as port combination;
  - υ_p as probability of this combination;
- *r* - remarks.

An example of Tabel is presented in Table 4.4. At the beginning channels between processes are not described separately, but through related processes as process input and output characteristics (i.e. in the same single row). Since most of the given processes do not have explicit timeset and they are executed by another process, the column *T* is not given in Tabel explicitly and its values are presented in remarks.

The Q-model table *Q* and the *process table W* are related to each other. The rows in the proposed table *Q* should correspond to the processes described in the *process table* (*W*). Process descriptions and values for certain process parameters (i.e. the columns) in Tabel *Q* are derived from Tabel *W*, process interconnections (sequence, synchronisation, data exchange) are taken from activity and sequence diagrams of the component *U,* as illustrated in Figure 4.6. Table 4.5 lists fields in Tabel *Q* and compares them with fields used in the process table *(W)*. There are no suitable tools available for converting the process table *W*, use cases and activity diagrams given in the UML notation into the Q-model notation - therefore this work is to be performed manually.

Table 4.4 An example of work processes in the Q-model notation

Process ID	Process name in the Q-diagram	Input		Duration of the process, minutes		For output selector process		Equiva-lence interval, minutes	Remarks
		Ear-liest data con-sumption	La-test data con-sumption	Shortest exe-cution time	Longest exe-cution time	Port combi-nation	Combi nation proba-bility		
q	n	η_a	η_b	α	β	υ_d	υ_p	Θ	r
1	2	5	6	7	8	9	10	11	12
	Vehicle_ theft	-	-	0,5	3			0,01	period length 100, deviation 10
II	Receiving of calls and controlling of patrols								
J2.2.A + J2.6	J22A_J26_ Information_ receiving	0	0	27	31			5	
J2.9.A	J29A_Event _registering	0	0	2	3			2	
	Record_start	0	0	0,01	0,01			0,01	
	Record_end	0 / 0,1	0 / 0,1	0,02	0,02			0,01	
	Pol_db_ result	0	0	0,2	0,6	port5	1	0,5	
VI	Activities of VRC								
	VRC_db_ result	0	0	0,1	0,3	port3	1	0,01	

Some guidelines can be given for the describing of processes by using the Q-model. The *process table (W)* was usually developed from top to bottom from more general processes until *elementary processes* (see Section 4.2.3). The description of the processes in the Q-model notation is recommended to start from the *elementary processes* since conditions of timing and interactions (that are crucial in this step) can be more clearly specified for simple processes. The next step is then to model larger components of work processes, such as combinations of the *elementary process* modules, in line with the *process table*. One can call the models of these medium-level components as *component process modules*.

The Q-model allows a hierarchical representation of processes. When creating a *component process*, inner structure of *elementary processes* (and also *component processes*) that participate in the process module will be handled as black box and they will be presented as a single node in the model.

As all real actions in the organisation can be successful or unsuccessful, it is important to model the so-called “wrong” but realistic outcomes, as well. In the Q-model different behaviour scenarios can be implemented by using selector processes (see Section 3.2 of the thesis). It can be discussed and decided dynamically during the modelling to which extent “wrong” outcomes should be modelled in order to keep the Q-model diagrams understandable. For better clarity it is reasonable first to describe only "correct" and planned solutions and when the correct behaviour is tested, to add also other possible behaviours.

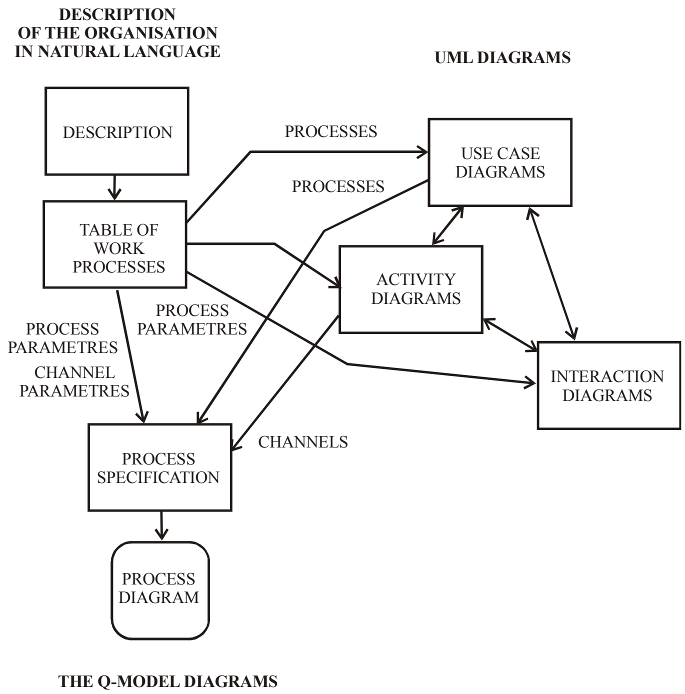

Figure 4.6 Use of model components *D*, *W* and *U* for composing process specifications in the Q-model

The whole behaviour is finally described as a combination of process modules and/or the modules of elementary work processes. There may be different suggestions about how big those process modules should be and how many levels to use until the topmost level is reached. This problem is similar to the problem regarding the creation of different classes while modelling by using the UML and therefore suggestions given by Booch et al. (1999) are appropriate. In general, the number of different models must be small enough to allow the handling of the entire picture, but big enough to allow easy combining and to avoid repetitions of descriptions if the processes are similar in reality.

Tabel *Q* serves for the preparations of modelling the work processes in a software environment. For resultant modelling, the information given in Tabel needs additional specification. For example, for convenience purposes the channels were not exhibited in Tabel but as for implementation, Tabel may need elaboration with regard to the information on the specification of the channels, depending on the specific project.

Table 4.5 Comparison of fields in the *process table* and in the Q-model

The fields (components, parameters) of *W* that are used in Q-model tables and diagrams			Notation and usage (implementation) in the Q-model (table and diagrams of the component *Q* of *M*)			
Column in Tabel *W*	Notation	Description	Column in Tabel *Q*	Notation	Description	Where used
1	2	3	4	5	6	7
1	q	Sequence No (process ID as used in UML diagrams)	1	q	Process ID	Process name
2	n	Action name	2	n	Process name	Process name
3	p	Purpose, justification				
4	μ	Input description (free text)	3	μ	Starting conditions	Used for the determination of incoming channel types
			In an input selector process:			In input selector processes
			3A	μ_d	Port combination	Port combination
			3B	μ_p	Probability	Probability
5	t	Input timing parameter (activation frequency and when data will be used)	4	T	Starting period	Start period (period length, deviation, simulation mode)
			11	Θ	Equivalence interval	Equivalence interval
			5	η_a	Earliest data consumption	Earliest data consumption (input port)
			6	η_b	Latest data consumption	Latest data consumption (input port)
6	d	Description of the activity (free text)				
7	φ	Duration	7	α	Shortest execution time	Shortest execution time (output port)
			8	β	Longest execution time	Longest execution time (output port)
8	a	The usual actor				
9	υ	Output description (free text)	In an output selector process:			In output selector processes
			9	υ_d	Port combination	Port combination
			10	υ_p	Probability	Probability
10	τ	Output validity time (also in relation to next activations)		τ		Channel function (earliest data generation, latest data generation)
11	r	Remarks	12	r	Remarks	

The Q-model analysis and simulation are implemented in Limits PC CASE Tool (Limits, 1998). All diagrams and models, stored as Limits CASE Tool projects, are seen as the component *L* of the model *M*.

Process specification in the Q-model environment comprises the composition of the Q-model diagrams. Some misbehaviour may be understood already form process specifications. The main analysis concentrates on the analysis of timing diagrams, as described in Section 3.2.

The Q-model has a significant disadvantage. Models of real organisations with many details become very complicated and are extremely difficult to handle in the Q-model notation. For that reason it is recommended to choose only a minimal necessary set of processes.

4.4 MODELLING OF ACTORS

The process of devising an *organisation model* continues with describing actors as agents. The whole organisation is then seen as a multi-agent system (MAS). The purpose of this step is to capture emergent behaviour, therefore actors' choices and interactions between different actors are analysed and simulated. *Actors* are entities of an actual organisation. An *agent* is considered as a modelled entity of an *actor*. An agent represents either an actual person or an active artificial component (e.g. database engine).

The agent model consists of the following components:

1. Agent class diagrams in the UML (a part of the component *U* of the model *M*).
2. Multi-agent simulation use case, activity and sequence diagrams (component *A* of the model *M*).

Agent class diagrams are described in this section; multi-agent simulation and its diagrams will be described in Section 4.4.4 of the thesis.

Agent class diagrams are developed from use case diagrams. The development of agent class diagrams starts from the determination of agent classes. The *analysis task* (described in Section 4.2.1) determines what structural units or aspects (i.e. views on the behaviour or performance) of the organisation are to be modelled or focused on more deeply.

In general, all real key actors (e.g. persons with their planned ideal tasks, artificial agents like robots, software agents, database engines etc) that are related to the modelled processes in different roles and described in use case diagrams should be described as agents. Additionally, the described artificial actors may improve the completeness and trustworthiness of the model (i.e. for representing all the most important interactions).

A separate class should represent each actor that has different characteristics from each other (for example, classes SalesClerk and Customer). An agent is thus an instantiation of this agent class and corresponds to the respective actor in the use case model (e.g. SalesClerk for sales clerks and Customer for customers).

The agent class model can be layered (Ferber, 1999): actors belonging to a structural unit can be described in MAS that describes a corresponding structural unit; in this way structural units in the organisation may be described as multi-agent systems. At the same time (e.g. if collective knowledge, goal, motivation or behaviour of a structural unit as a whole are important to take into consideration), a structural unit can also be seen as an agent in (another) MAS that represents the overall organisation or system (the organisation with other outside actors and the environment).

When analysing the system and modelling actors, it should be documented what processes are done by which actors in which roles, what kind of actors are executing the different roles, what their skills, training needs etc. are, how other actors are related (in which roles respectively). There should also be the analysis on the difference of work processes in general and in each unit if there are similar (e.g. territorial) units, as to what the decision-making background is, what kind of direct or indirect issues affect the decision-making. For example: direct influence factors (actor's evaluation of the situation, conditions, attitude and obtained information) and indirect influence factors (initial and additional training, methodical guidelines and communication with colleagues). In some situations or organisations the initial training may have a bigger or smaller importance but such aspects should clearly be considered or omitted. Information exchange should be considered both from the organisational viewpoint and the person's viewpoint.

Agents that model one employee or artificial actor are considered as *elementary agents*. Agents that model groups of agents or units will be named *composite agents* respectively. The agent class may include internal classes that describe the agent architecture and behaviour (this is necessary for modelling purposes when implementing an agent as a piece of software). Internal architecture and abilities of an agent are described in general lines in sections 3.3 and 3.4.

The class model should be able to represent both aspects – organisational and human subjective aspect. The agent professional and personal properties should be derived from the system description and seen as a whole for the modelled system. This means that both professional and personal (human) characteristics should be described by a set of interrelated characteristics and by a hierarchy of classes, presented as follows. In general, two kinds of uppermost parent classes should be described: one group for describing job and professional characteristics and work methods and the second for modelling human characteristics (for example, classes OrgTasks and Person).

4.4.1 Specification of tasks

Work methods and routines are structured into a hierarchical system of classes where topmost class is, for example, OrgTasks. Similar work routines for multiple actors are reasonable to collect into one class, which in turn can be a parent class for more specific work descriptions (for example, class SalesClerksTasks is a parent class for classes FoodSCTasks and HouseholdSCTasks) to distinguish different, more specific tasks and roles.

The overall hierarchy in this case is simple, as described in Figure 4.7: class SalesClerkTasks inherits some of its operations and attributes form the class OrgSalesMethods and this, in turn, is a child class for general classes OrgTasks and GeneralSalesMethods. The class SalesClerk represents an actual actor. For more complicated cases, for example if activities of sales clerks depend on different departments where they work, the class SalesMethods may have child classes FoodSalesMethods, HouseholdGoodsSalesMethods, etc. Further detailisation of those classes eventually leads to agent classes (FoodstuffSalesClerk, HousholdGoodsSalesClerk). Similarly, the class Customer inherits some of its attributes and methods from the class CustomerSkills and personal characteristics form the class Person.

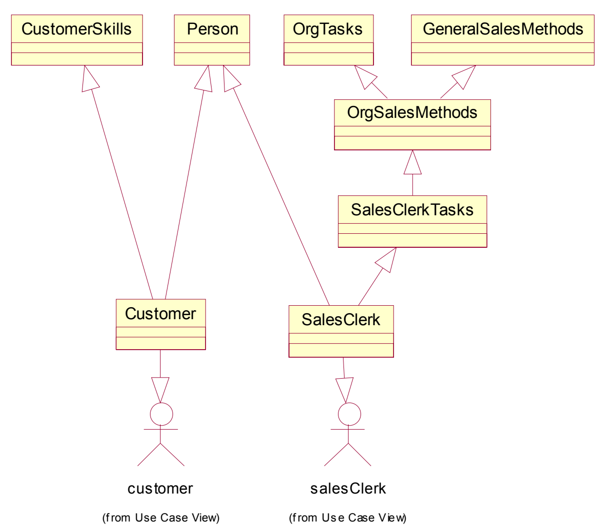

Figure 4.7 A simple example of a class diagram for agents

This kind of hierarchical system of method classes helps to better understand what methods are common for different actors and what routines are different. It also helps to ensure that in case of modifying some work routines, methods for all corresponding actors would be updated. The actual mapping of different characteristics into different classes depends on the wish of the designer of each specific system.

A designer may wish to model both actual and ideal work processes. In this case this may be described in the same agent model or in different agent models. The latter is especially recommended if there is a more detailed analysis of processes and the possibility for broad re-structuring of the organisation. In this case it is recommended to determine first what the ideal roles should be and which employees are needed and then to take the best employees one by one, put them into model and then look which changes have to be made compared with the initial ideal proposal.

4.4.2 Modelling of motivations and personal characteristics

The methodology stresses that the actual actors are humans with their own beliefs, desires, intentions, priorities, preferences and free will. Those characteristics cannot be derived from work and process descriptions.

The human side of the actors is considered by inheriting some of the agent properties from an upper class that represents human characteristics (the class Person). Several approaches can be taken regarding what to consider as the attributes and methods for this class (for example as described in different agent approaches in Section 3.3.2). In this research a set of characteristics is chosen, some of which are important for this modelling approach (e.g. attributes like preferences, priority tasks and the list of actor's own priorities or methods like select activity) and others are just added as examples for enabling a better review of the complete picture (e.g. attributes like beliefs, desires and intentions or methods like update priorities and interests). This set, given hereinafter, does not pretend to be universal but indicates the most important characteristics for this approach:

1. Attributes:
   - 1.1. Name – name of the actor
   - 1.2. Preferences – personal preferences
   - 1.3. Ideas – actor's own ideas
   - 1.4. Priority Tasks – actor's own important tasks, goals and aims
   - 1.5. List of Actor's Own Priorities – a list that gives priority to important tasks
   - 1.6. Beliefs – world view, used for the interpretation of received information
   - 1.7. Desires – wishes, more general than explicit priority tasks
   - 1.8. Intentions – desired or chosen activities according to the received information and actor's own desires, influences on choosing the activity
2. Methods (operations):
   - 2.1. Select Activity – choose the corresponding activity

2.2. Calculate Interest – evaluate what the priorities are and how much interest there is to choose work activities or duties (is used in agent simulations)
2.3. Update Priorities and Interests – adjust attributes according to the changed situation, past activities and received information

The inheritance of human characteristics can be implemented hierarchically where the topmost class is the class Person or Humans and several classes at middle levels describe different ethical and other properties for different groups of actors in the system (e.g. classes Policemen, Criminal, SalesPerson, LoyalCustomer and Employee). Often the chain of the inheritance of personal characteristics is quite short compared with the inheritance chain of work characteristics and methods. The motives of employees (or more precisely, the designer's estimations about the issues that characterise motivations and other personal characteristics) should be modelled as much as possible, as described hereunder.

Therefore, according to the currently proposed methodology, for each agent that represents a human actor in the system its agent class methods usually constitute an individual combination of properties of two uppermost classes: one uppermost class describes work processes and methods and another class describes human characteristics. Since the UML does not allow dual inheritance, for the sake of correct implementation classes in one inheritance chain (for example those that describe work methods) are seen as interfaces. For example, it can be suggested (but is not obligatory) that the personal characteristics should be implemented as actual classes and classes that describe job methods and tasks are described as interfaces.

The above described approach enables to describe actors from two perspectives – as employees or executors of job characteristics and as persons with their own interests, motives and habits. The modelling of each system is always unique and therefore it is up to the designer to decide how many of those proposals can be omitted and how the model of the system can be simplified.

For a better illustration we can consider another example about the police where also one specific class (called the actor class) is designed for each actor: the class POOfficer describes the patrolling officer and the class Constable describes constables. Each of those classes inherits its descriptions of work tasks from the specific duty class, respectively PODuties and ConstableDuties. The job descriptions for those classes are a sub-part of the description of all methods that the police performs on the field; therefore those classes are child classes of the parent class PoliceFieldDuties. In this way it is possible to construct an even longer hierarchy that will eventually lead to the uppermost class PoliceDutiesAndMethods. This approach is illustrated in Figure 4.8 with an example from the law enforcement domain.

Capturing of the methods and actual properties of the actors cannot be done only from use case diagrams but certainly activity diagrams (that represent the dynamics of the system) should be used, as well. If necessary, a more specific description of the system behaviour and communication should also be described by using sequence diagrams.

If it is necessary to distinguish in more detail different actors of the same actor type (e.g. multiple customers, each with its own attitude) or to explicitly specify what we consider the personal characteristics of a single specific actor, we can create an agent class (like ThisPerson, AngryCustomer) that inherits the attributes and methods of the general class Person (or Customer in the described case) and modifies those actual behaviours.

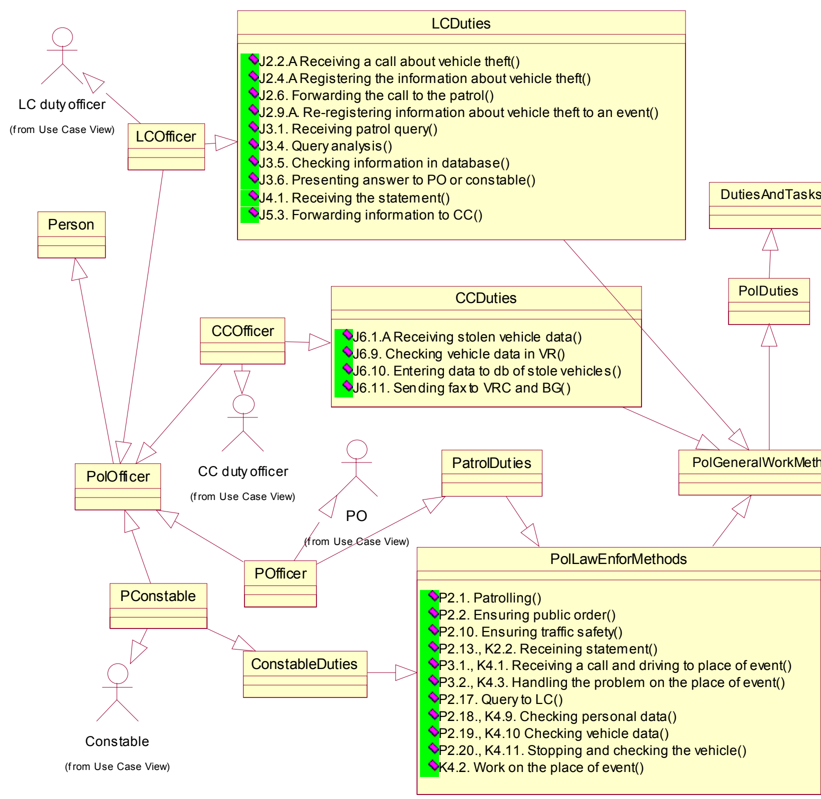

Figure 4.8 An example of a class diagram for agents

The **agent basic architecture** may be different, depending on agent properties. For the pure modelling of human characteristics, the agent could have one of several different architectures. For example, one possibility is BDI (belief-desire-intention) architecture: such agent has necessary properties – beliefs, desires and intention. In reality the proper agent architecture depends on the tasks of the agent. All necessary human characteristics can be modelled also by other (even simpler) architectures, for example planning or reactive architecture with reflex or teleonomic behaviour (according to the classification in Ferber, 1999 or in Section 3.3.1 of the thesis).

The actual behaviour of an actor depends on **motives**. Personal characteristics like sympathies and antipathies between employees or problematic issues like the goals of the organisation versus ethical and moral (and other) visions or goals of the organisation versus personal interests of the employee, including career interest, should also be discussed and modelled whenever possible. At the same time the modelling of employees' motives is sophisticated. Sometimes the modelling of motivations and co-operation can be solved by introducing additional procedures or special behaviour (as agent methods).

It is possible to simulate **decision-making** in an actor that takes into account organisational goals and personal interests. Organisational goals can be represented as a semi-formal description. The Balanced Scorecard (Kaplan and Norton, 1996) is taken as the basis in the current research but other solutions are also suitable. In the BSC (as described in Chapter 2) strategic goals for the organisation consist of sub-goals. For each goal there is a measure that has a target value. Goals and target values can be used in the agent description so that internal (state) variables store the priority of the goal, the weight of each measure, the current state (value) of each measure and the desired state of the measure. In addition, additional internal variables can be used, describing the desired speed for the improvement of each measure. One of the agent methods is designed for weighing and choosing correct behaviour, determining the importance (according to the consideration of the agent) of the measure and, respective to this, taking a corresponding action. BSC model is possible to use as a representation of the "official" part of agent beliefs and desires since BSC system for an organisation can be represented in a semi-formal way and BSC charts can be designed at a very detailed level (e.g. the level of employee or role) for different tasks and activities.

The representation of **"personal" beliefs and desires** of an agent can be done in different ways. If "official" goals are represented in BSC, it is also possible to construct a BSC chart for "personal" characteristics. This enables to compare priorities and different aspects as well as to implement a uniformed approach for representing both motivations for behaviour in a uniformed way in the agent model. In Figure 4.9 the representation of free will and preferences of an actor are illustrated: actor employee1 (left) is supposed to act according to job preferences, actor employee2 takes first into consideration its own priorities (while still trying to fulfil job expectations).

Learning and adaptation can be implemented either as modifications in activities that execute the chosen behaviour or by introducing special more complex behaviours as agent methods and during agent design the designer implements the corresponding decision mechanism within the agent.

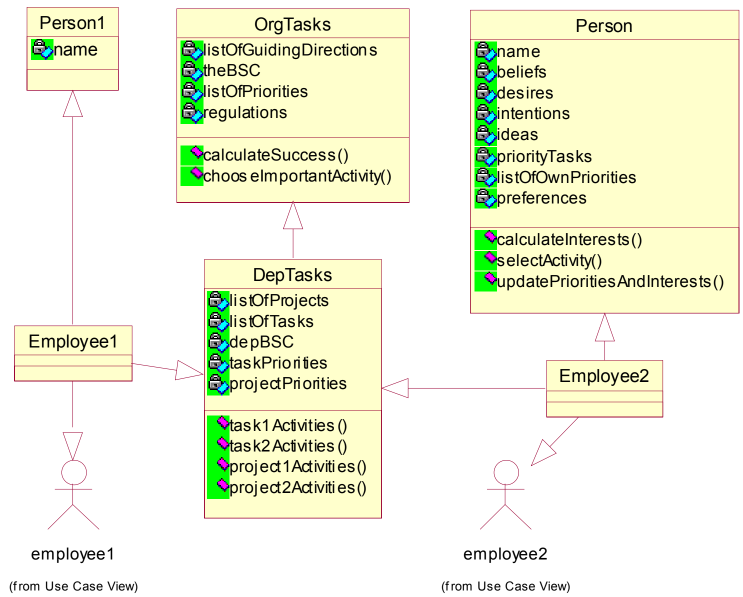

Figure 4.9 Representation of an actor class: without personal preferences (employee1) and with its own choosing of activities (employee2)

4.4.3 *Completing the agent class model*

The internal structure of an agent (or how the methods and components in an agent are organised) is not determined at this stage of the research. An agent is composed of goals and strategies so that a strategy is dedicated to fulfilling a goal. FIPA (see FIPA, 2005a) interface is determined in agent class descriptions in methods and in remarks.

For a more detailed modelling of dynamic behaviour in agent methods activity diagrams can be used. These are similar to diagrams used for the specifying of work processes but here the emphasis is on the agent methods and their proactive selection by an agent. Sequence diagrams are used to model inter-agent behaviour and interaction protocols.

For modelling purposes, internal description of an agent (e.g. methods) may be described in more detail in the UML and in the Q-model and stored in the components *U*, *Q* and *L*. In this way the set of the Q-models in the organisation and the set of descriptions using agents are very closely related in the model, as presented in Table 4.6.

Table 4.6 Relation between agents and the Q-model

Level	Category	Agent representation	The Q-model representation
1	The organisation	Multi-agent system	- (Not used)
2	The organisational goals	- (Not used)	- (Not used)
3	List of actions / processes, requirements	- (Not used)	Work process (the highest level in hierarchical representation)
4	The structural unit	(Composite) agent	- (Not used)
5	Plans, work process descriptions, instructions	Agent methods	Work process (component process)
6	An employee	(Elementary) agent (class diagrams)	- (Not used)
7	Employee's job description (tasks for the role)	Agent methods	Component process
8	Employee's daily routines	Agent methods	Elementary work process

The agent model is now almost complete. The other actors (like databases, system components and other organisations) should be also described by their respective class models and possibly described as agents. In this way the overall system can finally be described by a number of class models that together perform a complete set of classes that describe the structure and possible performance of the system.

The multi-agent approach models easily choices and interactions. The multi-agent approach also enables the modelling of personal characteristics like intention for co-operation, readiness for resultant activity, motivations and other characteristic more easily than with UML use case, activity diagrams or the Q-model diagram. Unfortunately even the models of MAS can express "inside characteristics" of agents at a very limited level only. Actually, only the designer's estimations about these "internal characteristics" of actors that probably influence personal choices and actual work processes will be modelled.

The presented approach allows an easy modelling of human characteristics and choice making (e.g. between multiple tasks) at the agent level. This is also applicable at the organisational level.

4.4.4 Multi-agent simulation

The modelling of a selected part of the organisation as MAS is seen as a complementary (additional) measure for simulation, analysis and dynamic visualisation of emergent behaviour. If possible, the simulation would cover only the most necessary part of the analysis scope at the same time, presenting still the important issues of the overall behaviour. If the *organisation model* is quite voluminous and complicated, there may be several simulations, each of which concentrates only on a certain part of the system or its behaviour.

When collective knowledge, goal, motivation or behaviour of a structural unit as a whole is interesting, the unit can be modelled as a group, coalition or family of interacting agents and studied more deeply. A unit may also be represented as a single agent. According to Ferber (1999) and Kinny and Georgeff (1997) multi-agent systems can be hierarchical so that each level may be an agent that consists of several agents belonging to a lower level.

Multi-agent simulation allows modelling of what happens accordingly to specific messages and what their results are. In the Q-model simulation it was very difficult to analyse the possible reactions to messages with different content. While in the Q-model it was important to specify the performance of work processes, in agent model the agents (actors) and the communication between them is emphasized. The overall system is built up as descriptions of agents and their methods. Cooperation, any kind of control and storing of the data in the log file are merely the result of inter-agent communication. In the agent model the agent can be created, executed (run) or stopped and messages can also be sent to them. Proceeding from this there is a very good possibility to model their communication and understanding ability or non-understanding ability of messages received and the behaviour exhibited according to this.

Simulation model is derived from the *organisation model (M)*. The initial existing use case, activity, sequence and class diagrams serve as a basis for constructing use case, activity, class and sequence diagrams of the computer-simulated MAS.

The simulation model is a simplified *organisation model*. The simulation model serves for devising a multi-agent simulation that is the most useful one for representing key aspects of emergent behaviour from multiple actors' viewpoint. The simulation model (component *A* of the model *M*) consists of the following components:

1. UML use case diagrams,
2. UML activity diagrams,
3. UML sequence diagrams,
4. UML class diagrams.

The design starts with composing the model of simulated MAS. While the components *D*, *W*, *U* and *L* of the initial model *M* are platform-independent and emphasize essential characteristics of processes and actors in the current modelling task, agent simulation models (components *A*) are platform-specific and therefore also include components related to the specific implemented multi-agent system (e.g. communication activities). Figure 4.10 illustrates the usage of the existing description, UML diagrams of the whole model and the Q-model process diagrams for devising UML diagrams of the simulation sub-system. For example, sequence diagrams that describe planned behaviour are of great importance in designing agent behaviour.

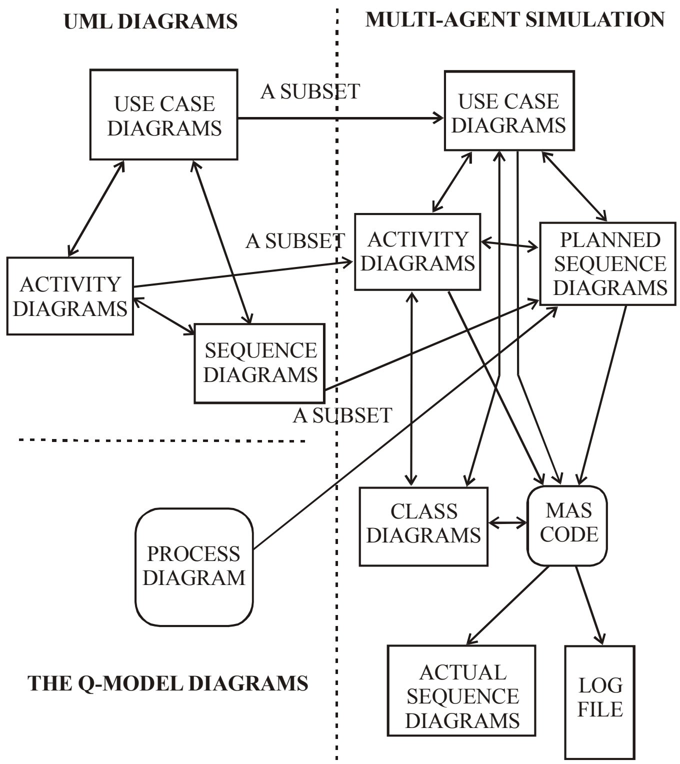

Figure 4.10 Use of model components *D*, *W* , *U* and *L* for composing diagrams for multi-agent simulation

Simulation model has the purpose to exhibit emergent behaviour. This is the reason why they differ from the UML diagrams of the organisation (the component *U*) but they must in principle represent all the important characteristics of the actual models. The simulation use case, activity and sequence diagrams often emphasize communication and interactions between agents. Internal activities inside agents (i.e. in agent methods) are described more generally (or just as a timelock where the only activity is to wait for a certain time). It is necessary to use sequence diagrams for the detailed presentation of communication situations inside planned processes.

Class diagrams depend on the implementation platform and they encapsulate only related characteristics and methods. Figure 4.11 illustrates a simulation class diagram that has only the most important methods and platform-specific approach compared with the class diagram given in Figure 4.8. For example, class diagrams for JADE agents need a different approach than platform independent class diagrams described in Section 4.4. In JADE an agent is built up from detailed and very specific behaviours, therefore classes in JADE must be much more detailed than in the platform independent UML agent model of the overall system. This is also one reason why simulation in MAS modelling can be considered as complementary to UML description. The simulated agents' class diagrams could be collected into the same repository with other UML diagrams of the current modelling.

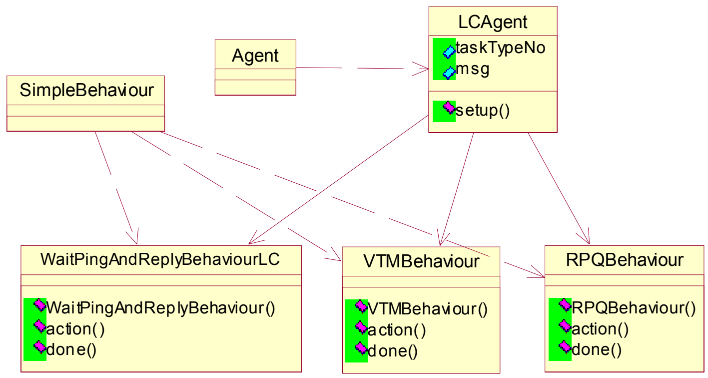

Figure 4.11 Class diagram of a simulation

The choice of what actors and processes are essential to simulate, depends heavily on the aim of modelling. One must still keep in mind that emergent behaviour emerges as a cumulative collection of multiple different aspects and actions. It might also be useful to consider the outer environment or other agents that should be taken into account - as one agent or as specific additional routines in modelled agents.

The agents that are used for modelling can be of different architecture. For a more exact simulation of human behaviour BDI (belief-desire-intention) architecture would be used (e.g. Kinny and Georgeff, 1997). Currently more simple agents are used during the implementation of this methodology that may have reflex cognitive, reflex reactive or teleonomic reactive attitude (as described in the first case study in Chapter 5). The choice is based on the suggestion to use as simple agents as possible that could describe the modelled behaviour accurately.

The agent simulation can be easily used for a deep analysis and demonstration on how agent (and so the overall system) behaviour depends on modifications in agent methods. For example, agents may initially be designed for fulfilling the organisational tasks of the highest priority. Later, during the simulation, additional characteristics and personal tasks may also be included in the models. This is a good place to simulate idealistic (i.e. strictly according to plans) and more realistic (with different disturbances) behaviour.

It is possible in agent methods to describe as internal algorithm how the choice and performance of certain tasks will be made. This can be also individualised by describing in the method class (or interface) the concrete choice criteria and in agent methods the additional preference criteria of a specific agent. In the current example the process of making personal choices is not implemented (as cognitive agent) and each agent performs in the optimal way according to its understandability (as reactive agent).

JADE (Bellifemine et al., 1999) agent modelling software tool is used for the study of agents' behaviour. The overall agent simulation is stored in a JADE project. The executable code must be implemented manually and checked by the modeller that it would correspond to the UML diagrams. Co-operation, any kind of control and storing the data in the log file are merely the result on inter-agent communication. The JADE code means the description of an agent in a class and behaviours in its sub-classes as given in the first case study, presented in Sections 5.1 – 5.4 of the thesis. Inter-agent communication can easily be analysed by using an interaction diagram, illustrated in Figure 4.12. The diagram is readable from up to down; arrows exhibit messages between agents (represented as rectangles). Text on the arrow indicates a performative (a performative exhibits message type).

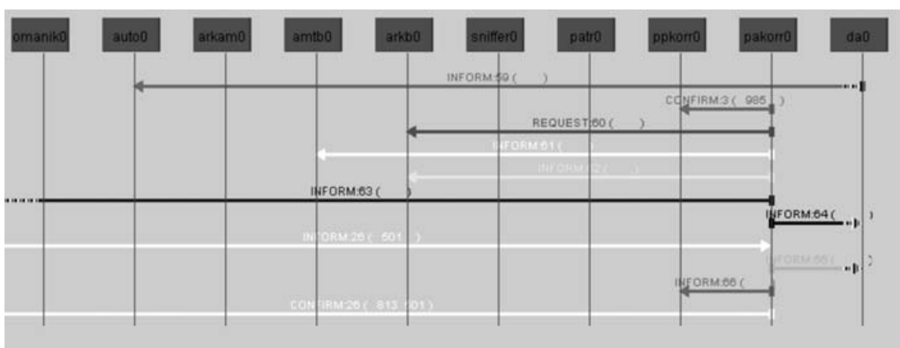

Figure 4.12 An interaction diagram of a MAS simulation

The complete agent model is a part of the component *A* of the model *M* and the simulation model is the component *B* of the model *M*. The completion of the agent model can be validated by testing. The completed and approved model is ready for multiple simulations and the demonstration of possible organisational behaviour in the chosen aspects and sub-parts.

4.5 THE ORGANISATION MODEL: SUMMARY

Sections 4.2 – 4.4.4 described how the components of the *organisation model* (*M*) are developed. The current section summarises the most important principles of the design of the model *M* and its components.

The *general description of the organisation* together with the *process table*, use case, activity and sequence diagrams, the specification of work processes by using the Q-model diagrams and the description of employees and structural units in agent class diagrams and multi-agent simulation diagrams form together the model *M* of the organisation *O*. The model *M* is a semi-formal description of an organisation and corresponds to the organisation at the moment of modelling. The overall process of devising the model *M*, described in the previous sections, can be summarised by using a picture as illustrated in Figure 4.13.

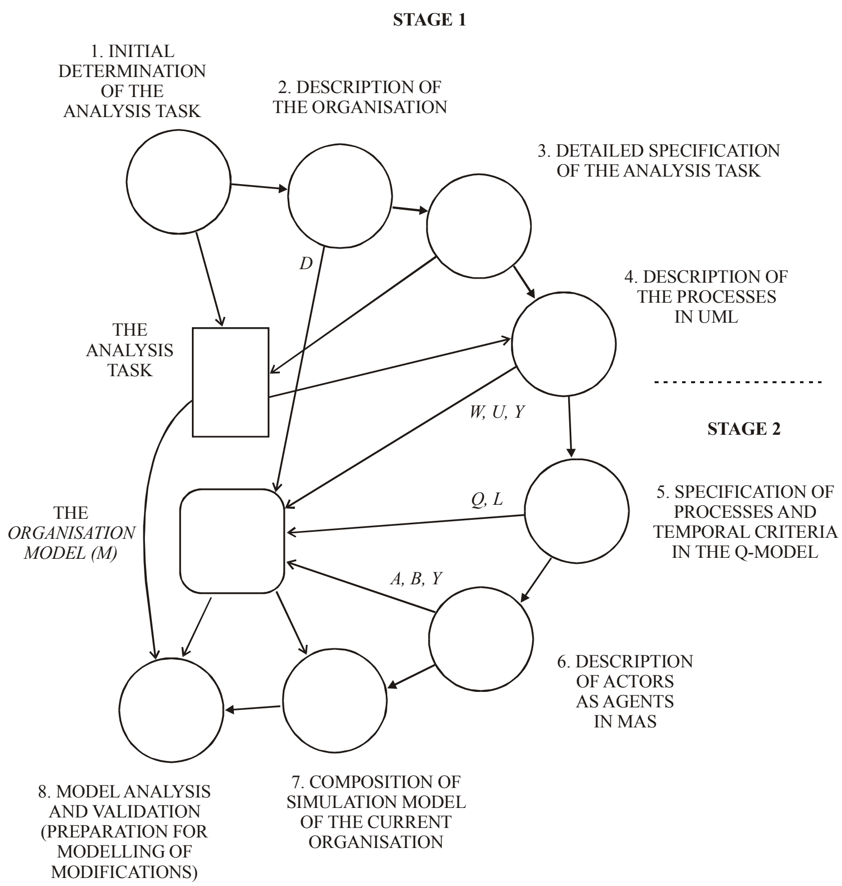

Figure 4.13 Process of devising the model *M*

The general process of devising the model *M* begins in the first stage of the methodology with the determining of the analysis task (document *H*) as well as the description (document *D*) of the organisation, detailed specification of analysis task, composition of process table and UML diagrams.

The second stage of the methodology is designated for the finalisation of the *organisation model*. Work processes, interactions and behaviour of actors are modelled in more detail by using additional tools and technologies (the Q-model and agent technologies). This stage consists of four steps:

- Step 5: modelling of timing parameters and interactions of work processes by using the Q-model,
- Step 6: modelling of actors and the behaviour of the organisation by using agent technologies,
- Step 7: simulations on the model and the improvement of the model,
- Step 8: analysis of bottlenecks, finding reasons why the existing situation is not satisfactory.

As a result, the model *M* of the organisation *O* consists of the following components:

1. General description and structural model (description) of the organisation (*D*).
2. Process table (*W*).
3. UML model of processes in use case and activity diagrams (*U*).
4. Description of work processes in the Q-model notation (*Q*).
5. Description of employees and structural units as agents by using class and sequence diagrams (*A*).
6. The UML model file (*Y*).
7. Catalogue (the model file) of organisation work processes models in Q in any simulation environment (*L*).
8. Catalogue of models of organisation as agents in any multi-agent simulation and modelling environment (or actual simulation) (*B*).

Components *A* and *B* may include both the modelling of specific persons and artificial agents as well as the modelling of both real and ideal roles / agents. The catalogue of models in Q-notation and the catalogue of models as agents are still separated from others since no unified simulation environment exists today for the UML, the Q-model and agents. Therefore also keeping of the parts together and updating should be done by using some document handling software or just by a modeller (a person or a team).

One of the aims for the simulation of activities of the organisation is to continuously specify the model in more detail until it corresponds to the expected activities of the organisation at the needed level of precision. As a result of simulations, it may be necessary to go back to earlier modelling steps in order to re-model by using the Q-model or to use case and activity diagrams and re-analyse the work processes and the process table. In this way the validation (i.e. checking the correctness) of the model is not difficult and can be easily reached during a deep analysis. The simulation should enable to test the work processes in a complete way. Since no unified simulation environment exists, validation and verification of Q-model processes may be reasonable to do in the step 5 and validation of the agent and simulation model in the step 6. Simulation and validation of the whole model should be done in the step 8 of the suggested methodology.

A more detailed description of the terms *validation* and *verification* will be presented as follows. The *model validation* is checking the correspondence of the model to the modelled organisation, i.e. checking whether the model exhibits the important organisational characteristics correctly and represents the behaviour of the organisation adequately. The *model validation* is the step 8 of the methodology. The validation can later be confirmed or rejected by experiments as an additional measure. Validation of the model is also continuous simulation and further improvement of the model. During simulation it may be necessary to go back some steps and make some parts of model more precise. Deeper experiments by using the model will be described in Section 4.6.

Verification is formal proving that the components of the model are formally adequate to each other and the model represents the organisation. *Verification* of different parts of the model and ensuring their correspondence is quite difficult and almost impossible. Some issues on the verification of agent and other models were considered in Section 3.5.

Model analysis and, resulting from this, organisation analysis is a deep, complex and time-consuming process. Some of the activities during the analysis of the model M (that is actually an analysis of the organisation by using the model *M*) include:

1. Are the goals compatible with the rule base?
2. Are the work processes (components *W*, *U*, *Q* and *L*) relevant to the goals and requirements?
3. Are goals of the structural units relevant to the overall goals?
4. How the employee’s goals are coherent with organisational goals?
5. How well does the organisation carry out its tasks?
6. What are the roles of employees (both ideal and real)?

The result of the model validation and the analysis of its behaviour is a common understanding of organisational processes and a reliable *organisation model* that can describe what likely happens in the organisation and why according to the dynamic input from the environment.

Table 4.7 Analysis of the model requirements for the *organisation model*

No	Requirement	How the model *M* meets the criterion
1	Encapsulating the most important organisational aspects	Components *D* and *W* encapsulate all important work processes, processes and their characteristics are analysed from different aspects – use cases and sequence of processes (use case and activity diagrams in component *U* and *Y*), temporal criteria and usage of resources (components *Q* and *L*), class diagrams and simulation of key processes (components *A* and *B*)
2	Hierarchy of business processes	Processes can be hierarchically presented in the process table, UML diagrams (components *U* and *Y*) as well as in the Q-model diagrams (components *Q* and *L*)
3	Goals, tasks, roles and resources	Goal representation and usage can be illustrated in agent class models that can also integrate representation e.g. in the form of Balanced Scorecard
4	Interactions and co-operation	Interactions and co-operation or competition are presented in MAS simulation and grounds for this are described in different UML diagrams, message exchange is also modelled in the Q-model
5	Relation of the existing information systems	Is specified during the description of the organisation, all other components include also artificial actors (i.e. active components of the information system), passive components of the information system are seen as means of communication
6	Modelling of development activities	Can be considered as other work routines but is not specially considered or supported
7	Straight-forward interpretation of modelling results into day-to-day activities	All components are in natural language (description *D* and process table *W*), the UML and the Q-model diagrams are visual with simple notation, in this way quite easily understandable for non-analysts and non-IT-specialists; Straightforward interpretation of analysis and modelling results is not supported at this stage of the methodology and will be considered in the third stage (Section 4.6)
8	Possibility of the persistent adjustment of the model	The methodology stresses the importance of the persistent updating of the model. However, there are two problems: first, there is no single tool for all components of the model, therefore different parts of the model should be updated manually; second, it is difficult to force updating of the model if no immediate benefit is seen from this activity
9	Modelling temporal criteria	Temporal criteria are considered in the Q-model (components *Q* and *L*) and simulated in a simulation environment (components *A* and *B*)
10	Comparing planned and real behaviour	This can be easily implemented in different sub-models of different components (e.g. UML diagrams), this can also be simulated in one diagram, using selector processes in the Q-model (components *Q* and *L*)
11	Observing all the essential behavioural aspects	Special aspects of the characteristics of employees, including the modelling of making choices and connecting goals with actual activities are described in the agent class models (component *A*) and can be simulated in the MAS simulation (component *B*).
12	Description of formal and informal communi-cation	Communication and message exchange are modelled at a detailed level in the Q-model and MAS simulation

The step 8 completes the second stage of the introduced methodology. The model *M* is tested and ready for experiments on simulating the organisation's behaviour in different circumstances. The *organisation model* already at this stage can be useful for several levels of process modelling, analysis and pre-planning. As known, simulations in the model are still much more economic than untested modifications in the real organisation. Well-determined modification plans support strongly respective modification of a process-assisting information system that might already exist in the organisation.

In principle, the model should be updated when changes occur in the organisation. This is necessary in order to guarantee a persistent quality level of the model and to help with easier modelling of new tasks. This stage may be one of most time-consuming until a satisfying solution will be found. The catalogue of models in Q-notation and the catalogue of models as agents are components of the model but for technical reasons they are still kept separately. Therefore keeping parts of the model coherent and updated should also be done by using some document handling software or just by a modeller (a person or a team).

The modelling stages are interactively interrelated. For example, additional information about work processes and behaviours may become known in the process of specifying the Q-model diagrams. If needed, elementary processes and component processes may be remodelled. The process table should be updated and kept in line with Q-model specifications.

In theory all processes described in the process table *(W)* should be modelled in the Q-model. In practice the model to be composed and its component processes may be different from *W* as it may be more reasonable. For an excellent solution to this problem a two-way automatic tool should be elaborated that would allow modification of processes by users in one of the description components (Tabel *W* or in Tabel *Q* in the Q-model notation) and afterwards can make automatic (possibly on-time) converting of data in both modules.

The result achieved so far is a conventional result in the modelling of organisations: we have reached a model of an organisation that describes this organisation from multiple viewpoints. Specific to the methodology is the *process table*, specification of temporal criteria in the Q-model, agent class models and multi-agent simulation. In Table 4.7 it is examined how the model *M* meets the requirements for models given in Section 2.1.

As can be understood from Table 4.7, in some points the model *M* does not correspond to all given requirements, especially with regard to supporting the planning of modifications. Therefore an extension of the model M is described in the framework of the current methodology in the next section.

4.6 MODELLING OF MODIFICATIONS: THE CHANGE MODEL

The second stage of the methodology results in the model *M*. As seen from the Table 4.7 (given in Section 4.5), the model *M* does not fully meet the requirements specified in Section 2.1. The model *M* does not support change management, thus an additional model called *change model* is introduced in this section for supporting the planning of modifications and change management. Devising a *change model* and possible modification plans is the aim of the third stage of the introduced methodology.

The third stage of the methodology comprises simulations on the reliable model *M* and testing of emerging behaviour according to the *analysis task*. The aim of the experiments (simulations) is to acquire more knowledge about the organisation. In order to understand how the organisation will behave if processes, actors and activities are modified or changed, the existing model needs to be modified (e.g. modifications in timing characteristics or changes in class models or methods). Each model modified in that manner will then represent a modified organisation.

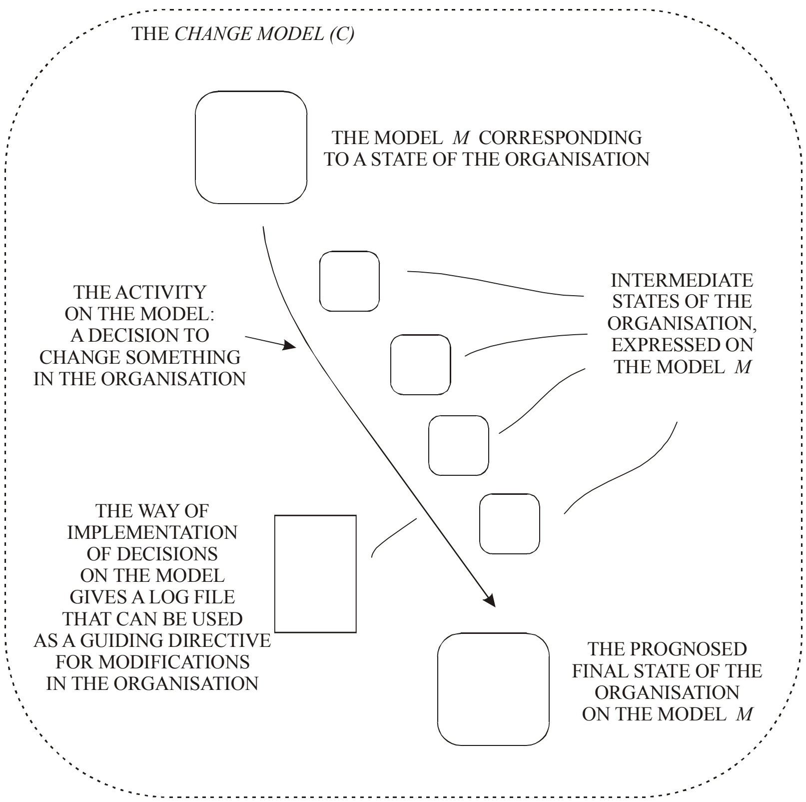

Figure 4.14 The *change model C* as a process

The initial stage of the model will correspond to the initial state of the organisation during each simulation session. The applied control activities and modifications of the model are recorded during simulations. It is supposed that during simulations we will finally get a modified model that corresponds to the desired state of the organisation. If chosen modifications are implemented stepwise in the model and the results are stored, we will get a partially ordered set of models that reflects how we made the particular changes.

As a result, the state of the model after each experiment session is stored together with the intermediate states of the model, if the model (e.g. some parameters of processes) is changed. This (partially ordered) set of sub-models for one modelling and modification sessions is considered as one combined model, called the *change model* (will be denoted as *C*). The *change model* is a collection of different states of model *M* during one modelling session. The model *C* is a semi-formal description of the behaviour of the organisation during experiments. It also implicitly stores control activities implemented on the model *M* during that modelling session. The *change model C* and the process of devising the model are illustrated in Figure 4.14.

For comparing different possible modification scenarios, the model should always be in the same initial stage. Therefore control activities performed and changes made on the model in different modelling sessions should be stored in different versions of models. This allows a later analysis of the behaviour of the model and a choosing of the best possible modification strategy.

The *change model* can be used for supporting the implementation of stepwise changes in the real organisation. This approach can be presented as the following.

For analysing the behaviour of an organisation (denoted by *O*), the model (denoted by *M*) of this organisation is needed. We are looking for mapping F_{OM} that establishes the correspondence between the organisation and its model with minimal efforts. The mapping F_{OM} is defined as a process of composing the model *M* of the organisation *O*. This can be presented as

$$F_{OM}: O \rightarrow M.$$

Equation 4.1

We can present the model *M* as a result of applying the mapping F_{OM} to organisation *O*:

$$M = F_{OM}(O).$$

Equation 4.2

In order to use the model for assessing the effects of modifications prior to introducing actual changes to the organisation we need a reverse mapping

$$F_{MO}: M \rightarrow O'$$

Equation 4.3

and, similarly to *Equation 4.2*,

$$O' = F_{OM}(M).$$

Equation 4.4

The mapping F_{MO} maps the characteristics of the model back to reality and assigns the parameter values and evaluation results obtained in a model to the appropriate structural units of the organisation. Hence the mapping F_{MO} describes the model analysing process and relates the events and phenomena studied in the model with the real world.

Let O_0 be the initial state of the organisation O and M_0 the *organisation model* that corresponds to the organisation at state O_0 so that

$$M_0 = F_{OM}(O_0).$$

Equation 4.5

Let then O_n be the desired state of the organisation after modelling and M_n the model that corresponds to the state of the organisation O_n so that

$$M_n = F_{OM}(O_n).$$

Equation 4.6

Modifying of the organisation and its processes is seen as a transition F_{CO} so that

$$F_{CO}: O_0 \rightarrow O_n$$

Equation 4.7

and we can also represent O_n as

$$O_n = F_{CO}(O_0).$$

Equation 4.8

Let us now define transition F_{CM} as the process of modifications of the model M from its initial state M_0 to another state M_n:

$$F_{CM}: M_0 \rightarrow M_n$$

Equation 4.9

That we can also represent as

$$M_n = F_{CM}(M_0).$$

Equation 4.10

Instead of implementing non-simulated changes on the organisation we can now improve the modification quality by introducing a three-step activity:

1. Devise a proper *organisation model M* (*Equation 4.1*): This is done during the first and second stages of the methodology.

2. Modify the model until the desired results are reached: implement transition F_{CM} (*Equation 4.9*). This is the third stage of the methodology.
3. Compose the actual change list in organisation terms for implementing the model in the organisation: implement F_{MO} (*Equation 4.3*).

These activities together can be seen as representing the transition F_{CO} by mapping F_{OM}, transition F_{CM} and mapping F_{MO} so that

$$F_{CO} = F_{OM} * F_{CM} * F_{MO}.$$

Equation 4.11

As resulting from this, the modified organisation can also be represented as:

$$O_n = F_{CO}(O_0) = F_{MO}(F_{CM}(F_{OM}(O_0))).$$

Equation 4.12

Simulations on the model *M* and stepwise modification can be represented as intermediate stages M_j of the model *M*. Process on modifications at each stage is expressed by a transition F_{CMj} so that

$$F_{CMj}: M_{j-1} \rightarrow M_j$$

Equation 4.13

and

$$F_{CM} = F_{CM1} * F_{CM2} * \ldots * F_{CMn}.$$

Equation 4.14

We can implement *Equation 4.11* step by step. Let *C* be the *change model* as defined in the beginning of Section 4.6. In this way *C* is to be considered as:

$$C = \{M_0; \{F_{CM\,j}; M_j\}_{j=1..n}\}$$

Equation 4.15

where

M_0 – *organisation model* that corresponds to the initial state of experiments

M_j – *organisation model* that corresponds to the resulting state of stage *j* experiments

$F_{CM\,j}$ – real control actions done on model at j stage of experiments on the model M_{j-1}

n – total number of steps in an experiment

By applying the mapping F_{MO} to each M_j we can represent the expected organisation O_j as

$$O_j' = F_{OM}(M_j).$$

Equation 4.16

This means that in addition to a partially ordered set of organisation models M we can also have a description of the corresponding states of the organisation. In this way the model *C* is a description of *M* states that can be converted into the terms of the organisation's states. As a result, we can implement the changes in the organisation stepwise, according to the model of changes and at the same time to observe whether the organisation corresponds to the expected organisation at this state. As the result, the final state of the organisation after implementing the modifications should be the desired state.

The transition F_{CM} as a whole, i.e. the way of the development of model *C* (according to *Equation 4.14*), is a list of modification suggestions. So F_{CM} is a state transition machine (see Meriste and Penjam, 1995) implemented on the model.

Organisation is then modified as

$$F_{CO} = F_{CO1} * F_{CO2} * \ldots * F_{COn},$$

Equation 4.17

where

$$F_{COj}: O_{j-1} \rightarrow O_j$$

Equation 4.18

similarly to *Equation 4.13*. As a result,

$$O_n = F_{MO}[F_{CMn}(F_{OM} \ldots F_{MO}[F_{CM2}(F_{OM}(F_{MO}[F_{CM1}(F_{OM}(O_0))]))] \ldots)].$$

Equation 4.19

The best solution to the improvement of the organisation can be composed of different scenarios. Decision-making persons should choose the most suitable scenario. This means that multiple *change models* will be devised for a modelling task. Each *change model* represents a single modification scenario.

According to the selected scenario and the corresponding *change model*, modification suggestions will be composed. This stage results in a sequence of suggested organisational changes that will hopefully lead to the desired state and behaviour of the organisation.

Since a *change model* describes the expected emergent behaviour during and after modifications, the *change model* can be used as a strong support material for preparing modification suggestions. Decision-makers and other employees see the benefit of the *change model* if its usage is easy enough, the models are correctly designed and represent understandable notations.

This approach has similarities to the idea of EKD (described in Section 2.3). M_0 can be considered as a business process model for the initial state, M_n represents a business process model for the desired state and F_{CM} is a change process model.

In Table 4.8 the model *C* is evaluated against the requirements given in Section 2.1. The comparison is similar to Table 4.7.

Table 4.8 Comparison of model requirements and the model *C*

No	Requirement	How the model *C* meets the criterion
1	Encapsulating the most important organisational aspects	Each model *M* encapsulates all important work processes from different aspects – use cases and sequence of processes, temporal criteria and usage of resources, class diagrams and simulation of key processes; process of introducing the modifications in the organisation (e.g. informing of employees, process of elaborating new guidelines, etc.) is not considered
2	Hierarchy of business processes	Processes can be hierarchically presented in *M* and *C*
3	Goals, tasks, roles and resources	The model *M* meets the criterion as described in Tabel 4.7
4	Interactions and co-operation	The model *M* meets the criterion as described in Tabel 4.7
5	Relation of the existing information systems	The model *M* meets the criterion, as described in Tabel 4.7
6	Modelling of development activities	The model *C* is considered to be a supportive tool for planning process modifications
7	Straightforward interpretation of modelling results into day-to-day activities	The model *M* meets the criterion partially. The model *C* gives also suggestions as to what modifications should be done and how to implement the modifications step by step; in this way the model *C* meets the requirements that the model *M* could not meet (as described in Tabel 4.7)
8	Possibility to persistent adjustment of the model	The methodology stresses the importance of the persistent updating of the model. However, there are still the same problems that were described in Tabel 4.7.
9	Modelling temporal criteria	The model *M* meets the criterion, as described in Tabel 4.7
10	Comparing planned and real behaviour	Behaviour related to processes is encapsulated by each model *M* and also by the model *C* as the overall process of the transition of the organisation.
11	Observing all the essential behavioural aspects	The model *M* meets the criterion, as described in Tabel 4.7
12	Description of formal and informal communication	The model *M* meets the criterion, as described in Tabel 4.7

As seen from Table 4.8, the model *C* meets all given requirements (also those that were not met by the model *M*).

Simulations on the model could be concluded in a formal document called *modelling report*. The *modelling report* consists of reviews of modelling stages with their results and a list of suggestions for modification. The third stage of the methodology, as illustrated in Figure 4.15, consists of two steps:

1. Step 9: Simulations on the model.
2. Step 10: Choosing the modification scenario and formulating modification suggestions.

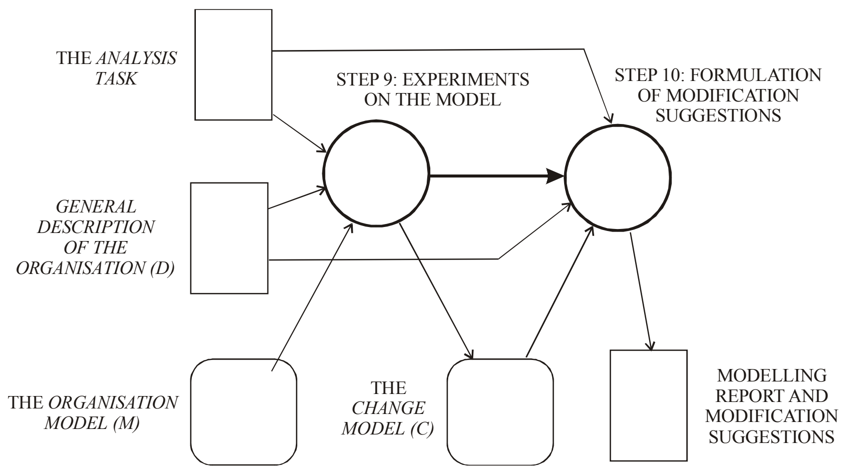

Figure 4.15 The third stage of the methodology

4.7 IMPLEMENTATION OF MODELLING SUGGESTIONS

This section will describe how the modelling results and modification suggestions will be implemented in the organisation. These activities form the fourth stage of the described methodology. Activities in this stage are not related to modelling but to change management in a real organisation – i.e. implementation of modifications in the organisation, monitoring and the evaluation of results. The implementation of modifications consists of five steps (illustrated in Figure 4.16):

1. Composing of the implementation plan,
2. Actual implementation in the organisation,
3. Monitoring of the organisation,
4. Analysis of results,
5. Evaluation of the successfulness of the current session and the decision about future needs for changes in modification.

This stage of the methodology is quite traditional and similar to any other general modification project in an organisation (e.g. transition phase in the

Unified Process). Mode detailed description and some special recommendations related to current methodology are given hereunder in this section.

The suggestions for modifications in the modelling report will be used as a basis for composing the actual *implementation plan*. The process of composing the implementation plan depends on the routines established in the organisation. The implementation plan itself has to take into consideration specific factors of the modelled organisation and should be as detailed as reasonable. It must answer the questions: who, when and what must exactly be done, why and what are the expected results. In most cases the implementation plan must be justified by the results of experiments, possible analysed risks and decided activities to prevent those threats.

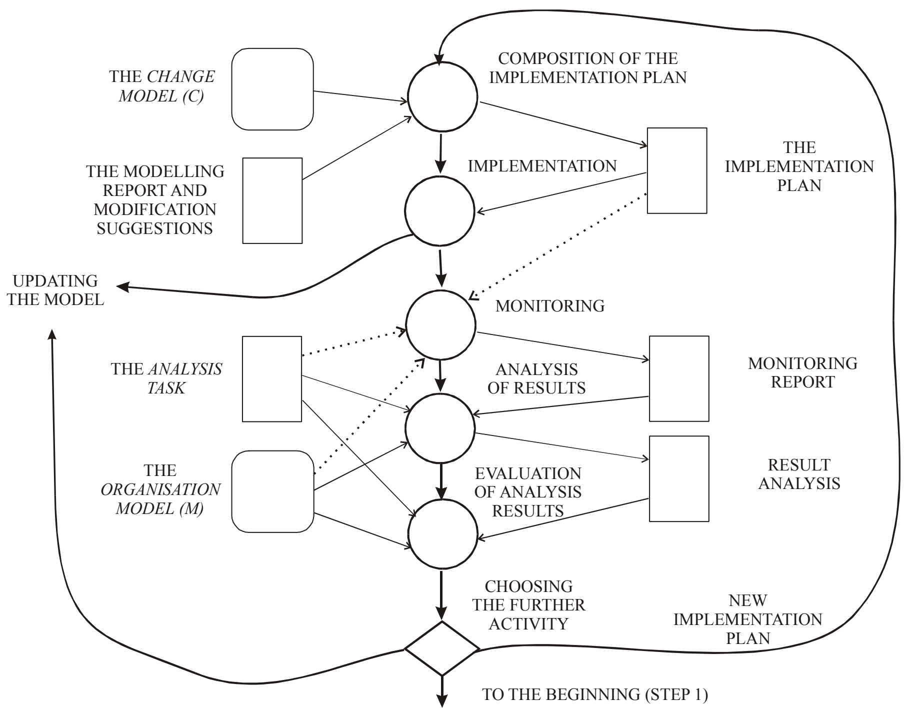

Figure 4.16 Detailed steps in the implementation stage of the methodology

The implementation plan differs from modification suggestions: modification suggestions are derived from the modelling results and they emphasize optimisation from the modelling point of view; implementation plans are based on organisational needs and priorities. Often the list of suggestions consists of modification proposals for reaching the desired performance in the organisation. The implementation plan is more conservative and takes into account the real needs of the organisation, when comparing the benefit gathered from the modifications with the cost needed for modifications. The modellers' and management opinions about the necessary cost are often different from each other.

The implementation plan can also be given as a table. Each row in the implementation plan describes an action to be done. For each action the following columns are used:

1. Action sequence number
2. Action name (string)
3. Action description (text)
4. Start time
5. Responsible actor
6. Additional remarks (free text, e.g. action purpose, duration, etc)
7. Expected state of the organisation, O_j.

The implementation of modifications in the organisation can be seen as a stepwise performing of the transition F_{CO} according to the implementation plan. The *change model C* describes the expected states of the organisation during modifications, supporting thus the actual changes. During the implementation it may be detected that the organisation behaves still differently from the model. This can be caused by:

- Errors in the model *M*.
- Errors in the model *C*.
- Errors in mapping F_{MO} or in the list of modification suggestions or
- Dynamic changes in the organisation during the time when modelling was done and the model was not updated.

In such case it is important to make corrections in the corresponding model and simulate the new situation. This may cause changes in the model *C* and in the implementation plan.

Monitoring of the actual performance of the organisation will follow the implementation of modifications. The period of monitoring usually depends on the extent of changes and varies from one month to six months, in exceptional changes until one year. If necessary, amendments should be made to some parts of the model. The analysis of the results of specific modifications is important for the evaluation of specific changes and for a better planning of further specific changes.

The effectiveness of the analysis, the modelling and modifications of the organisation should be evaluated. It should also be evaluated whether the real results corresponded to the expected ones (that were predicted by the models *C* and *M*) or not. If not, then possibly why? Was it caused by differences in implementation (in this case it is acceptable from the view of modelling) or by errors in the model (which is an unsatisfactory result).

Modification concludes with the decision of what actions to take in the future (possible activities are also illustrated in Figure 4.16):

a. Compose a new implementation plan – go to step 11 in the beginning of the fourth stage,
b. Devise a new *change model C* – go to step 9, or
c. Start the analysis from the beginning – go to step 1 of the methodology and devise a modified analysis task.

In everyday life some changes are always planned in the organisation. It is suggested that all changes should be first simulated and analysed on the model before implementing the modifications in the organisation.

The modelling approach stresses the importance of the permanent updating of the *organisation model* in order to keep it in line with the organisation. Regular updating of the model allows to use the model also for smaller decision-making activities during everyday management process. It also allows the reuse of the model with less effort. Further modelling can be iteratively more precise and there is no need to start from the scratch. The modelling can be based on the quality of modelling done previously. This reduces re-modelling costs and guarantees stabile quality. Model updating enables to start new modelling by using the already actual model. This also facilitates stable modelling quality in different modelling sessions.

On the other hand, permanent updating of the model is questionable in practice since

1. Model updating and continuous maintenance is time and resource demanding.
2. It is difficult to justify the need for spending time and resources for a task which use is not clear.
3. Usefulness of the model is hard to predict since future modelling may emphasize other actors and units than the previous modelling. This means that the existing model does not possibly cover the necessary units and we must build a new model.

4.8 SUMMARY AND EVALUATION OF THE METHODOLOGY

This chapter introduced a novel methodology for modelling time-sensitive emergent behaviour in human organisations. The proposed methodology consists of four stages. The stages, in turn, consist of steps. There are the total of 15 proposed steps in the methodology as illustrated in Figure 4.17. The figure presents also the most important models and documents that are devised and used in different stages of modelling.

Permanent actualisation of the model and its re-use means that the overall lifecycle is like a spiral where each implementation (planning of modifications, modelling and implementation of modifications) is just an iteration in the implementation of the *organisation model* and one circle in the lifecycle of the methodology. This means that after finishing step 15 the work will continue from step 1, but ready-made models are not thrown away but concretised stepwise. The spiral way of lifecycle is also used in several methodologies.

The suggested methodology can be considered as an EM (enterprise modelling) approach (or an addition to any EM approach). It uses a set of sub-models, similar to an EM approach: goals model, business rule model, actors and resource model, concepts model and business process model. Some models can be imported form other EM modelling tools or exported for an additional analysis.

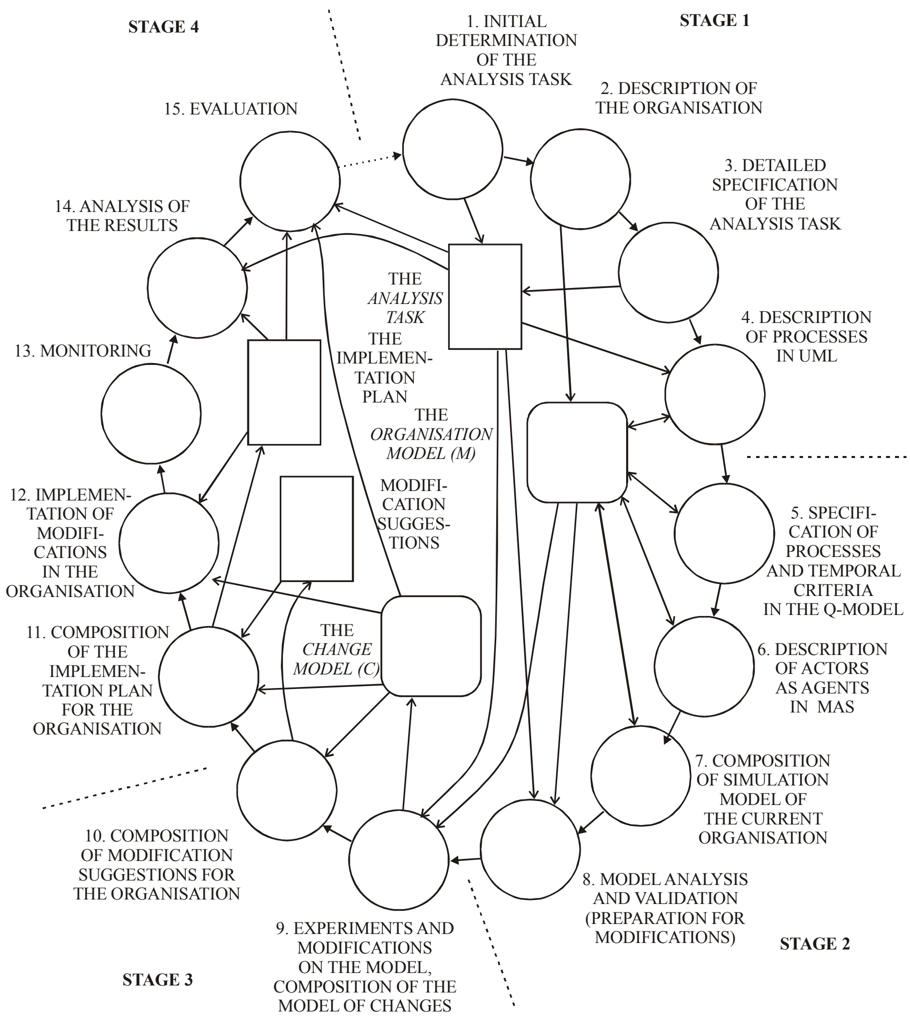

Figure 4.17 The overall illustration of the methodology

Correctness of the initial model *M* is checked together with the correctness of model *C* during the implementation of modifications. In this way the actual validation can be done on a strong real basis. Verifying different parts of the model and ensuring their correspondence is quite difficult and almost impossible. Some issues on the verification of the agent and other models were considered in Section 3.5, correspondence and transition of different components of the model were presented when introducing the methodology in Chapter 4.

Table 4.9 List of documents and models composed during modelling

Se-quence number	Name of document / model	No-tation	Composed in step	Mainly used in step(s)	Description and remarks
1	2	3	4	5	6
1	Initial analysis task		1	3	
2	The description of the organisation	*D*	2	4-6, 14,15	
3	The analysis task (final)		3	4, 8, 9, 13-15	
4	The process table	*W*	4	5-15	Later, after step 6 is used as a part of *organisation model* M
5	Description of the organisation in the UML: use case and activity diagrams	*U*	4	5-15	The model file (e.g. .mdl)
6	Description of processes in the Q-model	*Q*	5	6-15	
7	Diagrams of processes in the Q-model modelling environment	*L*	5	6-15	
8	Description of class models and agents in the agent technology	*A*	6	7-15	
9	A simulation of a case	*B*	7	8-15	
10	Completed *organisation model* (consists of *D, W, U, Q, L, A, B)*	*M*	2-7	8-15	*M* is also a part of *change model C*
11	The *change model*	*C*	9	9-12	
12	The modelling report and suggestions for modifications		10	11	
13	Implementation plan		11	12, 14, 15	
14	Information and results of monitoring		13	14	
15	Analysis of results		14	15	

Table 4.10 The comparison of modelling methods, reviewed in the thesis

No	The property	PERT	Best Value	BSC	EKD	TEBA
1	The aim	Planning of actions and their dependencies, required time in a project	Evaluate the characteristics of the service level (standardised comparison to other participating organisations with similar duties)	Evaluate the organisation's performance (how well goals are supported, what the existing and necessary activities are)	Supportive method for preparing and implementing changes in an organisation by incorporating the knowledge of all participants	Analyse and model the organisation (its processes and behaviour)
2	The scope	A project	Organisation's performance	Organisation's performance	A big project or organisation to be restructured	The whole organisation
3	Necessity of a computer aided tool	No	No	No	No	No
4	Existing tools	?	?	No	No	No
5	Strengths	Task dependencies	Common criteria for a set of participating organisations, comparison, placement	Integrated approach for all aspects of activity	Capturing of all important existing knowledge	Capturing different aspects (processes, interactions, intentions)
6	Weaknesses	Used for a project only, a pure list of activities on a time-scale	No analysis for an organisation	Too much management and financially oriented and too little process-oriented	Voluminous	No unified modelling environment exists, voluminous (less than EKD)
7	Consideration of time issues	Yes	Can be used as a performance indicator	No	If considered	Yes
8	Motivations, attitudes	No	Can be monitored and evaluated against targets	No	Included as opinions from different knowledge sources	Yes

In the process of modelling some documents and models will be composed. Table 4.9 lists those main documents and models as well as describes where they are mostly used.

Table 4.10 displays the comparison between the modelling methods, reviewed in the thesis in comparison with the proposed modelling methodology (TEBA – modelling of Time-aware Emerging Behaviour using Agents). The same properties are presented here that were used for the comparison of different methods in the previous chapters. As it is seen, TEBA is suitable for modelling time-sensitive emergent behaviour.

5 CASE STUDIES

In order to illustrate the implementation of the introduced methodology some case studies are presented in this chapter as examples. The main example (given in Sections 5.1 – 5.4) is taken from the law enforcement domain. The example concentrates on processes related to information exchange in the police with regard to vehicle theft. This example is related to the modelling of multiple tightly related processes with binding temporal requirements and interactions between a number of actors. The final outcome of the processes in the current example (whether the activities of the police are successful and the vehicle will be found or not) depend very much on real choices of key actors and their response time. Such examples can be considered as a possible typical implementation of the suggested methodology.

For research purposes the given example is extended in Section 5.5. The extension concentrates on differences on the estimated behaviour with and without taking into account possible motivations, personal interests and competitive tasks.

In addition, two smaller examples are also presented at the end of this chapter (Sections 5.6 and 5.7). The examples illustrate the determination of responsibilities and co-operation between structural units in the organisation. The first of these case studies concentrates on the analysis of how development and maintenance tasks can be allocated between structural units in an organisation. The last case study looks for a solution of how a project organisation within the existing organisation should be set up for the development of an information system. In both of these case studies the implementation of the methodology was purely an additional activity that was used for checking whether the previous organisational decisions were justified or whether they could be improved. The shorter case studies are presented here for illustrating also other application possibilities of the introduced methodology.

Conclusions about the implementation of the methodology are drawn at the end of the chapter (Section 5.8).

5.1 EXAMPLE ON VEHICLE THEFT: BACKGROUND

The police is a very suitable organisation for taking it as an example for operational emergent behaviour since the police organisation demonstrates significant characteristics of emergent organisations:

- The main activities of the police are performed by an individual actor or a group (e.g. patrolling or investigation of crimes) in a highly dynamic environment.

- Teamwork and correct timing of actions play an important role.
- Operational situations are sophisticated; the information for the evaluation of the situation is often insufficient.
- Communication between actors inside the organisation and with actors outside the organisation (e.g. persons in the society) is an unavoidable component of the daily routine.
- Oftentimes a decision must be made and actions taken in minutes and seconds.

There exist pre-described rules that determine which action is preferred in each specific situation. The recognition of the situation and the responsibility for a correct response to the sometimes rapidly changing environment has remained to be the task of a corresponding actor (i.e. a person who is involved in a particular situation).

The modelling task concentrates on the police activities in registering and exchanging information in cases of vehicle theft. The example is based on real analysis (Savimaa, 2003a), the task was set up in the autumn 2002 and the analysis was performed from October 2002 until March 2003. The presented example is simplified in order to fit into the space of the present thesis and to offer the overall illustration of the model and the methodology, but the simplifications do not affect the main issues. The work processes described hereunder correspond more or less to the situation of that time. The modelling is limited to the activities that are directly related to the case of vehicle theft. It is also assumed that all actors are conscientious and perform their duties in the most possible way. As a result, the case study represents the analysis of a set of integrated workflows. Additional emerging behaviour, caused by the fact that motivations of actors may vary and their decision-making is free, is considered in Section 5.5.

According to the police statistics 2230 vehicles were stolen and 1368 found in Estonia in the year 2002. When a vehicle is stolen, one of the four typical scenarios is usually followed:

- there may be an attempt to re-register the vehicle in the vehicle registration office,
- the vehicle may be taken abroad,
- the vehicle is dismantled for spare parts, or
- the vehicle is used for driving inside the country for some time, after that the thief chooses behaviour 1 – 3 or abandons the vehicle.

When the owner notices the vehicle theft (s)he informs the police in person or by phone. A police patrol arrives to the place of theft and takes also the formal statement about the vehicle theft (if the formal statement has not reached the police yet). Then the command and control centre of the police performs all necessary activities and informs all the other police units, the vehicle registration centre and the border guard. To prevent illegal actions with the stolen vehicle, it is important that all necessary authorities will have corresponding information as soon as possible.

According to the analysis of the police statistics it becomes evident that the longest delay in the information exchange is caused by the fact that the owner notices the car theft usually some minutes to several hours later. Further information exchange, after the police confirms the case, takes minutes. Mistakes arise seldom, for example a vehicle wanted by the police is re-registered later in the vehicle registration centre, or the border guard stops a vehicle that is not anymore wanted by the police. This indicated that the information exchange procedures needed further systematic analyse.

The optimality of information exchange and continuous availability of all necessary data has not been analysed for a long time. Current work routines were implemented separately, according to different aims, priorities and possibilities.

The modelling task of the case study is to evaluate how operative the information exchange is from receiving the initial information about the vehicle theft until forwarding the information to all necessary databases in the police, the vehicle registration centre and the border guard. What suggestions can be given for modifications of related work processes? Are there duplicate work processes in the information exchange and if yes, what are the reasons for this? Is it possible to speed up the information exchange? Can the accessibility to the information be improved?

The analysis task (illustrated in Table 5.1) is given in the form of the prioritised list with a very brief description of the situation, expected outcome and criteria for evaluating how well each sub-task is fulfilled (as described in Section 4.2.1).

The police organisation is shortly described hereunder. The police are an executing authority of the state. Its main tasks are to ensure public order and to investigate crimes and administrative offences. The central authority of the police is the Police Board. A police prefecture is a regional police authority that performs all police tasks in its area of duty (region).

Several laws and normative acts regulate the activities of the police as permanent activity boundaries. Goals are determined on the basis of development programmes in this domain (elaborated by the police and by the ministries of interior and justice) and priorities for the year. Inside the police organisation decrees of police director general and police prefects, methodical guides and one-time regulations can be considered as establishing dynamic boundaries. All internal decrees are formally obligatory and have equal priority. Choices are often made in real-life situations where it is difficult to fulfil all regulations; these are also sometimes contradictory since they are enacted at a different time.

The police communicate widely with the environment, mainly with the population and companies. Communication with other law enforcement authorities in the state and abroad is also frequent. Actors outside the police act in different roles like the announcer, the victim, the witness, the suspect, the accused etc.

Table 5.1 The analysis task

Prio rity	Problem name	Short description of the problem	Existing situation	Desired result	Criteria for evaluation of the result
1	Opera-tivity of infor-mation ex-change	Evaluate whether the existing infor-mation exchange routines are optimal enough	Not known, the existing situation is to be described during the analysis	Time criteria for infor-mation ex-change are determined in all stages	Preciseness of the description, accu-racy of the model and a number of correctly specified temporal criteria
2	Dupli-cate pro-cesses	Evaluate whether there are duplicate processes during information exchange	No known existence and justification of duplication	The purpose and necessity of all related processes are known and evaluated	Exactness of the model, correct-ness of necessity evaluation
3	Infor-mation ex-change speed	Evaluate whether there are possibilities to speed up information exchange and if yes, give recommendations	Information exchange seems to be optimal, but the analysis is not made	There exist time criteria for infor-mation exchange and estimation about additional resources	Preciseness of the composed review
4	In-time avail-ability of infor-mation for police patrols	Evaluate whether the information in police databases on stolen vehicles is available on time for police patrols	Analysis does not exist	Analytical evaluation, simulation and comparison with practice	Detailness of review
5	Bottle-neck analysis	If possible, evaluate bottle-necks for the availability of information and give improvement suggestions	Missing	Composed list with the des-cription of bottlenecks, presence of improvement suggestions	Exactness of the description, usefulness of improvement suggestions and correctness of related criteria

The internal structure of the police can virtually be divided into law enforcement police (uniformed police) and criminal police. The main tasks for the law enforcement police are ensuring public order and investigating administrative offences, also investigation or pre-investigation of some (more simple type) crimes.

In this case study only some activities of the law enforcement police are reviewed. These are work processes of a duty officer in a police prefecture (local command and control centre, LC); work processes of a duty officer in the command and control centre of the police board (CC) and work processes of a patrolling officer (PO) in the police prefecture. The simplified hierarchy of the analysed roles is illustrated in Figure 5.1. The continuous arrows describe the control hierarchy and dotted arrows the main communication (information exchange). The analysed roles (LC, CC and PO) are marked bold. The roles of a constable are considered to be similar with PO in the current case study. People (e.g. announcers or statement givers, victims, etc) are communicating mainly with the duty officer, patrolling officer or constable. Employees on other posts are solving mainly intra-organisation tasks.

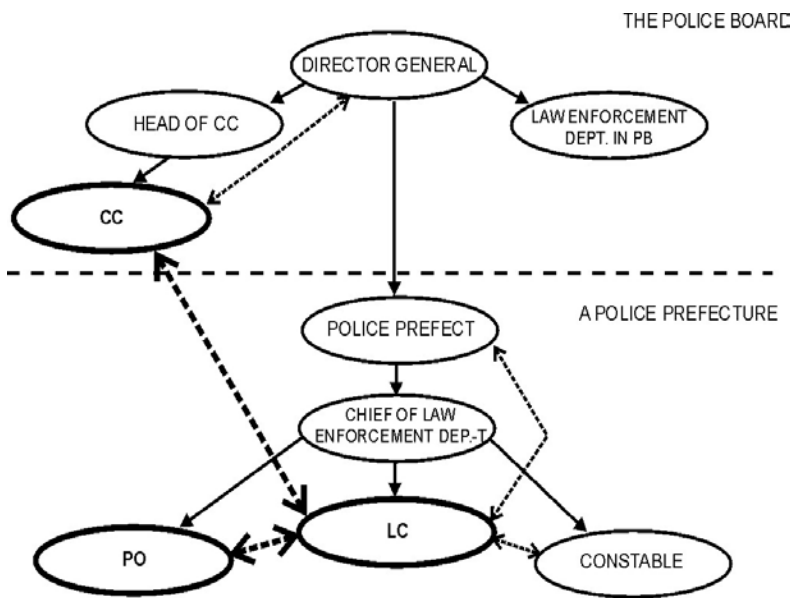

Figure 5.1 The hierarchy of roles in the police organisation.

5.2 DESCRIPTION OF PROCESSES

Information exchange in the vehicle theft case concerns LC, CC and PO from the police, the vehicle registration office (VRO) and the border guard (BG). The corresponding processes can be collected into two sets:

1. Processes related to receiving and registering the event in the LC, informing CC, VRO and BG and updating all necessary databases:

POLIS database and the database of stolen vehicles in the police, the vehicle register (VR) in VRO and the border guard database (BG db).

2. Processes of a patrolling officer (PO) (or constable) related to the actual checking of a vehicle and persons as well as performing related queries in databases by the PO and the LC.

Table 5.2 Some work processes of duty officer

ID	Activity	Input description	Description of activity	Duration	Output description
1	2	4	6	7	9
II	Receiving of calls and controlling of patrols				
J2.2.A	Receiving a call about the vehicle theft	Call	Answering the call, getting information, making decision	2..3’	Information to be used
J2.4.A	Registering the information about the vehicle theft	Call, info, decision	Registering in information system POLIS as initial information, checking the data in vehicle register	3’	Correctly entered information
J2.6+ J2.7	Forwarding the call information to a patrol	Decision, choosing the patrol	Calling the patrol, giving the task, determination of priority, continuing to observe the solving process	3’	Patrol starts solving the problem
J2.9.A	Re-registering the information to an event	Statement confirmation	The initial record is re-classified to “event” and information is sent (“send” button pressed) to CC	3’	Correctly entered information
J4.1	Receiving the statement and communication from statement giver	Statement, person	After received a personally given statement, LC also performs all necessary actions to start the case	20’.. 30’	Entered event
III	Answering to PO queries				
J3.1, P2.17, K4.8	PO or constable query by radio	Radio call	Presenting query about person and / or vehicle	30’’	CC understands the question
J3.4	Query analysis	Query, knowledge about sources	Choice of sources	10..30’’	Sources chosen
J3.5	Checking the information in the database	Query	Performing (partial) queries	2..3’	Answer to the query
J3.6	Presenting the answer to PO or constable	Answer to query	Answering to the query	30’’	Answered query
V	Information exchange with CC				
J5.3	Passing information about stolen vehicles to CC	Information about vehicle theft	Entering the information into POLIS and pressing the “send” button, sends an e-mail to CC	ca 10’	Entered and passed information

In the following description the processes are grouped according to the actors. First, processes of the LC and CC are presented and then related processes of VRO, BG and PO are reviewed. At the beginning processes of each actor are introduced in general, then in the form of Tabel and finally by use case and activity diagrams. Simple use case diagrams show single actions and then these diagrams are combined for the more general review of all related processes. Activity diagrams illustrate the sequence of related activities for performing certain tasks. Many diagrams are omitted compared to the original analysis in order to fit into the space.

The local command centre (LC) is the 24-hour acting centre for organising the operative police work in a police prefecture. It has a wide range of tasks that are quite well regulated by the decrees of the Police Board and the Ministry of Interior. These regulations are generally correct and used as the basis for choosing work activities, but do not always correspond to the real life because of continuous changes in the activities. Available resources have to be used for solving multiple concurrent tasks. The current analysis scope covers a subset of activities of the LC, more precisely:

1. Registering of received information in a prefecture, operative reacting to it and exchanging the information with patrols, constables and CC;
2. Controlling police patrols in a region.

An example of the work processes related to the stolen vehicle is presented in Table 5.2. In the first column of Tabel a structured identity number of a process is given, as it is used in all tables and diagrams during the modelling. For example, “J2.4.A” means that this is a modified version (“A”) of the fourth process (“4”) in task 2 of actor LC (“J”). The duration of processes is given in minutes in the column 7 of Tabel. As for all specified processes the value for input time criterion is “on start”, the output validity criterion value is “when finishing, lasting” and the actor is LC, columns 5 (used for specifying the process activation frequency and input data consumption time), 8 (actor) and 10 (output validity time) as well as columns 3 (purpose of action) and 11 (remarks) are omitted in the given table compared with the original table presented in Section 4.2.3 (Table 4.2).

The reviewed tasks of a LC consist of several sets of work processes. In the following there are use case and activity diagrams, starting with simple ones and later combined use case and activity diagrams are presented. The review is given in the following order:

1. Receiving a call
2. Checking the received information and registering it in the database
3. Re-registering the information about the stolen vehicle to ‘event’ and forwarding it to the CC

Actors involved in the process of receiving a call are the announcer / statement giver (by telephone or personally) and a LC as the receiver of this call or statement. The process of receiving a call or statement about a stolen vehicle (activity J2.2.A) can be seen as a specialization of the general activity of receiving calls and statements (activity J2.2). This is illustrated in Figure 5.2.

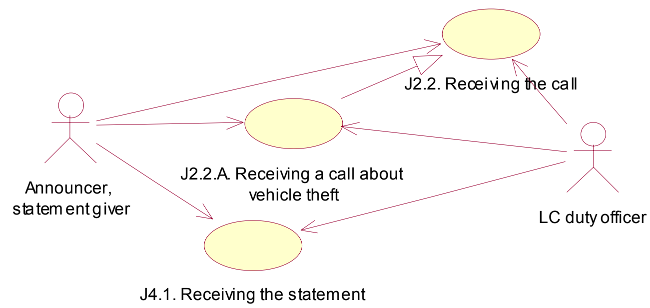

Figure 5.2 A use case diagram related to receiving a statement in LC

After receiving a statement or a call, the information is checked and registered in the database POLIS (a police information system of event and case management) as initial information. These processes require at least four actors: LC duty officer, POLIS database, Vehicle register and other possible databases (seen as one actor). The corresponding use case diagram is given in Figure 5.3.

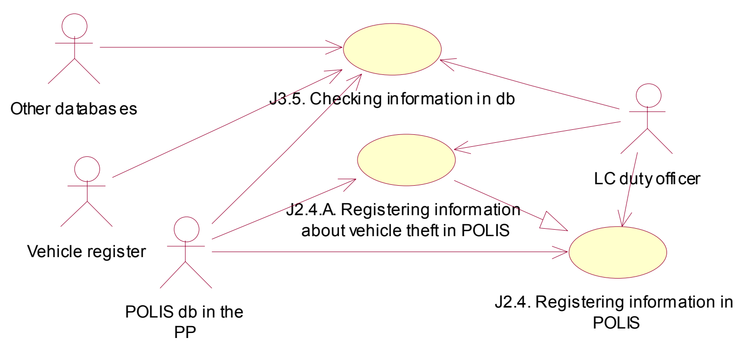

Figure 5.3 A use case diagram related to the initial checking and registering information about stolen vehicles in LC

LC sends a police patrol to the place of event and if the initial information is confirmed by the patrol, it will re-classify the information in the POLIS database under the class 'event' and then forward it to CC via e-mail system. Related use cases are given in one diagram in Figure 5.4.

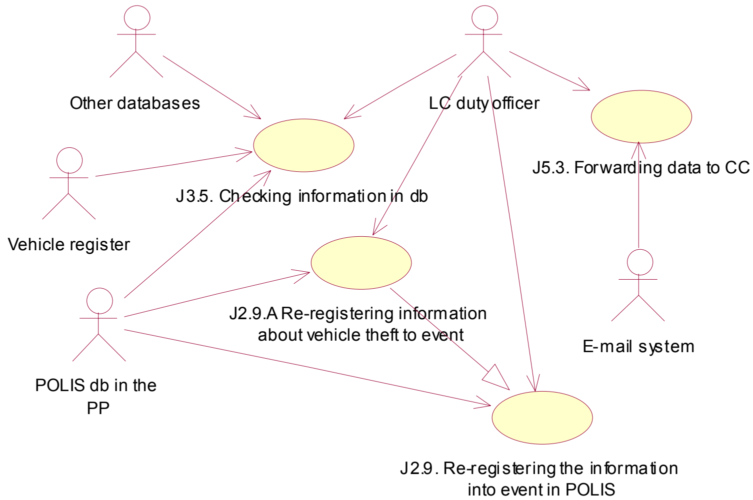

Figure 5.4 A use case diagram of event registration and information exchange

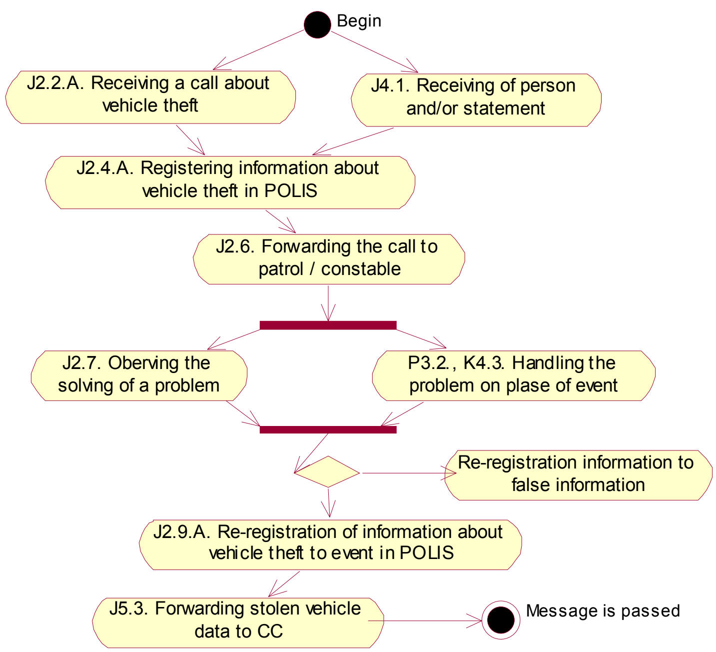

Figure 5.5 An activity diagram of event registration and information exchange about stolen vehicles

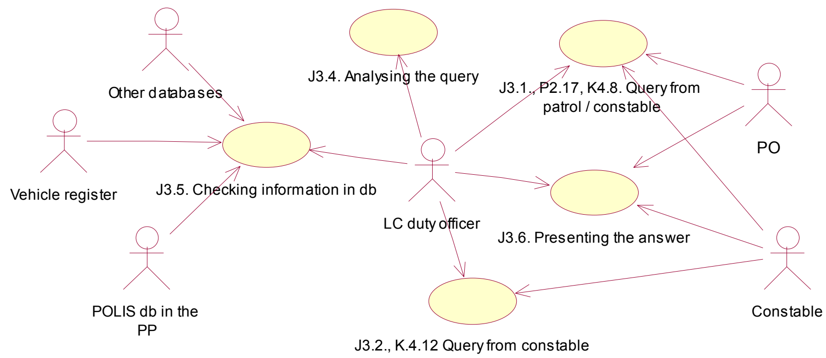

Figure 5.6 A use case diagram of patrol queries and answers of LC

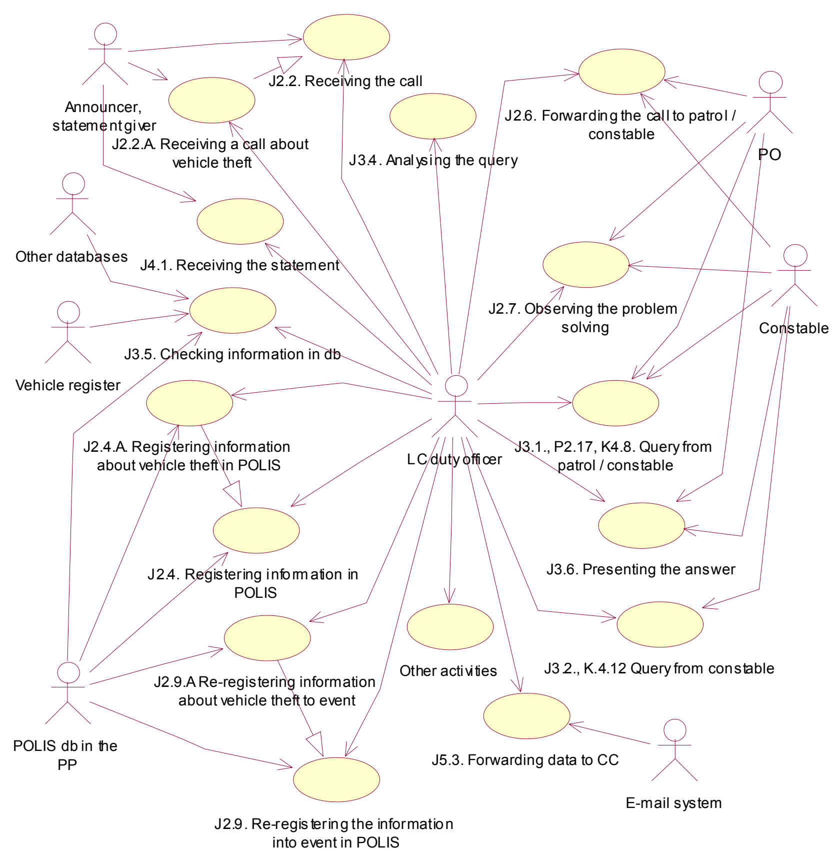

Figure 5.7 The combined use case diagram of LC activities related to stolen vehicle

An activity diagram given in Figure 5.5 illustrates the sequence of activities in the process from receiving information about vehicle theft until forwarding the information to the CC. The activity diagram contains the same processes of LC, described in Tabel 5.2 and in use case diagrams, processes of PO are added.

The following use case and activity diagrams describe processes of LC that are related to serving PO query. Usually PO or constable presents a standardised query about a person or a vehicle by using police radio. The task of the LC is to find answers from databases (e.g. exact personal data, previous violations, existence of driving licence, vehicle identification data and whether it is reported stolen or not) and report the results back to PO or constable, respectively. On the use case diagram (illustrated on Figure 5.6) four stages of the process are represented: the first stage is receiving the query, the second – analysing and optimising the query, the third stage is making queries from databases and finally the fourth stage is reporting the results back to PO (or to constable). Because of its simplicity, the corresponding activity diagram is omitted.

The processes of LC, given previously in multiple use case diagrams, can be presented by using a single combined use case diagram (given in Figure 5.7).

The tasks of **the central command centre (CC)** in the case of the vehicle theft are:

1. Receive the message (by e-mail).
2. Check the data in the vehicle register (again, since CC is responsible for the official information given out to VRC and BG and there is not enough trust in the activities of LC).
3. Enter the information into the database of stolen vehicles and forward it to the vehicle registration centre and to the border guard by using fax and inside the Police Board by using e-mails and telephone. Later (3 times per day) the information is also automatically forwarded by the database engine to update databases on stolen vehicles in the police prefectures and the border guard.

Work processes of **the vehicle registration centre (VRC) and the border guard (BG)** are reviewed only in relation to the present case study. In bureaus of the vehicle registration office (VRO) where vehicles are registered or re-registered, an operator marks a vehicle stolen according to the fax received from the police. Information about stolen vehicles is entered into the border guard (BGO) information system from police faxes and (later) by database updates. In both authorities vehicles that were found to be stolen and related persons are detained and the police informed. An activity diagram of BG processes is given for illustration on Figure 5.8. The diagram conceals the possible activities: updating data in databases, automatic updating of data in databases (by the database engine) and processes related to vehicle border crossing. In this activity diagram all activities of BGO are seen as the activities of one actor. Processes in VRO are similar and are not illustrated in more detail.

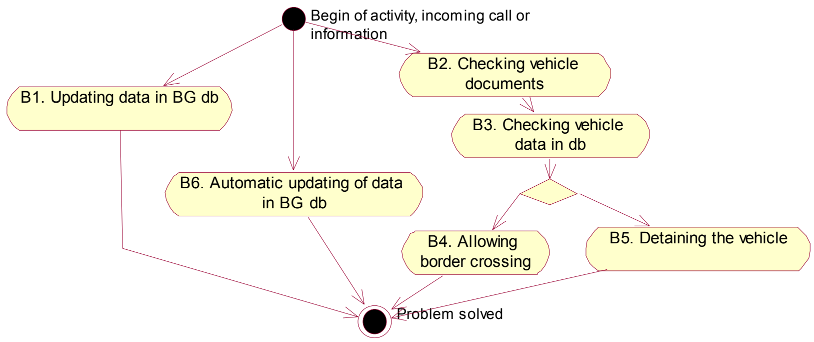

Figure 5.8 The activity diagram of BG, activities related to information exchange and vehicle checks

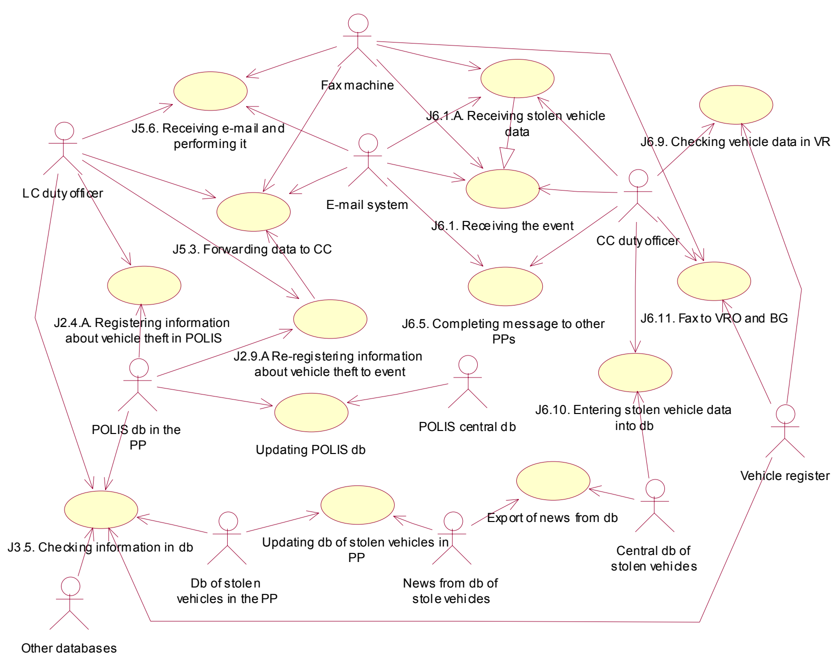

Figure 5.9 The use case diagram of information exchange

A special analysis of **activities related to information processing** has been missing so far. Information exchange is observed only in relation to single problematic cases and as a result, mistakes are sometimes detected. Those mistakes are avoided in the future by specific measures only. The system of information exchange is normally modified only if the existing solution provides incorrect information often enough (according to the view of users) or when the technology has to be changed. This means that systematic review of information exchange is done very seldom. The use case diagram, illustrated on Figure 5.9, presents actors and uses cases from information exchange perspective. The selected focus allows an easier detailed analysis of information exchange.

The activity diagram in Figure 5.10 describes the overall sequence of processes from receiving the initial information until updating all databases.

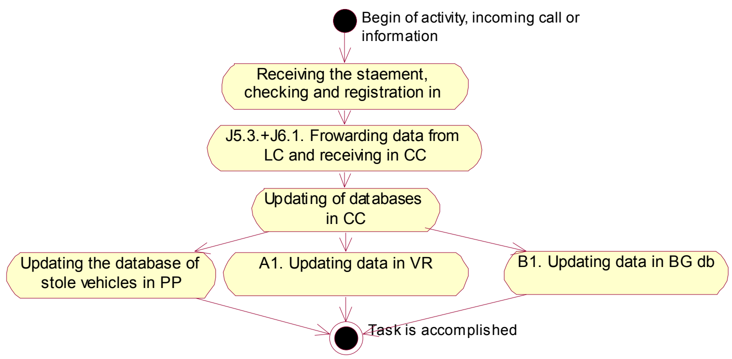

Figure 5.10 The activity diagram of event registration and information exchange

The role of **police patrol officer (PO)** is to ensure public order and safety of people by preventing and reacting to crimes and administrative offences. PO activities, tasks and methods are regulated by the decree of the Director General of the Police. PO is given operative tasks by LC. PO activities are typically concurrent, (for example, patrolling, serving the call on the place of event, checking the vehicle and detain the suspect) whereas usually one of these may be fulfilled at a time. Patrolling may include other sub-activities, e.g. ensuring public safety or driving safety or serving the call (that, in turn, comprises, among other activities, also driving to the place of event and communication with the people in different roles -- witness, announcer, suspect). While checking the vehicle, the PO makes a request to LC to receive data that enable to identify a person or vehicle. LC checks the data in different databases and reports the answer to PO who then decides to release the vehicle or detain the vehicle and persons.

A simplified activity diagram, related to vehicle checking by PO, is given in Figure 5.11.

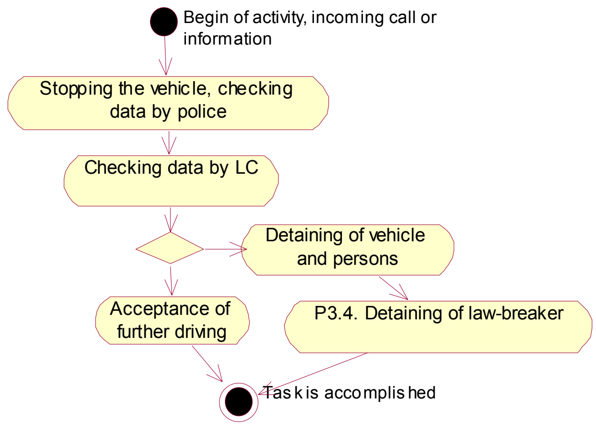

Figure 5.11 A simplified activity diagram of patrol queries

The overall process as a combination of activities and processes of different actors is the following. In general, two subsets of processes can be distinguished:

1. Processes related to the event of vehicle theft, informing the police and all related activities and information exchange databases of the police, VR and BG.
2. Checking the vehicle by the police, BG or VRO, queries to databases and corresponding activities based on the results of queries.

Use case and activity diagrams presented previously in the current section are combined here and some new processes are also added (e.g. vehicle theft itself). The general process for the present case is the following: the process starts from the event of vehicle theft and informing the police by the owner. The process continues with information exchange within the police and updating of databases. Vehicle checks by the patrolling police officer and requests to the databases by the employees of the police, the vehicle registration centre or the border guard will eventually lead to detaining the stolen vehicle if the stolen vehicle is checked. Alternatively, if databases are not updated, then the police, VRO or BG may not detain the stolen vehicle (or detain a vehicle that is already officially found and returned to the owner).

For the complete picture the previously described work processes and use case diagrams are combined. The detailed presentation of the overall picture is too complex and gives too little information. Therefore the overall pictures are simplified and only the most important aspects are described (e.g. processes related to checking the paper correctness of documents etc are omitted). Use case diagram is omitted in the thesis but the activity diagram is given in the Figure 5.12. The complete picture combines vehicle theft and information exchange with checking vehicles by PO, re-registration activities in VRO and border crossing. Besides work processes performed by various officials also additional activities are added into the activity diagram, like stealing of the vehicle, choosing further action by the thief and actions related to the vehicle (driving, re-registration, border crossing, or dismantling for spare parts).

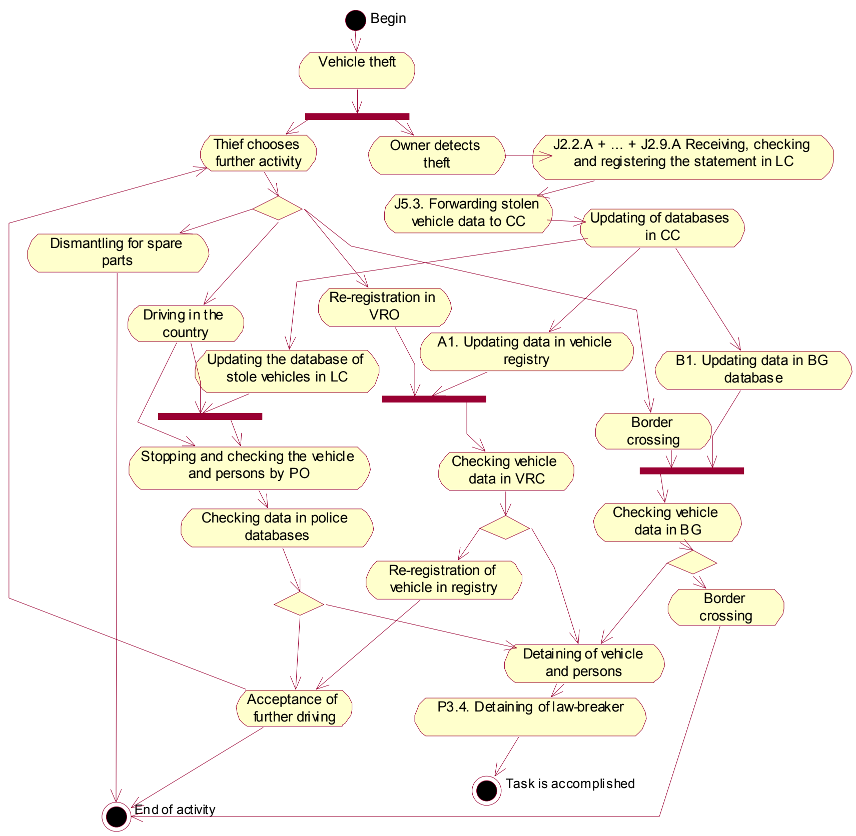

Figure 5.12 The overall activity diagram of the case

The given activity diagram illustrates the concurrency of activities related to vehicle theft and information exchange. It represents the key point of the analysis – as a consequence, a vehicle theft may lead to different results, depending on whether the information is exchanged in time (the stolen vehicle is detained) or is delayed (the stolen vehicle may drive further). The activity diagram illustrates the procedures in information exchange and possible different consequences.

In real life the situations are not so "clear" and may have different variations, like:

1. Documents are may-be correct and PO has already a suspicion even if the database does not show any violations.
2. Vehicle is not checked in the database because it does not seem to be necessary.
3. Owner finds the vehicle or buys it back from the thief but does not inform the police. The vehicle remains wanted and causes additional problems when the owner crosses the border or changes the ownership legally in the VRO.

Those variations do not change the processes substantially and therefore are not considered in the current example.

Temporal analysis in the Q-model is crucial for the resultant analysis of the case. As seen from use case and activity diagrams, resultant information exchange must be quicker than the activities of the thief. Such temporal behaviour requires temporal analysis with taking into account of the real temporal characteristics and is not possible to analyse by using only UML activity or sequence diagrams.

The process model in the Q-model notation of the vehicle theft case results from the conversion of the use cases and activity diagrams given in UML notation. Both human work processes as well as software processes are included in the Q-model. In order to get useful results, only main activities are considered. For better readability, the resulting model is slightly simplified, for example some strictly sequential processes are combined into one model. The processes that are implemented in Q-model notification can be illustrated in a table as presented in Table 5.3, Table 5.4 and Table 5.5. As introduced in Section 4.3, the Q-model process table is similar to the process table presented in Table 5.2. The columns in the current table express the properties of the Q-model processes (as described in Section 4.3). All values of timing parameters are expressed in minutes. The rows are grouped according to the actors.

The Q-model analysis is used here for detailed modelling of inter-process data exchange and timing parameters that characterise the overall behaviour of the model. This aspect is important since the queues, concurrency of work processes, shared resources and bottlenecks play a significant role in this example.

The Q-model diagram results from the model in the Q-model notation. The analysis of the timing diagram demonstrates that under given circumstances the vehicle may cross the border before the border guard will have updated their stolen vehicle database.

Table 5.3 Processes of LC, CC and PO in the Q-model

Process ID	Process name in the Q-model	Input data consumption		Execution time		In case of output selector process		Equi-valen-ce inter-val	Remarks
		Ear-liest	Latest	Shor-test	Lon-gest	Port combi-nation	Combi nation probab ility		
1	2	5	6	7	8	9	10	11	12
I	Beginning of the process								
	Vehicle_theft	-	-	0,5	3			0,01	Period length 100, divergence 10
	Owner_notices_theft	0	0	0,5	60			0,01	
	Informing	1	2	5	10			0,01	
II	Activities of LC and CC								
J2.2.A + J2.6	J22A_J26_Mess_receiv	0	0	27	31			5	
J2.9.A	J29A_Event_registr	0	0	2	3			2	
J5.3.A	J53e_Infor-mation_pass_to_CC	0	0	0,2	0,5			0,01	*incl e-mail*
J6.1.A + J6.2 + J6.3	J61A_J62_Receiving_message_CC	0	0	0,45	0,6			0,2	
J6.9	J6_9_Check_VRC	0	0	0,25	0,5			0,01	
J6.10	J6_10_Input_into_AMT	0	0	2	3,5			0,01	
J6.11	J6_11_Message_(fax)_passing	0	0	0,5	1			0,01	
	AMT_db_update	0	0	0	0			450	Period length 480, divergence 0
IV	PO activities								
P2.20, K4.11	P220_K411_Stopping_the_vehicle	0	0	0,05	0,2			0,2	
P2.19, K4.10	P219_K410_Check_vehicle	0	0	0,15	0,30			0,3	
J3.1 + J3.4	J31_J34_V_d_query	0	0	0,25	0,75			0,01	
P2.21, K4.11	P2_21__K4_11_Vehicle_check_decision	0 / 1,6	0 / 1,6	1,7	2,5	Port1 Port4	0,5 0,5	1,5	

Table 5.4 Processes of VR and BG in the Q-model

Process ID	Process name in the Q-model	Input data consumption: Earliest	Input data consumption: Latest	Execution time: Shortest	Execution time: Longest	In case of output selector process: Port combination	In case of output selector process: Combination probability	Equivalence interval	Remarks
1	2	5	6	7	8	9	10	11	12
A1	A1_VRC_ bd_update	0	0	1	1			0,01	
	ARK_db_ result	0	0	0,08	0,08	Port3, Port4, Port5, Port6	1	0,075	
A2	A2_Begin_ reregistering	1	1	2	2			0,01	
A4	A4_VRC_ decision	0 / 2,2	0 / 2,2	6	8	Port1 Port4	0,5 0,5	15	
B1	B1_BG_db_ update	1	2	2,5	3			30	
	BG_db_result	0 / 0,01	0 / 0,01	0,08	0,08	Port1, Port3	1	0,05	
B2 (part 1)	B2_Begin_ vehicle_check	0	0	1	1			0,01	
B2 (part 2)	B2_BG_ decision	0 / 1	0 / 1	2,5	3,5	Port1 Port4	0,5 0,5	4	
B4	B4_Border_ crossing	0	0	2	5			4	

Table 5.5 Other processes of the case, described in the Q-model

Process ID	Process name in the Q-model	Input data consumption: Earliest	Input data consumption: Latest	Execution time: Shortest	Execution time: Longest	In case of output selector process: Port combination	In case of output selector process: Combination probability	Equivalence interval	Remarks
1	2	5	6	7	8	9	10	11	12
	Pol_db_result	0 / 0,01	0 / 0,01	0,05	0,08	Port5	1	0,049	
J3.6	J36_answer	0 / 0,1	0 / 0,15	0,5	1,5			1,1	
A5, B5	A5_B5_ Detaining	0	0	6	15			0,01	
	Driving_in_ country	0	0	10	10	Port1 Port4 Port5 Port2 Port6	0,75 0,05 0,05 0,05 0,1	8	*drive border rereg. spare check*
	For_spare_ parts	0	4	5	10			10	

5.3 ACTORS AND ROLES

In the present section key actors are described and their activities, roles and behaviour emphasized. The five main officials considered here are duty officer in LC, duty officer in CC, patrolling officer (PO), an officer in vehicle registration centre (VRO) and a border guard officer (BGO) in the border point. Other related human actors are the owner of the vehicle and the thief. Class diagrams are presented for police officials only.

The overall behaviour of the system emerges form the actors' activities. The actual result (whether the police will find the vehicle or not) depends on the actions of all related actors and therefore can not be predicted. At the same time different enablers can still be analysed and illustrated that lead to one other or to another result, and some grounds for choosing the actual behaviour can be presented.

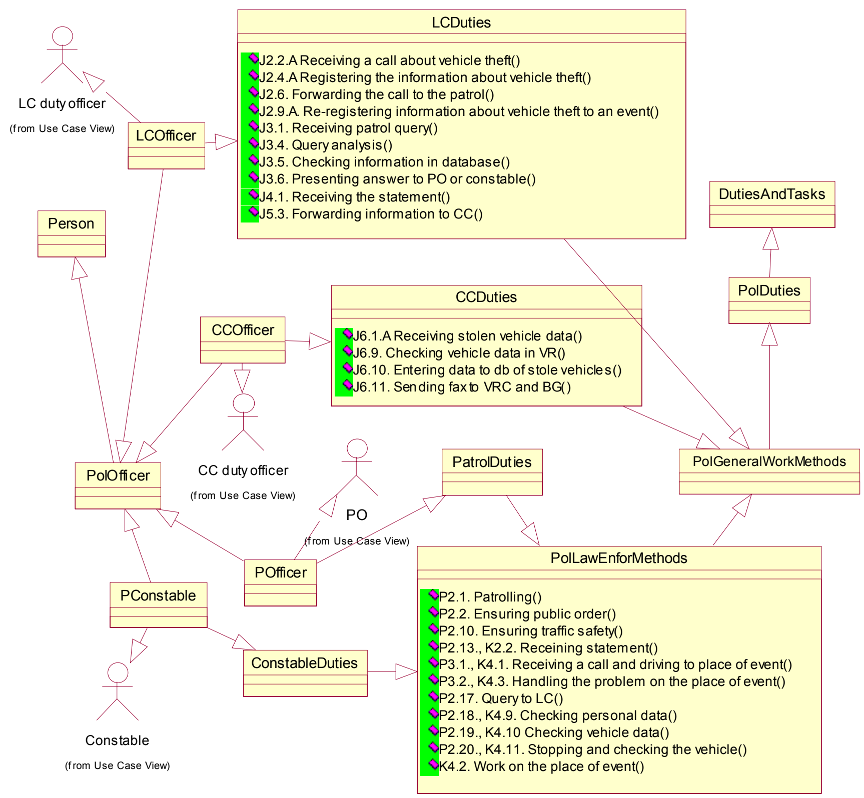

Figure 5.13 Description of actors of the system and their roles as classes

The **class model** for police officials in the current example is the following (illustrated in Figure 5.13). A specific class (named the actor class) is designed for each actor LC, CC and PO. Those classes are LCOfficer, CCOfficer and POfficer, respectively. Each of these classes inherits its properties from two upper classes:

1. The characteristics of a police officer for all actor classes are inherited form the class PolOfficer.
2. The descriptions of work tasks for each actor class are inherited from the specific class that describes the duties of different work positions respectively (classes LCDuties, CCDuties and PatrolDuties).

Since the UML does not allow dual inheritance, the classes that describe duties are seen as interfaces. In the specified example patrolling officer has the same work methods as constable; therefore these are inherited from the class PolLawEnforMethods, which is their super class.

For each police official (duty officer or patrolling officer) the methods that he or she uses are a combination of the two parent classes with a great amount of individuality.

The aim of the class diagrams is to specify agents' methods and attributes. Agent class diagrams help to formalise the agent's activities and priorities.

The **agent simulation** of the case study is the following. The organisation, in the context of vehicle theft case, is now modelled as a multi-agent system where twelve agents are currently representing the most important human or software actors. JADE environment is used for modelling. Communication of the agents corresponds to the FIPA standards. The agent class names, actor names and the description (roles) are presented in Table 5.6.

Table 5.6 Agents and their roles in the modelled system

Class name	Agent name	Role
Vehicle	vehicle0	the vehicle
Owner	owner0	the owner of the vehicle, also the announcer of the theft
Thief	thief0	the thief, performs the theft
LCAgent	lc0	duty officer in LC
CCAgent	cc0	duty officer in CC
POfficer	patr0	patrolling officer
AMTBaas	amtb0	database of stolen vehicles AMT
ARKBaas	arkb0	vehicle register
PVABaas	pvab0	information system of the border guard
ArkAm	arkam0	official of a bureau of the vehicle registration centre
PvaAm	pvam0	border guard officer in the border crossing point
Logger	logger0	an agent that is responsible for creating log files on the messages exchanged between the agents

The initial task of the simulation is to express emergent behaviour that results from multiples tasks. The roles of the officials are initially seen only form the official viewpoint and for the simplification of the model the personal habits etc are omitted.

The agent class models in agent simulation are platform-specific, i.e. platform-independent class diagrams are modified for being able to be executed in the agent development environment JADE.

The multi-agent system functions as follows. The agent owner0 checks periodically whether the vehicle is present. For this it sends a message "query-if" to the agent vehicle0 and waits for the answer "agree". If the car is stolen it sends no message or it sends the message "failure" to the agent owner0. If the agent owner0 receives no proper message from agent vehicle0, it sends a message "inform" about the vehicle theft to the agent lc0.

The agent thief0 sends at a random time a message "inform" to the agent vehicle0 to inform that it is stolen. After this the agent thief0 waits for some amount of time and then tries to do one of the following:

1. Re-register the vehicle in the vehicle register, thus sending a message "request" to the agent vro0;
2. Cross the border with the vehicle (perhaps after re-registration), thus sending a message "request" to the agent bgo0;
3. Dismantle the car by sending a message "inform" to the agent vehicle0.

The agent patr0 checks periodically the agent vehicle0 by sending a message "request" to it. The received data is then sent to agent lc0 for performing checks. If the vehicle is not marked stolen, then the agent patr0 sends a message to the agent vehicle0 so permitting to continue driving. If the vehicle is marked stolen, then the agent patr0 orders the agent vehicle0 to stop.

Other agents are mainly implemented as ping-and-wait-behaviour-agents that wait for specific messages and then perform actions and send answers depending on the message received. A logging agent is implemented to log all messages that are exchanged between agents as well as between agents' internal states into a log file. The monitoring of communication can also be observed by using a sniffer agent in a graphical user interface of the JADE environment.

Figure 5.14 describes class diagram for agent LC (class LCAgent). The figure represents that the agent behaviour is a composition of four behaviours:

1. WaitPingAndReplyBehaviour – an agent waits for a message, if it understands the message, it chooses an action (implemented as another behaviour).
2. VTMBehaviour – activities related to vehicle theft (including checking the information, forwarding information to CC).
3. RPQBehaviour – activities related to receiving information request from police patrol, checking it in databases and responding to the answer.

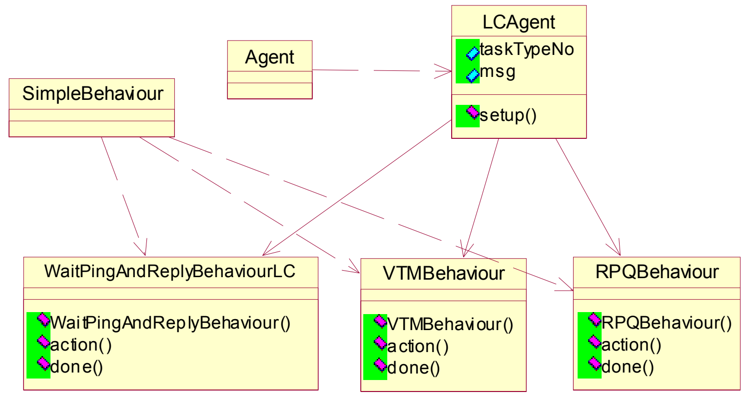

Figure 5.14 A fragment of the LCAgent class diagram

Since hard real time is not important in this case, the whole simulation is performed on a single processor (an ordinary office computer). In the current example the process of making personal choices (e.g. duty officer responding to multiple concurrent tasks) is not implemented as cognitive agent, each agent performs in a rather trivial way according to its perception ability (e.g. as reactive agent).

The modelled systems allow to look dynamically at the behaviour of the processes. The elaborated system displays all interactions between agents into the command window and also writes them all into the log file where all the inter-agent communication is recorded. The content of the log file can later be used for analysis.

In the current modelling the used time criteria are realistic. The duration in minutes is replaced in the model by the same numeric value in seconds, so the activities are performed 60 times quicker. Timing properties of the agents were taken from the *general description of the organisation* and *process table*. Since no hard real time was important to the model, the whole simulation was performed on a single processor on an ordinary office computer.

A part of an agent code is presented as an example. Here is an agent class LC that extends the existing class Agent. The code is commented in the text of the code itself.

```
public class LC extends Agent {

    protected void setup() {
    // instantiation and registration of the agent
    ...
    DFService.register(this,dfd);
    logFile.log("Agent: " + getLocalName() + " born");

    // description of behaviours
```

```
        // behaviour 1 - wait message and reply
        WaitPingAndReplyBehaviour pingBehaviour = new
      WaitPingAndReplyBehaviour(this);
      addBehaviour(pingBehaviour);
      // käitumine 2: message about vehicle theft
      // vehicle-theft-message-behaviour
            VTMBehaviour vtmBehaviour = new
VTMBehaviour(this);
      addBehaviour(vtmBehaviour);
      // käitumine 3: query form patrol
      // response-to-patrol-query-behaviour
            RPQBehaviour rpqBehaviour = new
RPQBehaviour(this);
      addBehaviour(rpqBehaviour);
      } // end setup


      class WaitPingAndReplyBehaviour extends
SimpleBehaviour {
      // description of behaviour 1
      private boolean finished = false;
      public WaitPingAndReplyBehaviour(Agent a) {
          super(a);
      }
      public void action() {
          // real actions
          // 1. Waiting for message
          msg = blockingReceive();
          taskTypeNo=1;

          if(msg != null){
              senderName = msg.getSender().getLocalName();
              ACLMessage reply = msg.createReply();
              if(msg.getPerformative()==
ACLMessage.INFORM){
                String content = msg.getContent();
                if ((content != null) &&
((content.startsWith("Vehicle theft")== true) ||
(content.startsWith("Statement on vehicle theft")== true))
){
                    // 1. a) vehicle theft message from
owner (J2.2A)

reply.setPerformative(ACLMessage.CONFIRM);
                    reply.setContent("Vehicle theft message
received");
                    taskTypeNo =2;
                      }
                else {
                    // 1. c) another message
```

```
reply.setPerformative(ACLMessage.CONFIRM);
                    reply.setContent(" Message received");
                    taskTypeNo =1;
                }
              }
              else {
                . . .
              } //
            if (taskTypeNo != 3) {
                . . .
            }
          } else {
              // empty msg
              //System.out.println("No message received");
          }
      } // end action

      public boolean done() {
          return finished;
      }
      } //End class WaitPingAndReplyBehaviour
  . . . // other classes
```

Instantiation of the class and its execution is done in the JADE environment. For observing the communication of agent graphically and getting the interaction diagram a sniffer agent should be executed form graphical user interface.

5.4 MODEL ANALYSIS

5.4.1 The completed organisation model and its analysis

According to the description presented in Sections 5.1 – 5.3, a combined model *M* of the organisation is elaborated. The current model consists of the following components:

1. Description of the organisation (given in Section5.1);
2. Description of organisation work processes and UML diagrams (given in Section 5.2);
3. Description of timing parameters and interactions between processes in the Q-model notation (given in the same section);
4. Description of actors and their roles (class models and agent simulation, given in Section 5.3).

Checking of the correctness of the model and its correspondence to reality was performed on the basis of descriptions, use case diagrams and activity diagrams by using the expertise of the actors of the described organisation (the same police authorities that were used for devising the model). Since real work processes are different in many prefectures and three prefectures were considered during the initial modelling, possible gaps in the model were looked for. Those were not found, therefore the model is considered to be exact enough at the demanded level and allows the modelling and analysis of problems described in the analysis task.

A number of simulations were made on different components of the *organisation model*, mainly on the Q-model in the Limits environment and multi-agent model (agent simulation), e.g. evaluation of timing parameters of the existing work procedures (some interesting results were achieved, which are described in the next section), modification of work processes to speed up the information exchange and changing some of the information processing procedures in the police information exchange. In addition to simulations, also activity diagrams and use case diagrams were analysed and different expert opinions from the field were considered. An example of the piece of multi-agent simulation log file is presented hereunder.

```
[15.03.03 12:08:22] thief1:->vehicle1:INFORM. The vehicle
  is now stolen.
[15.03.03 12:24:30] owner1:->vehicle1:QUERY-IF. Checking
  the vehicle.
[15.03.03 12:24:32] vehicle1:->owner1:FAILURE. The vehicle
  is stolen.
[15.03.03 12:31:48] owner1:->lc1:INFORM. Vehicle theft:
  123ABC, owner: First Owner.
[15.03.03 12:41:11] lc1:->cc1:INFORM. Vehicle theft:
  123ABC, owner: First Owner.
[15.03.03 12:51:51] cc1:->poldb1:INFORM. Vehicle theft:
  123ABC, owner: First Owner.
[15.03.03 12:52:17] cc1:->vrdb1:INFORM. Fax: Vehicle theft:
  123ABC, owner: First Owner.
[15.03.03 12:52:42] cc1:->bgdb1:INFORM. Fax: Vehicle theft:
  123ABC, owner: First Owner.
[15.03.03 15:12:02] thief1:->bgo1:REQUEST. Request for
  border crossing: vehicle 123ABC, owner: First Owner.
[15.03.03 15:12:14] bgo1:->bgdb1:REQUEST. Check data:
  123ABC, owner: First Owner.
[15.03.03 15:12:17] bgdb1:->bgo1:INFORM. The vehicle IS
  STOLEN.
[15.03.03 15:12:20] bgo1:->thief1:REQUEST. You and the
  vehicle are detained.
```

This communication can also be represented graphically in an interaction diagram.

Different solutions for the modification of information exchange procedures, items of information were exchanged and queries were made. The experiments were fixed for later analysis. Successful modifications and proposals were collected into different sub-models and packets and they virtually form the *change model* (*C*).

The analysis of timing parameters (performed on the initial Q-model diagram) shows that the whole process from car theft until updating the databases may be divided into four stages. First, time interval from car theft until noticing the event by the owner is in most cases approximately 20 minutes. In some cases the owner notices the fact of theft in the next morning or even several days later. Clearly so late announcement cannot be eliminated by quick information processing, therefore such cases are not considered separately. Second, the time interval between the police has received the information and checks it at the place of event until the confirmation is sent to LC is in reality approximately 30 minutes. Prior to analysis the asked persons had estimated it to be approximately 10 minutes. Additional time elapses since the police usually want to check the event on the spot and have the owner's signature on the statement. Third, the time needed to change data in the police and in all necessary databases is 10 minutes on the average (including sending faxes and entering the information). At last, updating local databases is done 3 times per day. This is not critical since information about the latest vehicle thefts reaches police prefectures by fax and the database is just an overall list of all stolen vehicles. Moreover, the checks in prefectures (e.g. LC) are often made in central databases. Therefore time for updating local databases is really not so critical and offers just additional information about the thefts.

The analysis of actors has shown that the main objective difficulty in performance is the overall workload of duty officers in a local police prefecture. The actual duplication of processes is that CC checks the information again in the vehicle register and only then sends a fax and enters data into the database.

The previously presented time values are based on multiple simulation series and are therefore generalized. The initial modelling in the Limits environment on the Q-model has shown that if the thief has prepared falsified documents for the car, in most cases they success in re-registering the vehicle in the VRO or crossing the border. The simulations in the agent model displayed in principle the same results. For more realistic experiments the probability value for driving inside the country before re-registering or border crossing was increased, comparing with the initial model. As the outcome shows, the test results were more optimistic – the vehicle will be detained eventually during driving (if stopped by a police patrol), when performing re-registration or at the border. Results of different agent simulations were stored in the log file and analysed later.

According to observations in the police prefectures all related actors perform their activities correctly and personal sympathies and antipathies do not influence the work processes. Therefore alternative behaviour is not necessary to model. For purely experimental purposes some possible alternative behaviour of LC was modelled. This is described as an extension to the current example in Section 5.5 of the thesis.

The analyses in the Limits environment on the Q-model and simulations on the agent model in the JADE environment have shown that correctness and speed of information exchange plays an important role in detaining stolen vehicles and unnecessary delays may cause unwanted results. Some reasons for possible delays were detected during the analysis (e.g. overall workload of LC; also the LC must be sure that this information is correct before transmitting it to CC; the CC re-checks the information again in the vehicle register for being able to be responsible for the correctness of the information). The timing analyses lead to proposals to speed up information processing steps, especially initial checking of the correctness of the information and to eliminate duplicate processes.

Four alternative solutions were proposed on how to modify information systems; the most promising solution was to concentrate on the quick development of the information system POLIS, to build up one central database and connect it to the vehicle register and BG information system. In this case data are entered only once at any level and information is quickly accessible to all related authorities. Automatic checks should also be introduced to all necessary registers as well as a preliminary warning system to allow the entering of initial information into all databases before the final confirmation about the theft is gathered. To use one central database with an informing feature about received important information that is added into the database should economize on 15 minutes for each case. For simplified querying the use of mobile data terminals by police patrol cars should be considered.

5.4.2 The modified organisation model

The modified model is made as a result of experiment. The given model expresses how the system shall behaviour after the implementation of modification suggestions into actual work processes. To fit into the space, only modified parts of the initial model are presented hereunder. The given model corresponds to the final suggestions.

The information system of the police was modified to avoid duplication and reduce the execution time of the work processes. It was suggested that the system should be re-structured according to use-case models represented in Figure 5.15. As observed, CC, BG officer and VRO officer are no longer a part of the essential information exchange and CC is merely a unit for information exchange to other actors in the Police Board. According to the modified model, the vehicle registration office and the border guard may have direct queries to the police database. The modified use-case model will also lead to a simpler activity diagram in Figure 5.16 as compared with the one in Figure 5.10.

The actual work processes of checking and querying are the same as previously, but in the future the use of mobile data terminals for police patrols should simplify and speed up the checking procedure (LC as bottleneck can be omitted for simple queries).

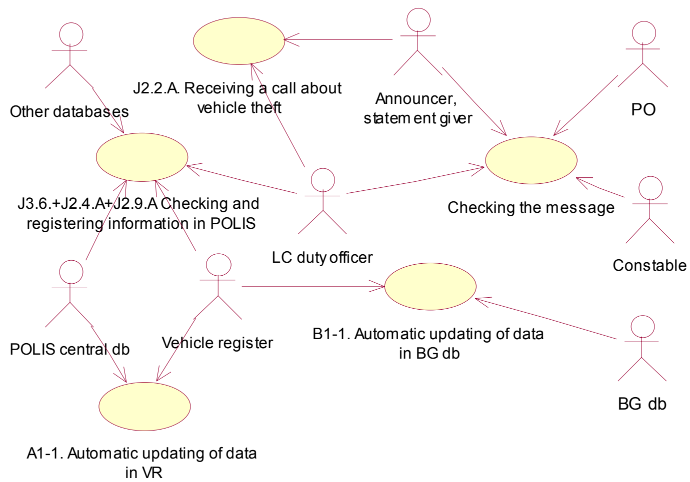

Figure 5.15 A use case diagram of modified event registration and information exchange

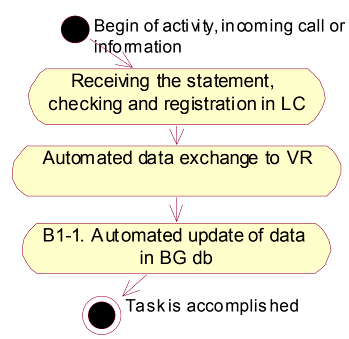

Figure 5.16 An activity model of modified processes

The activity diagram, describing the overall process is presented in Figure 5.17 (in comparison to the previous one, given in Figure 5.12).

The modified and newly introduced processes in the Q-model notation are described in Table 5.7. In the modified model processes J5.3.A, J6.1.A, J6.2, J6.3, J6.9, J6.10 and J6.11 are eliminated. Other processes remain the same. Also, the processes of checking the vehicle as well as querying remain the same until the mobile data terminals are introduced. The mobile terminals for patrols should simplify and speed up the checking procedure.

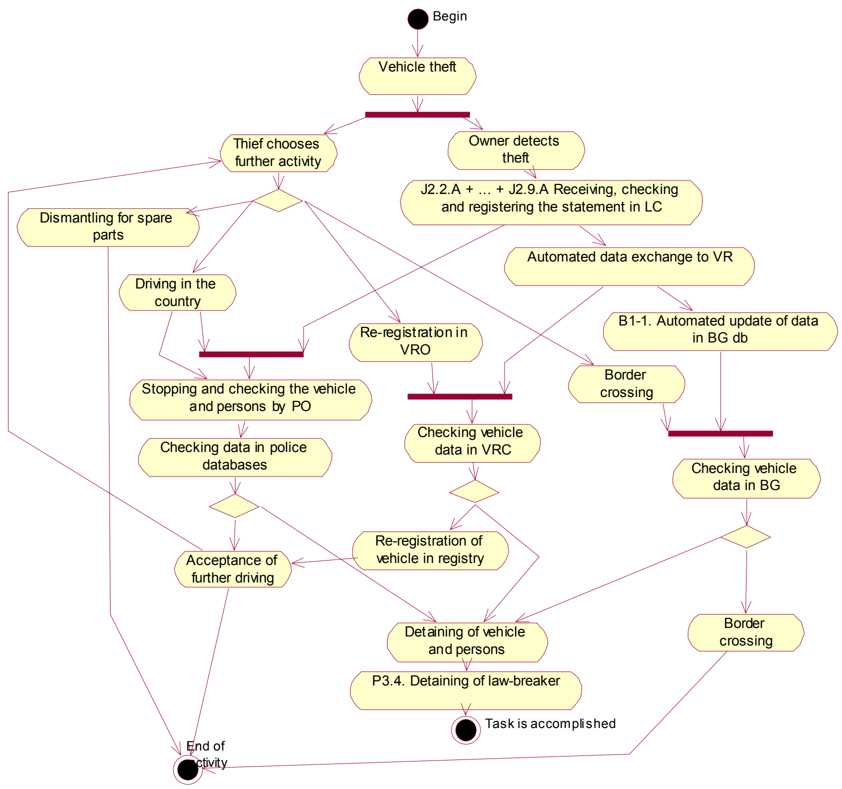

Figure 5.17 The overall activity diagram with modified processes

Table 5.7 New and modified processes in the Q-model

Pro-cess ID	Process name in the Q-model	Input data consumption		Execution time		In case of output selector process		Equi-valence interval	Re-mar ks
		Earliest	Latest	Shor-test	Lon-gest	Port combi-nation	Combi nation prob-ability		
1	2	5	6	7	8	9	10	11	12
II	LC and CC activities								
	Record_start	0	0	0,01	0,01			0,01	
	Record_end	0 / 0,1	0 / 0,1	0,02	0,02			0,01	
IV	PO activities								
	V_d_mobile_ query	0	0	0,1	0,3			0,01	

Since the behaviour of the agent model is based on the messages exchanged between agents, it is necessary to change the messages that are passed between agents so that the agents will behaviour rationally when receiving new kind of messages. Therefore some of the methods for agent LC, ArkBaas and PvaBaas will be modified. Also the new important agent PolisBaas was introduced (in the initial modelling this agent was not modelled since it does not play a role in the current example system except for just making a delay and this was implemented in the agent LC). The agent AmtBaas was not used anymore.

5.4.3 Summary of the case study

It was proposed that the implementation of given proposals should cause only one-time modification and there should be no need to increase manpower in everyday performance. As opposite, recommendation suggested standardising and automating information exchange procedures as much as possible.

The modified model is possible to present only by different sub-models (use case and activity diagrams, the Q-model diagrams, agent models) as is the initial one.

Table 5.8 Review of the analysis task

Prio rity	Achieved result	Evaluation of results
1	Timing characteristics for information exchange are known for each stage, information exchange is currently not optimal and it is possible to speed it up	Information exchange stages are described correctly
2	Duplicated processes and reasons for this are known	It seems that processes are described at a sufficient level. Feedback is missing
3	Information exchange can be sped up by process modifications	Review is exact; it includes the list of activities and temporal analysis
4	Police patrol can only make queries using radio or mobile telephone. The actual search in the database is made by a duty officer	Evaluation and conclusions are correct, it may be necessary to perform an additional analysis, related to specific problems
5	Bottlenecks are listed and suggestions for improvement (related to analysis task priorities 1-3) are given	Description, a list of bottlenecks, analysis and suggestions are described at the sufficient level of exactness and detailness. External feedback about the usefulness of the suggestions is missing

The initial model corresponding to the initial state of the organisation, the final model corresponding to the modified organisation and its components (use case and activity diagrams, agent model and other) were used for devising a not very detailed implementation plan (change management plan).

The evaluation of the fulfilment of the requirements of the *analysis task* (in Section5.1) is presented in Table 5.8. Priority numbers correspond to the initial table (Table 5.1).

The goals that were set up in the initial analysis and modelling task were partially fulfilled: the modelling clarified the existing stages in information exchange and the efficiency of information exchange methods. Some improvement possibilities were found and suggestions made. The proposals were reached by the analysis of the use case diagrams, Q-model representation of the case and its agent simulation. It was most important to describe planned work processes and especially their timing criteria. Considering the application characteristics, multi-agent simulation gave much visual illustration to actual processes and actors' behaviour. Since personal motivation and preferences were not problematic in this case study, no corresponding improvement suggestions were made. The case study enabled to illustrate all stages in the methodology.

The documentation in the case study was introduced to the organisation and generally positive feedback was received. The document was actually used for the implementation of the given suggestions.

The simulation of the duty officer (LC) with its other tasks actually showed some additional problems (e.g. a duty officer with heavy workload needs time for rest, multiple concurrent tasks may disturb the reception of information) and gave valuable information for the modification of the existing processes. This expressed the need to model not only directly related processes but also all the activities of participants that may interfere with the currently modelled processes. The recommendations given in the document were implemented in the police organisation. Information systems modifications and new work processes were implemented in March 2004 almost fully in accordance with the suggestions. Detected problems were solved but the organisation did not update the existing model, so a possible analysis should again update the model. It is planned to use the methodology in the future also for other analyses but currently it is not related to everyday decision-making in the organisation.

Excellent characteristics of an emergent organisation and an existing need to model complex tasks in the particular organisation are the main reasons for choosing the police as a modelling example. The handling of similar issues in other researches (e.g. Dugdale, Pavard, Soubie, 2000) gives a good opportunity for comparing the results.

In this case study the behaviour of the system was so far modelled without personal interests and depends directly on the messages received: e.g. each input message received by an agent causes specified response (a message or activity). The behaviour of the system under such circumstances is still emerging since the thief agent makes choices on a random basis and the period of checking the vehicle by the owner and the patrolling officer is made randomly within pre-determined time limits. Therefore the overall behaviour of the system is not determined and emerges actually from the activities and messages of all participating actors and varies in each execution of the Q-model and/or agent simulation.

5.5 MODELLING OF MOTIVATIONS AND CO-OPERATION

The aim of the additional modelling in this case study was to proceed from the initial task, given by the organisation (analyse only information exchange processes in the vehicle theft), and for research purposes to model the additional level of emergent behaviour that emerges from multiple simultaneous tasks with non-specified priorities of key actors (in this case LC).

The initial model is taken as the basis for evaluating the characteristics of motivation and co-operation. The overall activity diagram of the case study in the UML was given in Figure 5.12 (Section 5.2) and agent model described in Section 5.3.

As described, an organisational hierarchy, pre-described work processes and a hierarchy of messages exist in the system. The co-operation between the dispatcher and the patrolling agents, duties of the dispatcher agent and the performance of database agents is determined. In addition, agents might have their own priorities and their behaviour is a mix of different values.

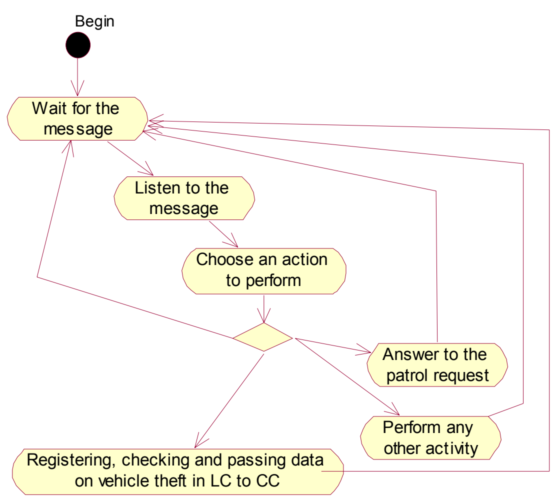

Figure 5.18 A detailed activity diagram of the dispatcher agent

In the given initial version each agent performs in a rather trivial way according to its perception ability (e.g. as reactive agent with reflective behaviour) and the process of making personal choices (e.g. dispatcher officer responding to multiple concurrent tasks) is implemented as a reactive agent. A detailed activity diagram of the dispatcher agent is given in Figure 5.18.

In reality, each of the agents has its own interests and goals. To model co-operation and personal interests and to study emergent behaviour of an actor, the behaviour of the dispatcher agent (LC) is modified to add personal interests and limits of LC. The dispatcher agent has additional behavioural modules, as illustrated in activity diagram in Figure 5.19. When a message comes, the agent decides to receive or not to receive the message (depending on its current activeness). Later it decides according to its own priorities which task to choose and how quickly to proceed with the task. Activeness and message listening interest indicates the general working interest and depends on previous activities (after performing some activities the agent needs some time for resting).

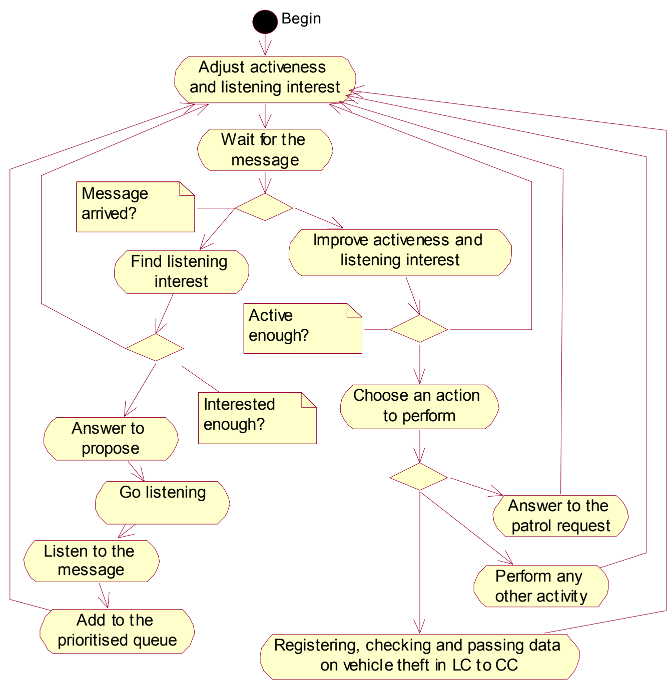

Figure 5.19 The activity diagram of modified behaviour of the dispatcher agent LC

The corresponding class diagram, illustrating the class diagram for modified actor LC in the computerised MAS simulation, is given in Figure 5.20 (as compared to Figure 5.14).

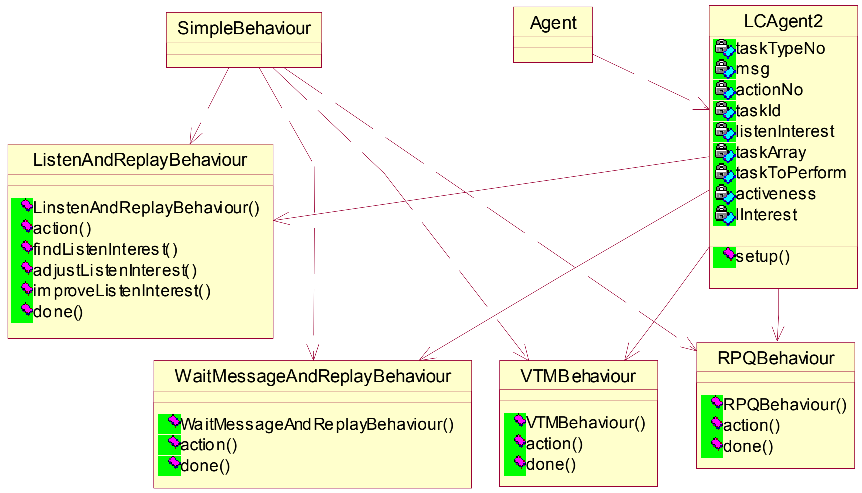

Figure 5.20 A fragment of a modified class diagram of LC agent

As seen from the figure, there is new behaviour introduced: ListenAndReplayBehaviour. According to this behaviour, an agent calculates its interest for listening to a new task or any request form other agents and if it is interested enough, waits for a message for a certain time period (WaitMessageAndReplayBehaviour) or performs any activity that was postponed previously. The WaitMessageAndReplayBehaviour is a modification of WaitPingAndReplayBehaviour, illustrated in Figure 5.20. Other behaviours are similar to the previous ones, given in Figure 5.14.

As a result, when adding personal interests to the behaviour of the dispatcher agent (mainly control over work load) in the current case study, the system is substantially more complex. Three reasons for this can be pointed out:

1. The communication with the dispatcher agent is more difficult: the agent may not wish to listen to others, other agents have to start the communication with a "propose".
2. The behaviour of the agent is more complex -- the message is received if the agent is active enough, it is placed in the queue and will be performed depending on the agent's activeness and priorities.
3. The behaviour of the agent cannot be directly influenced or determined by one single message but is determined by (all) earlier received messages, activeness, messages queue and also by the priorities of the messages that arrive later. This can be seen as a stream input – stream output over time where the total input stream influences the total output stream of the messages during the activity of the agent.

The tasks were solved more slowly in the modified version. Also the owner and the patrolling agents had to sometimes send their requests multiple times.

As seen, the model of the behaviour of an organisation can be improved when also subjective attitudes and motivations of employees are considered. Motivation, attitude and readiness for co-operation can be added to the model. In this case the emergent behaviour is more realistic: work processes are performed more slowly since additional time-consuming operations are considered and different disturbing factors will come out.

5.6 CASE STUDY ON TASK DECOMPOSITION AND CO-OPERATION OF STRUCTURAL UNITS

This case study illustrates the modelling of task decomposition and co-operation between structural units within the same organisation. The aim of the case study is to evaluate if and how the methodology is suitable for modelling (non-operational) co-operation between groups of actors. The chosen organisation (the Police Board) is hierarchical and multi-functional. The modelling task is to analyse what are the necessary activities related to the development and maintenance of police communication systems, to suggest how these tasks can be allocated between two structural units – development department and maintenance department.

Table 5.9 Initial task allocation between different units

Task No	Sub-tasks, activities and processes in the domain	Competence of DD	Competence of MD	Competence of PP
1	Development of communication systems, use of transmission channels and radio frequencies	+		
2	Maintenance of communication systems, networks and equipment		+	
3	Analyses and development of the quality of service and cost efficiency in communication systems	+		
4	Accounting of equipment and users		+	+
5	Organizing the use of equipment and systems (quality and quantity requirement analysis)	+		
6	Maintenance of telephone systems	+	+	
7	Elaboration of training programs	+		
8	Organizing the training of users		+	+
9	Co-operation with other authorities	+		

The domain of the development and maintenance of police communication systems has been in the same unit for several years. Recent managerial decision is that development department (DD) will be responsible for different development projects and activities in the area of information and communication technology. Maintenance of police communication systems will be one of the tasks of maintenance department (MD).

In order to simplify task allocation, a list of domain sub-tasks, activities and processes was prepared by the experts of this domain. This list (containing ca 50 different activities) foresees proposed task allocation between DD, MD and local police authorities (police prefectures, PP). A part of the initial task allocation list is given for illustration in Table 5.9.

The analysis task in the current case study is to analyse whether the initially proposed task allocation is compliant with actual needs and whether alternative allocation of some sub-tasks can be suggested. According to the introduced methodology, a simplified analysis task is given in the form of the prioritised list in Table 5.10.

Table 5.10 The analysis task

Priority No	Problem ID	Short description of the problem	Current situation	Desired result	Criteria for evaluating the result
1	Analyse initial list	Analyse the initially proposed list of activities and task allocation, check the logic, duplication of tasks and possible confusions in the list	The list may be overlapping, incomplete or contradictory	Contradictions are detected, if they exist	The rest of the list enables clear task allocation
2	Elaborate detailed analysis	Choose the most problematic components, elaborate detailed analysis for chosen components and activities according to the methodology	Problematic components are not known	The analysis is made at a satisfactory quality level	The analysis shows and justifies modification needs
3	Suggest modifications	Evaluate results of analysis and suggest justified modifications in the initially proposed list	–	Quality of the list is improved	Suggestions are accepted by experts

The analysis of the initial task allocation list indicated that there are some confusing statements (e.g. a task described at the general level was assigned to one department but some of its sub-tasks were assigned to different departments, for example, tasks 2 and 6 in Tabel 5.7). Three problematic task groups were chosen for a more detailed analysis:

1. Information exchange related to development needs in radio systems and co-operation in actual development and implementation.
2. Co-operation in the maintenance of telephone systems at the central level.
3. Preparation and performance of user training in new projects.

In the thesis only the first group of tasks is analysed and modelled in more detail.

Some of the most important processes related to co-operation in project development and implementation are presented in the process table, given in Table 5.11 and Table 5.12. The columns “purpose of the activity” and “remarks” are omitted in the current table, otherwise Tabel corresponds to the introduced *processes table W*, given in Section 4.2.3. Process tables were used for determining co-ordination activities and for estimating the need for resources in each unit (e.g. how many maintenance personnel is needed for a certain number of equipment).

The following five actors are distinguished:

- a project co-ordinator for the development of police communication systems in DD (will be noted as ‘PC’ in the model diagrams),
- a project development engineer in DD (‘PD’, actually a number of actors),
- an equipment/solution development engineer in DD (‘PE’, a number of actors),
- a team leader for ensuring the quality of the maintenance of the police communication systems in MD (‘MC’),
- a maintenance engineer in MD (‘ME’, a number of actors).

Each process has its own ID, given in the first column of the table. Some activities (e.g. activity D2) are divided into sub-activities and sub-activities are used in use case and activity diagrams as more appropriate. Activities M1-2 and M2-2, given in Table 5.12, are sub-activities of activity M1 (Maintenance co-ordination) that is skipped in Table 5.12.

For a more detailed analysis of the possible conflicts of responsibilities, the *process table* is described in the use case and activity diagrams. In Figure 5.21 the combined use case diagram is presented that illustrates co-operation on the elaboration and implementation of a new development project in a domain. The actors and the processes correspond to the *process table* (Table 5.11 and Table 5.12) with some differences:

1. Artificial actors (software, communication equipment and vehicle) are added to the diagram.
2. Some processes are omitted for the better clarity of the diagram.
3. A new process, called ParticipationNegotiation, is added to the diagram.

Table 5.11 Work processes, related to development of radio systems

Process ID	Activity name	Input		Action description	Duration[5]	Actor	Output	
		Description	Time criteria[4]				Description	Validity[6]
q	n	μ	t	d	φ	a	ν	τ
1	2	4	5	6	7	8	9	10
C2	Development planning	Ideas, needs, priorities	e	Planning of development and co-ordinating of development projects	h	PC	Project	f
D1	System analysis	Task, info	s	Analysis of general system characteristics	d	PD	Knowledge	c
D2	Elaboration of a new system	Task, info	s, a	Elaboration of a new communication system, participation in working groups	md	PD	Project	p
D2-1	Project idea	Task, info	s, a	Elaboration of project idea	d	PD	Project	p
D2-2	Project elaboration	Task, info	s, a	Detailed elaboration of a new project	md	PD	Project	p
D2-3	Project confirmation	Task, info	s, a	Confirmation of elaborated projects	d	PD	Project	p
D4	Integration plans	Task, info	a	Elaboration of integration plans (radio + data communication, radio + PTN)	d	PD	Project	f or c
D5	Project management	Info	c	Managing development project	d	PD	New system	f

[4] Used values: 'e' – information executes the process, 's' – information is obtained on starting the process, 'c' – continuous, 'a' asynchronous communication, 'b' – batch process

[5] Used values: 'o' – continuous on background, 'h' – varying in hours per week, depending on projects, 'd' – varying in days per month, depending on each task, 'md' – several days over several months

[6] Used values: 'c' – continuous until new information updates are existing, 'f' – final for this task, 'p' – part of the information is completed

Table 5.12 Work processes related to the development of radio systems

Process ID	Activity name	Input		Action description	Duration	Actor	Output	
		Description	Time criteria				Description	Validity
q	n	μ	t	d	φ	a	ν	τ
1	2	4	5	6	7	8	9	10
E1	Software elaboration	Software, plans	b	Preparing new or adjusting the existing programs for equipment	h	PE	Program	c (f)
E2	System modernisation projects	Task, knowledge	e	Proposing and planning modifications in systems	o	PE	Project	f
E3	Upgrade support	Task, project	s	Co-operative support for the implementation of system modifications and updates	h	PE	Work done	f
M1-1	Resource planning	Work to do	s	Completing work schedules	o	MC	Schedule	f
M1-2	Task allocation	Work to do	s	Completing work schedules	o	MC	Schedule	f
T1	Installation	Project, equipment	s	Installation of new equipment	h	ME	Work done	f
T2	Maintenance	Equipment, standards	s	Checking the equipment, repairing	h	ME	Work done	f
T3	Dismantling	Equipment	s	Dismantling of the equipment	h	ME	Work done	f
T4	Program insertion	Software, equipment	s	Insertion of a uniform program into an equipment	h	ME	Work done	f
T5	Project support	Task, project	s	Implementation of system modifications and updates	h	ME	Work done	f

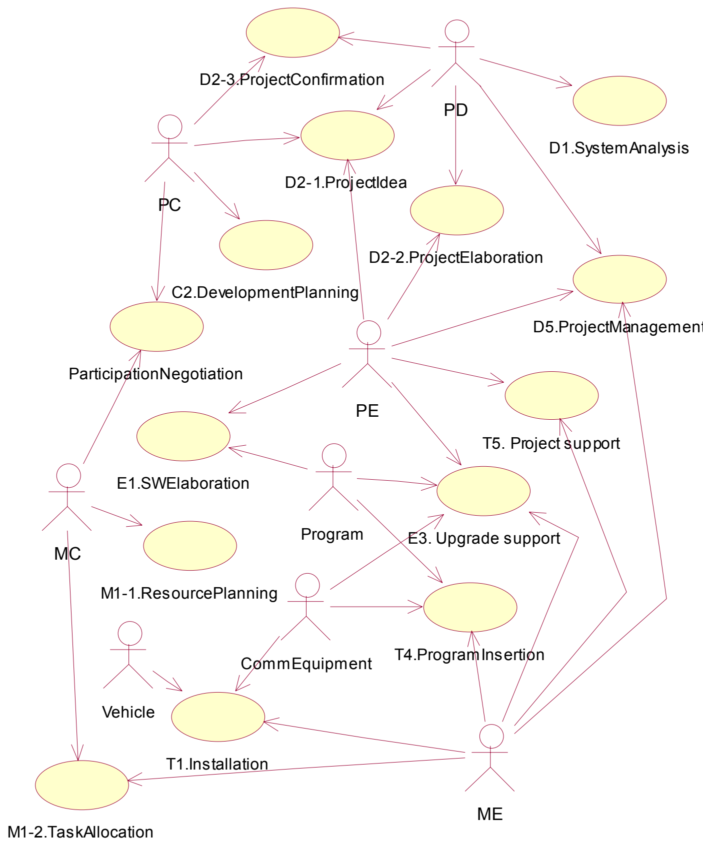

Figure 5.21 The use case diagram related to project development and installation

The process ParticipationNegotiation is a key process for allocation tasks between different units. The *process table* specified necessary processes from the viewpoint of each actor. The use case diagram (and as we will see in the future, also an activity diagram) indicated that there is actually no activity that integrates project development activities in DD and implementation activities in MD. The ParticipationNegotiation comprises the negotiation between the representatives from DD and MD on how (if at all) MD will implement the projects elaborated by DD. During the negotiation DD must consider the capabilities and priorities of MD and modify the project, if necessary.

The most important processes are illustrated in the activity diagram, given in Figure 5.22.

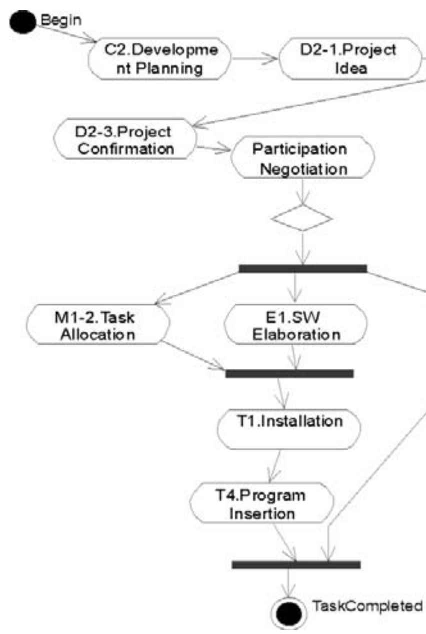

Figure 5.22 The activity diagram related to project development and installation

The current case study enables the composition of the Q-model diagrams for the described system. Still, this activity is not considered appropriate here since the processes do not have specified time criteria. Therefore, for saving modelling time, the Q-model analysis was not performed for task allocation analysis. The Q-model analysis is excellent for analysing possible optimisation of the work routines of maintenance engineers (e.g. installation, maintenance and dismantling of communication equipment). The Q-model diagrams support the estimation and justification of a number of engineers needed in the MD for executing this task on the expected quality level.

The agent class diagram and multi-agent simulation in this case study concentrate on the detailed modelling of co-operation and the possible conflict of interests of actors. The agent class diagram in Figure 5.23 illustrates important activities of the selected five actors.

Similarly to the general principle of the methodology, each actor inherits its properties from two upper classes: personal characteristics from class Person and work-related properties from the corresponding method class (e.g. MEngineer from MEngineerTasks). A brief list of personal characteristics similar to those described in Section 4.4 is given in the class Person for illustration. The work-related attributes are inherited from organisation tasks, goals and normatives that are described in the class OrgTasks. Each structural unit has its own tasks and priorities (MDepTasks and ITDepTasks).

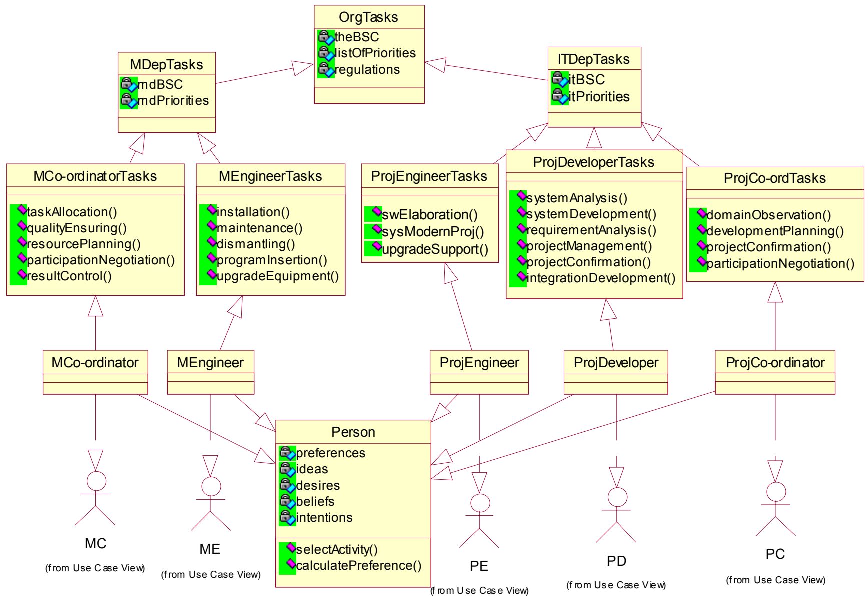

Figure 5.23 The agent class diagram related to project development and installation

The need for co-operation between departments on specific tasks is detected in two actors' groups: between project co-ordinator (PC) and maintenance co-ordinator (MC) and between project development engineer (PD), development engineer (DE) and maintenance engineer (ME). The need for specific co-operation processes (e.g. ParticipationNegotiation, FeedbackAcquiring, UpgradeSupport, etc) emerged already from use case and activity diagrams.

The MAS simulation was prepared by the analysis of use case and activity diagrams. The analysis in the agent model and MAS simulation expressed important issues for estimating expected behaviour. The given task allocation works properly only if both units (DD and MD) are willing to co-operate. The system is very vulnerable to any kind of jeopardizing towards co-operation. Only some alternatives were demonstrated in the MAS simulation for the possibility if one of the units is not co-operative enough. The agent model demonstrated gaps in two alternative cases: first – DD does not execute co-operation and second – MD does not want cooperation.

The motivation in both non-cooperative cases was similar – a non-co-operative unit maximises activities towards satisfying its own priorities and partially decides also on issues that are the competence of another unit (e.g. DD develops projects not taking into account the capability of MD, MD tries to solve minor development issues itself as it considers best).

Simulations concentrated also on how the real co-operation between the departments can be succeeded in given circumstances, in case both units are co-operative and taking their priorities as the basis. The goals and preferences of different actors are analysed and an elaborated version of the balanced scorecard that follows the official priorities is considered.

All different co-operations simulations were done on the same actors (and class diagrams) simply by changing the co-operation attitudes (for this agents were equipped with a similar priority calculation and choosing mechanism as described in the first case study in Section 5.5).

The completed *organisation model* (the *process table*, use case, activity and class diagrams and MAS simulation) was used for multiple analyses. The analysis expressed that in general the initial task allocation was adequate. In some cases alternative decomposition of tasks can be suggested.

The modelling completed the initial analysis task (given in Table 5.10): the initial task allocation list was analysed, components for detailed analysis detected and some detailed analysis elaborated. As a result, suggestions were given to modify the task allocation list. The suggestions were supported by corresponding models and diagrams.

The conclusion was that even the activities and competence between units was allocated according to development / maintenance questions, some task allocation decisions should be reconsidered for giving more freedom to units and reducing the time necessary for everyday co-ordination of activities.

In this regard the analysis fulfilled its aims: to stress the importance of a deeper analysis of task allocation issues by using a supportive tool (in this case this methodology) as well as being supported by the suggestions with corresponding results form simulation and modelling – the simulation enabled a better justification of tasks in the organisation.

The task in the current case study was considerably simpler compared with the first case study (information exchange in case of vehicle theft, described in Sections 5.1 – 5.5 of the thesis). In the current case study the similar acceptable result would possibly be reached also by other methods (brainstorm, pure SWOT, etc), but the current case study indicated the implementability of the current methodology also for the given modelling tasks.

5.7 THE THIRD CASE STUDY: PROJECT TEAM FOR INFORMATION SYSTEMS DEVELOPMENT

In this case study the introduced methodology is used for suggesting necessary roles for a new inter-departmental project organisation for the organisation's information system development. The need for this case study also emerged form actual life – the Police Board had to set up a project team for information system re-engineering. The organisation has so far not applied any project management methodology and resource needs from different structural units were under discussion.

The known conventional project roles and methods were reviewed during analysis. The analysis indicated a specific sub-problem for the organisation: it is not clear whether the project leader should be from a unit that is responsible for the main activities or whether he or she should be from an IT unit and know well the modern IT-project management technologies. Both related departments pointed out that this might be the responsibility of another department. The additional question is of how user requirements will be considered during IS development.

The current case study in this thesis (as a subset of actual analysis) concentrates on:

1. Specifying tasks for roles: Identifying what resources are necessary for the current project
2. How the overall tasks for different project leader roles could be allocated between actors -- Suggesting suitable mapping: Mapping the requested resources to the existing organisation – detailing what changes are needed.

The structural units of the organisation related to the current case study are: development department (DD, responsible for organisational development), information technology department (ITD, responsible for the development and maintenance of police information systems) and police unit (ORG – representative of users).

The **initially suggested list of key actors** is given in Table 5.13.

Table 5.13 Initial mapping of roles and actors for project team

No	Role	Notation[7]	Task, description	Possible actor
1	Project leader	PL	A person from the main business side (in this case the police) who knows the main activity processes	An official from ORG (local level)
2	Representative of users	USR	To acquire user requirements and evaluation of the software	A number of officials from ORG (local level)
3	IT project leader	IPL	A person who knows the capabilities of the technology and their suitability into planned solutions	An IT-specialist from ITD
4	IT company project leader	CPL	To manage actual IS development in the IT development company	A project manager form IT partner company

Table 5.14 and Table 5.15 focus more specifically on actors PL and IPL respectively and list the most important tasks for these project leaders. Tabels are given in the form of *process table W*. The columns “Purpose” and

[7] As used in the UML and agent models in the case study

"Remarks" are omitted and a comment is added to a specific activity, if necessary.

Analysis of the initial task mapping. The use case diagram in Figure 5.24 also illustrates the need for different project leaders. Three different roles were distinguished: a project leader for the whole project (a specialist in the main processes), an IT-project leader (responsible for all IT-related aspects of the project) and a project leader form an IT-development company (responsible for the development of the specific software). The figure also indicates that an IT-project leader cannot be responsible for the overall project because of non-competence and executive power for organising the implementation and changing work process in the organisation.

Table 5.14 Some work processes of PL

Pro-cess ID	Activity name	Input		Action description	Du-ra-tion[8]	Ac-tor	Output	
		Des-cription	Time crite-ria[9]				Des-cription	Vali-dity[10]
q	*n*	μ	*t*	*d*	φ	*a*	ν	τ
1	2	4	5	6	7	8	9	10
P1	Project co-ordi-nation[11]	Project plan	s	Leading the project team, checking the coherence of different activities	c	PL	Project	p
P3	Project reporting[12]	Project	s	Reporting to the steering committee (regularly)	1h	PL	Report	c
P4	Organi-sing of implemen-tation	Elabo-rated SW	s	Organising of the project implemen-tation in the organisation	c	PL	Imple-mented project	p
P5	Project evaluation	Imple-mented project	s	Evaluation of the managed project	1d	PL	Evalua-tion	p
P7	Process modify-cation	Elabora ted SW, existing process es	s	Organising of the modification of the existing work processes, if needed	c	PL	Modi-fied work routines	f

[8] Used values: 'h' – hour(s), 'd' – day(s), 'w' – week(s), 'c' – continuous (until the end of the project)
[9] Used values: 's' – information is obtained on starting the process, 'c' – continuous
[10] Used values: 'p' – permanent during this project, 'c' – continuous until new information updates are existing, 'f' – final for this task
[11] Coherent and integrated management of all parts of the project
[12] In-time review of the project to the management group

Table 5.15 Some work processes of IPL

Process ID	Activity name	Input		Action description	Duration	Actor	Output	
		Description	Time criteria				Description	Validity
q	*n*	*μ*	*t*	*d*	*φ*	*a*	*ν*	*τ*
1	2	4	5	6	7	8	9	10
S2	SW requirements specification	User requirements	s	Specification of the requirements for software	1w	IPL	SW Requirements	f
S3	Architecture control	Existing architectures, proposed architecture	s, c	Checking whether the proposed software architecture is suitable for integrating with the existing one	c	IPL	Suitable Architecture, Remarks	c
S4	SW testing	SW	s	Checking the program code (with own team)	2..3w	IPL	Checked SW	f
S5	User requirement check	User requirements	s	Checking the correspondence of user requirements to the existing policies and standards	1w	IPL	Checked requirements	f
S6	Co-ordination between users and elaborating company	Project	s	Co-operating in the transition of user requirements to a development company	c	IPL	Good project	p
S7	SW technical implementation	SW	s	Implementation of software product to hardware and into the existing systems and solutions	1w	IPL	Implemented SW	f
S8	SW development co-ordination	Project	s	Communicating with software development company	c	IPL	SW	f

It demonstrates also that an IT partner company project leader cannot have the overall power without having a competent client from the organisation who solves intra-organisation problems and a competent IT expert who can evaluate the code produced by the IT partner company.

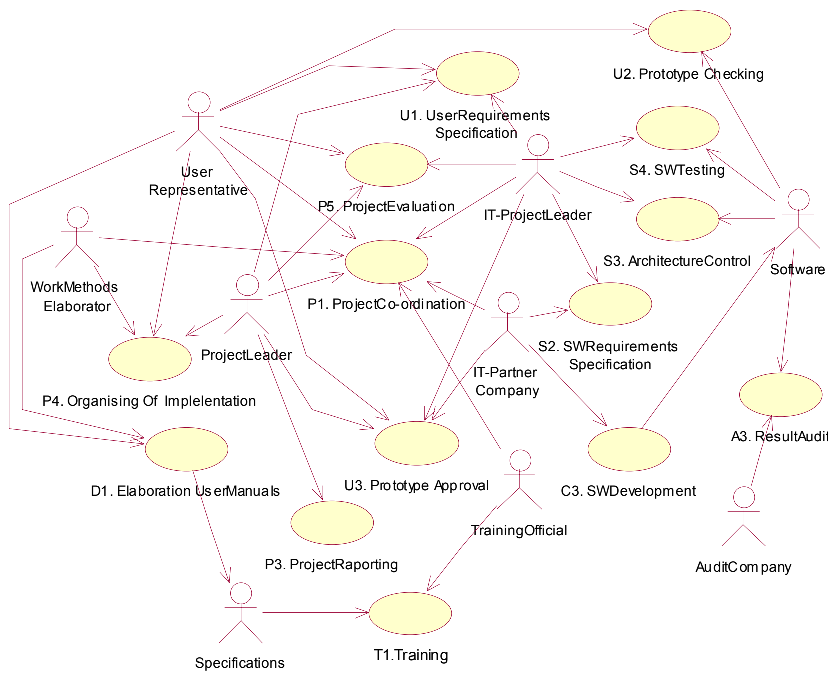

Figure 5.24 The use case diagram of project leader, user representative, IT-project leader and IT-company project leader

The activity diagram in Figure 5.25 illustrates the overall process sequence in the project team related to the management of the project.

In the current case study, the objective of the analysis was to model specific tasks of different roles (actors). Timing criteria were not crucial and collisions of resources over time were not analysed in this case study. Therefore the Q-model analysis was omitted. The Q-model is useful in a more detailed analysis of actual resource allocation during the project lifecycle and the modelling of different interests. In such case the analysis will be similar to the first case study (Sections 5.1 – 5.5).

The multi-agent simulation concentrates on different interests and priorities of three actors: the project leader, the IT-project leader and the user representative. The multi-agent simulation pointed out that:

1. The project leader may have different priorities in relation to its own authority and the overall project task.
2. The IT project leader may have too many tasks related to other IT-projects.
3. The user representative may have a conflict of priorities between everyday tasks and participation in the project.

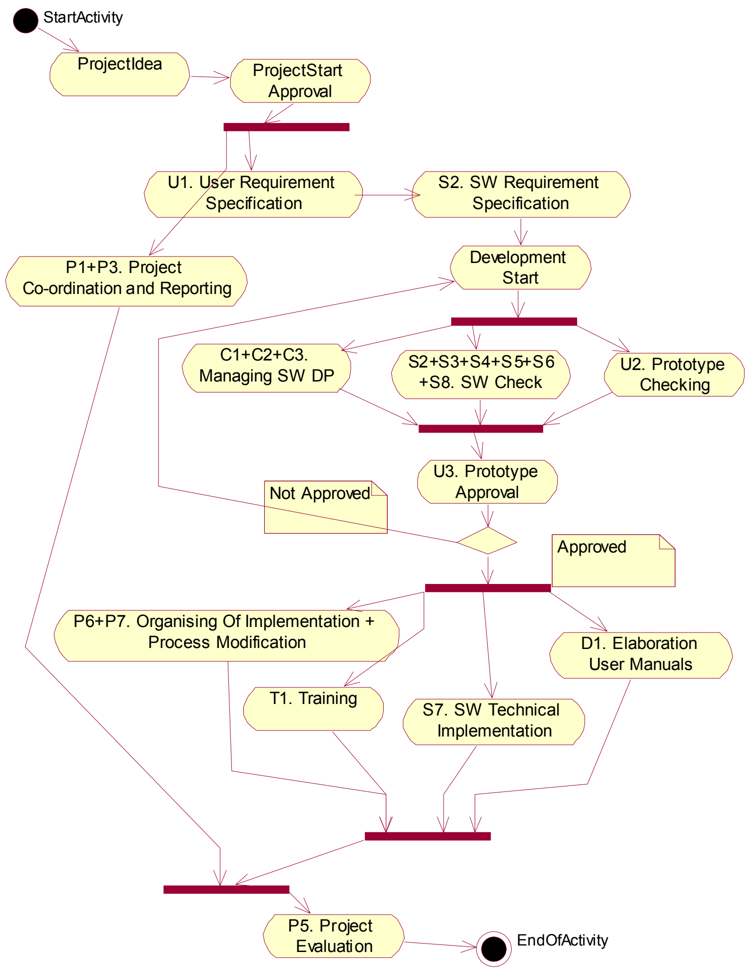

Figure 5.25 The activity diagram of activities of the project team

The indicated problems are classical resource allocation and task prioritisation issues (this was also expressed for the actor LC in the first case study. Here the problem can be partially solved by the prioritisation of projects. As an example, the class diagram of an IT-project leader is given in Figure 5.26. IT-projects are prioritised according to organisational tasks. The structure for representing the personal characteristics of IPL is taken from the description in Section 4.4 of the thesis.

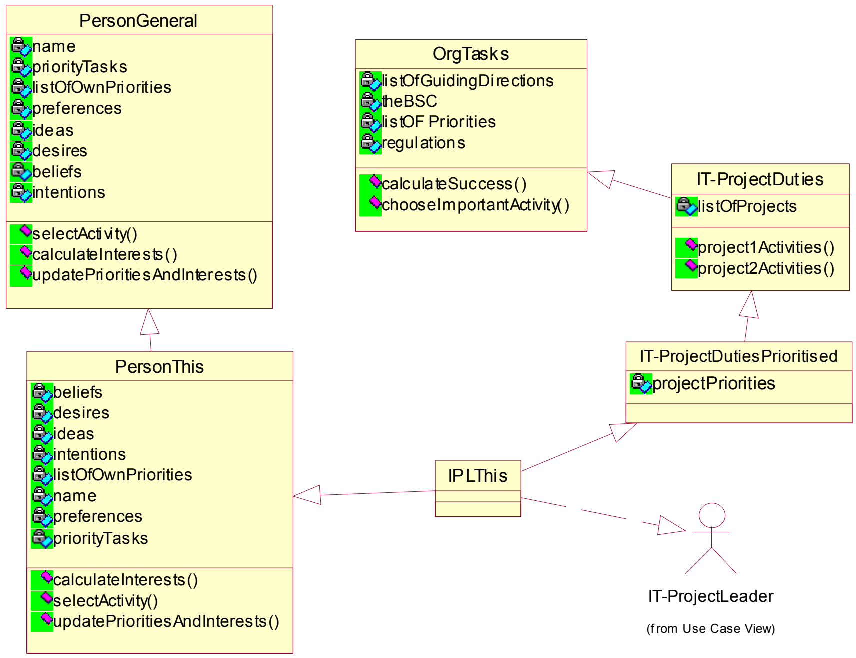

Figure 5.26 The simplified class diagram of IT project leader

The computerised MAS simulation is indicative and expresses communication between different actors. The MAS simulation should, in principle, also support the performance analysis of the project team in case all actors have also other multiple tasks (some with higher priority, some with less).

As a result of the analysis, some changes were introduced in the roles of project leader and representative of users. As a result, the leader of the project team (actor PL) is responsible for development and is a representative from the main activity unit. The IT-project leader (IPL) has the task of a classical project leader in the team (to uphold the efficiency of the project). Modified role mapping for project team is given in Table 5.16.

This case study demonstrated that the methodology could be implemented also in such cases, especially for analysing concurrent priorities and different preferences of actors. Since temporal characteristics were not modelled in this example, other more suitable approaches can be useful instead of the introduced methodology.

Table 5.16 Suggested mapping of roles and actors for the project team

No	Role	Notation	Initially chosen actor (as in Table 5.13)	Suggested actor after the analysis
1	Project leader	PL	An official from ORG (local level)	A senior official from the corresponding unit at the central level
2	Representative of users	USR	A number of officials from ORG (local level)	A number of policemen from the central level and local level

5.8 CONCLUSIONS ON IMPLEMENTATION OF THE METHODOLOGY

This chapter reviewed three case studies which illustrated the implementation of the methodology. The first case study (Sections 5.1 – 5.4 in the thesis) described the analysis and modelling of behaviour related to police activities in case of vehicle theft. Section 5.5 extended this case study and demonstrated that the behaviour of one key actor depends on its personal preferences and overall workload. The second case study (in Section 5.6) illustrated the modelling of task allocation and co-operation between different structural units in the same organisation. Finally the third case study (Section 5.7) analysed and suggested solutions for efficient task allocation and the fulfilment of roles for a project management team.

In all three case studies the intention was to follow the methodology as much as possible. The process started with setting up the analysis task, followed by the description of the organisation (or the existing situation) and the specification of key processes and actors. In all three case studies the organisational need brought about the modelling activities and a brief summary of the actual analysis and modelling of each case was presented in the present thesis.

The goals that were set up in the initial analysis and modelling tasks were more or less fulfilled in all case studies. The presented examples analysed the existing processes and suggested some improvement in all cases. This indicated that a more detailed analysis and modelling is beneficial to the overall result. The implementation of multi-agent approach for modelling employees and their decision-making was very suitable in the current application domain.

The given case studies permit to illustrate the usage of the methodology in different aspects. The obtained experience indicated that the modelling methodology is in principle implementable. The modelling results are more resultant in cases where the modelling task is to specify the processes on a very detailed level and where correct timing of processes is essential (e.g. the first case study). In cases where the organisation concentrated more on the principal task allocation and co-operation between different actors (e.g. departments or employees) as in the second and third case study, the importance of the Q-model component and the modelling of timing criteria decreased and the model was more concentrated on processes (process table, UML use case and activity diagrams) and multi-agent analysis (agent class models and multi-agent simulation, if the latter is considered to be worth of time spent for the model devising). Therefore it can be concluded that for the special modelling of actors' co-operation and interactions in non-time-critical environment the agent part of the methodology must be further elaborated. Of course, if organisational conditions and modelling tasks differ considerably from the initial conditions (described in Sections 1.1 and 1.2), the use of the introduced methodology may not be justified and other solutions should be found.

At the same time the experiments demonstrate that the modelling methodology is not a self-modelling and easy-describe system. Because of the generality of the methodology, the composed model may become complicated and its processing time-consuming. The composing of the timing model in the Q-model notation is most sophisticated, but on the other hand the Q-model shows the real (timing) co-influence and bottlenecks in the best way. Also the case studies indicated that sometimes the devising of all model components is too time-consuming (in comparison with the results).

It is highly suggested to limit the extent of the modelling and the scope of the analysis as mush as possible in each specific case. The resulting document shows that a relatively big amount of work is needed for the detailed analysis of quite a simple problem and a lot of information must be processed. This is also similar to other approaches (e.g. EKD-CMM).

6 SUMMARY AND CONCLUSIONS

The present thesis has treated the modelling of emergent operational behaviour in multi-functional organisations that operate in a dynamic environment. The thesis has focused on time-aware specification and simulation of processes and actors' behaviour in order to support the planning of process modifications. Common characteristics of the considered organisations, the requirements for models and the existing modelling methods were reviewed.

The thesis emphasizes the importance of interactive approach and outlines suitable modelling techniques (the UML, the Q-model and agent technologies). Also a novel methodology was presented and illustrated by a specific example. The novelty of the approach lies in handling an organisation simultaneously from multiple viewpoints: on the one hand work processes in the organisation are modelled by using the UML and the Q-model, thus process control techniques are incorporated for the specification of processes and their temporal characteristics. At the same time an organisation (and its actors) is also modelled as a hierarchical multi-agent system. Different viewpoints are collected into an integrated *organisation model*.

6.1 EXISTING WORK

The thesis has summarised the current research of the author on this problem. The research consisted of three phases:

1. Description of the problem, specification of model requirements and the review of the existing modelling approaches.
2. Introduction of a suitable novel *organisation model* and a methodology for devising the model.
3. Review of experiments and modelling cases.

The thesis formulated the problem (in Section 1.2) that no suitable solutions exist for the resultant modelling of time-sensitive operational emergent behaviour of multi-functional human organisations (e.g. the police, the rescue services, etc). This problem has emerged from the needs of real life. As a starting point, a review of the existing conventional approaches for analysing and modelling behaviour and processes in a human organisation was presented in the present thesis (Chapter 2). The required analysing power of models and the incapabilities of the existing models were also brought out.

The thesis formulated a hypothesis (in Sections 1.2 and 2.4) that suitable solutions could be found in the combination of soft real time modelling and multi-agent systems engineering. The principles and known approaches of the interaction-based paradigm and the modelling of temporal criteria were reviewed in Chapter 3. The chapter proceeds with presenting a number of existing approaches related to the engineering of multi-agent systems and the representation of human organisations in agent systems.

Based on the previous chapters of the thesis, a novel *organisation model* and a methodology for devising the corresponding models were introduced in Chapter 4 of the thesis. The *organisation model* integrates different views of the same organisation: processes are specified by using the UML, temporal criteria of the processes and their interactions are modelled by using the Q-model (a real-time model processor for the UML, corresponding to UML Profile for Schedulability, Performance and Time) and personal characteristics of human actors and the simulation of the overall behaviour of the system are devised as a multi-agent system. The model consists of four components:

1. A document that comprises the *general description of the organisation* together with its structural model, the *process table*, the description of use cases and processes in the UML use case and activity diagrams, diagrams in the Q-model and agent class diagrams in the UML for the modelling of employees as agents.
2. UML model (that actually contains use case, activity, class and sequence diagrams).
3. The process model in the Q-model simulation environment.
4. The multi-agent simulation model.

The components of the model are kept in different modelling environments for the UML, the Q-model and agent simulations. The implementation of computerised modelling environments and simulations enables to use more exact models for describing the organisation.

Support for predicting the expected behaviour of the organisation during and after the modification of processes was given in the thesis by introducing the *change model* – an inter-connected set of *organisation models*.

The methodology for devising the *organisation model* and the *change model* comprises a stepwise action plan. The methodology consists of four stages, which in turn consist of steps (altogether 15). The first stage comprises the problem description, the composition of analysis task and the description of the organisation and its processes in the UML. The second stage completes the model of the organisation by adding the Q-model, agent components and multi-agent simulation. Proposals for modification, based on the modelling task, are given in the third stage of the methodology. The *change model* can be used as a supportive material in the fourth stage of the methodology (implementing modification suggestions in the real organisation).

The suggested models and their devising methodology have been tested for solving some modelling tasks in a real organisation. Chapter 5 has presented three examples as an illustration for the implementation of the approach. The main case study – information exchange in the police on vehicle theft – illustrated that the methodology is fully suitable for modelling operational time-dependant emergent behaviour for short time criteria (minutes, seconds). The amount and detailness of implementing the methodology depends on the modelling task. As seen from other case studies (co-operation and task allocation between structural units and devising a project organisation for information systems development), the methodology is in principle also suitable for the modelling of more strategic and tactic co-operation and interactions of multiple actors.

6.2 MAIN RESULTS AND CONCLUSIONS

The main conclusion of the thesis is: time-sensitive emergent operational behaviour of a human multi-functional organisation can be efficiently modelled. A suitable solution is simultaneous modelling from the viewpoints of process and actors, using a model that combines the UML, the Q-model and agent technologies. The corresponding methodology is introduced in Chapter 4. New and important aspects of the suggested methodology are:

1. Agents are added to the description of the organisation to give more realistic behaviour by incorporating viewpoints of different actors.
2. The *organisation model* that integrates given components is new.
3. The *Change model* that supports the planning of modifications by capturing successful modification activities on the model (the general idea itself is similar to EM methods).
4. The importance of continuous actualisation of the model and its implementation while preparing any kind of decisions in the organisation is stressed. The issue of process re-engineering and component reuse is not new but the present thesis shows one possible implementation for it.

The main conclusion together with the presented material is the main results of the thesis. Four additional conclusions can be made on the basis of the thesis and the studied literature:

1. Known conventional methods are not aimed at and are not suitable for modelling operational time-dependant emergent behaviour. The review presented in Chapter 2 has demonstrated that the requirements for models indicate more aspects than the existing classical methods can analyse, therefore a suitable methodology has to emphasize interactivity issues. The proposed methodology as shown in Chapter 4 fulfils the requirements given to models in Chapter 2.
2. The behaviour of employees is possible to describe by an agent technology approach. The review presented in Section 3.3 illustrated that multi-agent systems are often used to model some issues of organisational behaviour. A stepwise review of different aspects (social intelligence, task decomposition, motivation, co-operation and

interactions) in Section 3.4 demonstrated the possible implementation of those questions also for human organisations.

3. In the organisation both the procedures and behaviour of actors can be modelled simultaneously (as described in Chapter 4).
4. The agent technology together with the Q-model and the UML is suitable for describing at least some types of organisations. This was shown by illustrative examples and case studies in Chapter 5.

In Section 1.2 the thesis pointed out four hypotheses. The research and results of the thesis confirmed the first and the second hypotheses. The introduced model and related modelling methodology also confirmed the third hypothesis. It is not analysed whether this approach is the best and therefore there are no claims to the optimality of such an approach for more general domains.

The fourth hypothesis was confirmed partially. The evaluation of effectiveness and usefulness depends on the overall stage of planning and modelling in the organisation and is tightly related to each specific organisation. In current circumstances (given in Sections 1.2 and 2.1) the solution is efficient and useful, but it should be noticed that the implementation of the model is time-consuming.

In Section 1.4 the expected contribution to the problem solving was stated. The current results satisfy the expectations – the suggested modelling methodology solves the problem of the adequate modelling of the operational emergent behaviour of organisations under given conditions in the pre-described activity domain. It also visualises the possible decision-making choices and to the some extent prognosticates the behaviour of the modified organisation. The results of the research in comparison to the research objective can be evaluated as satisfactory since the main goals were achieved.

6.3 FUTURE ACTIVITIES

The suggestions made in the present thesis regarding the modelling of emergent behaviour are not final and further work is needed to detail the approach that would cover all aspects of the given problems. Future work, related to the proposed methodology, would comprise the following four activities.

First – further development of the methodology. The methodology is presented completely in the present thesis but because of its relatively recent elaboration there are only a few modelling tasks solved in practice. It should also be noted that so far only trivial agent models have been used. The multi-agent part of the model and its analysis becomes more resource and time demanding when realistic cognitive models of agents are applied. A more detailed description on how to simulate an organisation by using agent technology is under development. Formal analysing tools for interaction-based computing are not yet available either, although their development in the world is supported by OMG initiatives in developing real-time extensions for the UML and by ongoing research in applying agent-based paradigm (e.g. Meriste and Motus, 2002). Encapsulation of the dynamic behaviour of an organisation during modifications should be described in more detail. Since the UML Profile for time and performance incorporates most of the principles of the Q-model and as the UML 2.0 offers support for this, there might be adjustments in the integration of the components of the models.

Second – the elaboration of a unified tool that supports the introduced modelling methodology. A number of activities in the proposed methodology currently contain manual work on storing and modifying different parts of the model in different software environments. To simplify the process and to get a more realistic simulation, a unified modelling tool should be used. The latter is more effective because errors are more clearly seen and it is possible to visually simulate the performance of an organisation. Real models are quite sophisticated and therefore manual modelling becomes too complex. The same criterion applies also to modelling when using agent technologies. In addition, more specialised test beds can also be elaborated that permit the simulation of organisations in certain domains. Such a solution enables a uniformed approach for modelling a particular group of organisations. As a starting point, it could be useful for building a piece of software that filters the agent simulation log file dynamically according to the wishes of the viewer. Future possible elaboration of a computerised tool should also consider the development and integration of popular modelling tools for the UML, agent technologies and the Q-model.

The third necessary activity is a more detailed specification of how to use the agent model for describing different aspects of the organisation and actors' behaviour. For example, the representation of human characteristics and informal communication in the organisation has to be modelled better by using simulations on MAS. This elaboration is related to the development of agent modelling environments.

Finally, the fourth foreseen activity is the extension of application domain of the methodology to different types of organisations. The methodology could to some extent be used in different modelling domains and for other modelling aims, as well (like evaluation and comparison of the efficiency of several organisations). More general suggestions could also be derived from different simulations of specific organisations (e.g. recommendation about structure, task allocation, planning of co-operation, etc.)

REFERENCES

Abeysinghe, G. and Phalp, K. (1997), “Combining process modelling methods”, *Information and Software Technology,* Vol. 39, 1997, Elsevier Publishers, pp. 107-124.

Alatalo, T., Oinas-Kukkonen, H., Kurkela, V. and Siponen, M. (2002), “Information systems development in emergent organizations: empirical findings”, *Information Systems Development, Advances in Methodologies, Components and Managements,* ed. M. Kirikova et al. Kluwer Academic / Plenum Publishers, New York, pp. 115-122.

Amin, M. and Ballard, D. (2000), “Defining New Markets for Intelligent Agents”, *IT Professional,* 2000, Vol. 2, No. 4, IEEE, pp. 29-35.

Apt, K. R. and Olderog, E.-R. (1997), *Verification of Sequential and Concurrent Programs,* Second Edition, Springer-Verlag New York Inc.

Aoyama, A., Batres, R. and Naka Y. (2001), “Technological Information Infrastructure for Process Industries”, International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, L’Aquila, Italy, August 06-12, 2001.

APD, 1992. *Academic Press Dictionary of Science and Technology*, ed. Christopher Morris, Academic Press, Harcourt Brace Jovanovich Publishers, San Diego, California, USA, 1992.

Arcieri, F., Giaccio, R., Nardelli, E. and Talamo, M. (2001), “A Framework for Inter-Organizational Public Administration Network Services”, International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, L’Aquila, Italy, August 06-12, 2001.

Arlow, J. and Neustadt, I. (2002), *UML and the Unified Process,* Pearson Education, 2002.

Balkenius, C. (1995), *Natural Intelligence in Artificial Creatures,* Doctoral Dissertation, Department of Cognitive Science, Lund University, Sweden, 1995. ISRN LUHFDA/HFKO—103—SE.

Barber, K. S., Goel, A., Han, D., Kim, J., Liu, T. H., Martin, C. E. and McKay, R. (2004), “Problem-Solving Frameworks for Sensible Agents in an Electronic Market”, *Multiple Approaches to Intelligent systems: 12th International Conference on Industrial and Engineering Applications of Artificial Intelligentce and Expert systems IEA/AIE-99, Cairo* (eds. I. Imam, Y. Kodratoff, A. El-Dessouki, et al.). *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 1611, pp. 470-479.

Barrios, J. and Nurcan, S. (2004), “Model Driven Architectures for Enterprise Information Systems”, *Advanced Information systems engineering: 16th International Conference, CaiSE 2004, Riga* (eds. A. Persson, J. Stirna). *Lecture Notes in Computer Science,* Springel-Verlag, Vol. 3084, pp. 3-19.

Bauer, B. and Müller, J. P. (2003), “Using UML in the Context of Agent-Oriented Software Engineering: State of the Art”, *Agent-Oriented Software*

Engineering IV (P. Giorgioni, J. P. Müller and J. Odell, eds.). *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 2935, pp. 1-24.

Bauer, B., Mueller, J.P. and Odell, J. (2000), "An Extension of UML by Protocols of Multiagent Interaction", *Proc. Fourth Int. Conference on Multi-Agent Systems*, Boston, Massachusetts, USA, pp. 207-214.

Bellifemine, F., Poggi, A. and Rimassa G. (2001), "Developing Multi-Agent Systems with JADE", *Proc. Intelligent Agents VII, Agent Theories Architectures and Languages: 7th International Workshop, ATAL 2000*, Boston, MA, USA, July 2000, *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 1986, p 89.

Bellifemine, F., Poggi, A. and Rimassa G. (1999), "JADE – A FIPA-compliant agent framework", *Proc of 4th International conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents, PAAM'99*, London, pp. 97-108.

Bergenti, F. and Poggi, A. (2000), "Exploiting UML in the Design of Multi-agent Systems", *Engineering Societies in the Agents world: First International Workshop, ESAW 2000,* Berlin, Germany, August 2000. Lecture Notes in Computer Science, Springer-Verlag, Vol. 1972, pp. 106-113.

Best Value (2003a), *The Best Value website*, http://www.scotland.gov.uk/about/FCSD/LG-PERF4/00014838/Home.aspx/, Accessed 7 July, 2003.

Best Value (2003b), *Best Value Performance Indicators web site*, http://www.bvpi.gov.uk/home.asp, Accessed 7 July, 2003.

Booch, G., Rumbaugh, J. and Jacobson, I. (1999), *The Unified Modelling Language. User Guide,* Addison Wesley Longman, Inc., Reading, Massachusetts, USA, 1999.

Bonjour, E., Dulmet, M. and Lhote, F. (2002), *An internal modelling of competency, based on a systemic approach, with socio-technical systems management in view*, http://ieeexplore.ieee.org, Accessed 26 August, 2003.

Bose, R. and Burd, S. D. (1997), "Control and coordination of heterogeneous transaction processing systems", *Information and Software Technology*, Vol. 39, pp. 171-184.

Boyne, G. (2000), "External Regulation and Best Value in Local Government", *Public Money & Management*. 2000, Vol. 20 No. 3, pp. 7-12.

Britanik, J. M. and Marefat, M. M. (1999), "Hierarchically Merging Plans in Decomposable Domains", *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans,* Vol. 29, No. 1, pp. 27-40.

Brooks, R. A. (1986), "A Robust Layered Control System For A Mobile Robot", In: *IEEE Journal of Robotics and Automation,* Vol. RA-2, No. 1, pp. 14-23.

Bubenko, J. A. jr., Persson, A. and Stirna, J. (2001), *User guide of the Knowledge Management approach using Enterprise Knowledge Patterns.* Deliverable D3, IST Programme Project HyperKnowledge – Hypermedia and Pattern Based Knowledge Management for Smart Organisations, project

No. IST-2000-28401, Appendix B: EKD User Guide. Department of Computer and System Sciences, Royal Institute of Technology, Stockholm.

Burrafato, P. and Cossentino, M. (2002), "Designing a multi-agent solution for a bookstore with the PASSI methodology", *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*, 27-28 May 2002, Toronto (Ontario, Canada).

Carley, K. M. and Prietula, M. J. (1998), "Webbots, Trust and Organizational Science", *Simulating Organizations: computational models of institutions and groups,* eds M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp 3-22.

Carley, K. M. and Prietula, M. J. (1994), "ACTS Theory: Extending the Model of Bounded Rationality", *Computational Organization Theory,* eds. K. M. Carley and M. J. Prietula, Hillsdale, N. J.: Lawrence Erlbaum Associates, pp. 4:55-89.

Caspi, P. and Halbwachs, N. (1986), "A functional model for describing and reasoning about time behaviour of computing systems", *Acta Informatica,* Vol. 22, Nr 6, Springer-Verlag NY Inc., pp. 595-627.

Castano, S. and Antonellis, V. (1997), "Engineering a library of reusable conceptual components, *Information and Software Technology,* Elsevier Science B.V., Vol. 39, pp. 65-76.

Castano, S., De Antonellis, V. and Melchiori, M. (1999), "A methodoly and tool environment for process analysis and reengineering", *Data & Knowledge Engineering,* Vol. 31, pp. 253-278.

Chaib-draa, B. (2002). "Causal Maps: Theory, Implementation and Practical Applications in Multiagent Environments", *IEEE Transactions on Knowledge and Data Engineering,* Vol. 14, No. 6, pp.1201-1217.

Clark, W. (1922), *The Gantt chart: a working tool of management*, New York: Ronald Press. Third edition: 1952, Pitman and sons, London.

Coderre (1988), "Modelling Behavior in Petworld", In: *Artificial Life,* SFI Studies in Sciences of Complexity, Ed. C. Langdon, Addison-Wesley Publ Co., pp. 407-420.

Cohen, G. P. (1992), *The Virtual design Team: An Information Processing Model of Coordination if Project Design Teams*, Ph.D. diss., Dept. of Civil Engineering, Stanford Univ., Stanford, Calif.

Cohen, M. D. (1986), "Artificial intelligence and the Dynamic Performance of Organization Designs" *Ambiguity and command: Organizational Perspectives on Military Decision Making,* ed. J. G. March and R. Weissinger-Baylon, Pitman Publishing, Marshfield, Massachusetts, USA, pp. 53-71.

Cossentino, M. and Potts, C. (2002), "A CASE tool supported methodology for the design of multi-agent systems", *The 2002 Int. Conf. On Software Engineering research and Practice (SERP'02),* June 24-27. 2002 Las Vegas, NV.

Decker, K. S. (1998), "Task Environment Centered Simulation", *Simulating Organizations: computational models of institutions and groups,* edited by

Michael Prietula, Kathleen Carley and Les Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp 105-128.

DeLoach, S. A. and Wood, M. (2001). "Developing Multiagent Systems with agentTool", *Proc. Intelligent Agents VII. Agent Theories Architectures and Languages: 7th International Workshop, ATAL 2000*, Boston, MA, USA, July 2000, (eds. C. Castelfranchi, Y. Lesperance). Lecture Notes in Computer Science, Springer-Verlag, Vol. 1986, pp. 46-60.

Denno, P., Flater, D., Gruninger, M. (2001), "Modeling Technology for a Model-intensive Enterprise", *International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, L'Aquila, Italy, August 06-12, 2001*, www.nist.gov/msidlibrary/doc/mt-for-mda.pdf, Accessed 7 February 2005;

DiCesare, F., Harhalakis, G., Proth, J.-M., Silva-Suarez, M. and Vernadat, F. (1993), *Practice of Petri Nets in Manufacturing,* Kluwer Academic Publishers, Dordrecht.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), "The Ant System: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man and Cybernetics – Part B,* vol. 26, No 1, pp. 1-13.

Dosch, W. (2003), "Stream-Based Design of Interactive Systems", lecture material in the 2nd Estonian Summer School on Computer and Systems Science (ESSCaSS'03), Taagepera Castle, Estonia, August 10-14, 2003, http://greta.cs.ioc.ee/yik/schools/sum2003/, Accessed 1 September 2003.

Drogoul, A. and Zucker, J.-D. (1998), "Methodological Issues for Designing Multi-Agent Systems with Machine Learning Techniques: Capitalizing Experiences fron the RoboCup Challenge".

Dugdale, J. and Pavard, B. (2000), "The application of a cognitive engineering methodology to agent based simulation in the social sciences", *Proceedings of Agent-Based Simulation Workshop 2000*. May 2-3, Passau, Germany. C Urban (eds.); Workshop 2000: Agent-based simulation. Workshop 2000: Agent-Based Simulation (Passau, Germany, May 2-3,2000), SCS publications, Ghent, Belgium, 2000, ISBN: 1-56555-185-0

Dugdale, J., Pavard, B. and Soubie, J. L. (2000), "A pragmatic development of a computer simulation of an emergency call centre", *Designing Cooperative Systems. The use of theories and modules – Proc. Of the 5th Int. Conf. On the Design of coop syst (COOP'2000)*. (eds. R. Dieng et al.) Vol. 58, IOS Press, Amsterdam.

Edmonds, B. (1997), "Modelling Socially Intelligent Agents", *Applied Artificial Intelligence*. 1998, 12 (7-8), Taylor and Francis, Ltd, pp. 677-699.

EFQM (2005), *EFQM Excellence Model - Public and Voluntary sectors*, EFQM, ISBN 90-5236-319-6.

ELEKTRA (2002), *Electrical Enterprise Knowledge for Transforming Applications (Elektra)*, Esprit Programme 7.1, Project No 22927.

Engelmore, R., Morgan, T., ed.-s (1988), *Blackboard Systems,* Addison-Wesley, 1988.

ESSCaSS (2003,) 2nd Estonian Summer School on Computer and Systems Science (ESSCaSS'03), II Eesti Arvuti- ja Süsteemiteaduse Suvekool (EASTS'03), Taagepera Castle, August 10-14, 2003, http://www.cs.ioc.ee/yik/schools/sum2003/, Accessed 10 june 2005.

ESSCaSS (2004), 3rd Estonian Summer School on Computer and Systems Science (ESSCaSS'04), III Eesti Arvuti- ja Süsteemiteaduse Suvekool (EASTS'04), Pedase, August 8-12, 2004, http://www.cs.ioc.ee/yik/schools/sum2004/, Accesed 10 June 2005.

Fayek, R. E., Liscano, R. and Karam G. M. (1993), "A System Architecture for a Mobile Robot Based on Activities and a Blackboard Control Unit", *Proc., 1993 IEEE International Conference on Robotics and Automation,* May 2-6, 1993, Atlanta, Georgia, USA, pp. 267-274.

Fasli, M. (2003), "From Social Agents to Multi-agent Systems: Preliminary Report", *Multi-Agent Systems and Applications III: 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003, Prague, Czech Republic, June 16-18, 2003.* Proceedings, LNAI, Springer-Verlag, Vol. 2691, p. 111.

Ferber, J. (1999), *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence,* Addison-Wesley, Harlow, Essex, 1999.

Ferber, J., Gutknecht, O., Jonker, C. M., Müller, J.-P. and Treur, J. (2000), "Organization Models and Behavioural Requirements Specification for Multi-Agent Systems", *Proc. Fourth International Conference on MultiAgent Systems,* Boston, Massachussetts, July 10-12, 2000, pp. 387-388.

FIPA (2003), FIPA Modeling Area: Temporal Constraints, Working Draft, version 1.0, 2003-04-18, http://www.fipa.org, Accessed 10 June 2005.

FIPA (2005a), *The FIPA Web Site*, Federation of Intelligent Physical Agents, http://www.fipa.org/, Accessed 10 June 2005.

FIPA (2005b), *The AUML web-page*, http://www.auml.org, Accessed 10 June 2005.

FIPA TC (2005), *FIPA Modelling Technical Committee*, http://www.fipa.org/activities/modeling.html, Accessed 10 June 2005.

Flake, S., Geiger, C. and Küster, J. M. (2001), "Towards UML-based Analysis and Design of Multi-Agent Systems", *International NAISO Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems*, (ENAIS 2001), In: *Proc of International NAISO Congress Information Science Innovations (ISI'2001)*, March 17-21, 2001, Dubai, 2001. Naiso Academic Press, pp. 695-701.

Fox, M. S. (1981), "An Organizational View of Distributed Systems", *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-11, No. 1, pp. 70-80.

Fox, M. S., Barbuceanu, M., Gruninger, M. and Lin, J. (1998), "An Organizational Ontology for Enterprise Modeling", *Simulating Organizations: computational models of institutions and groups,* eds. M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp 131-152.

Fougères, A.-J. (2002), "Model of cognitive agents to simulate complex information systems", *IEEE International Conference on Systems, Man and Cybernetics*, Vol 4, 6pp.

Frederiks P. J. M., Hofstede, A. H. M. and Lippe, E. (1997), "A unifying framework for conceptual data modelling concepts", *Information and Software Technology*, Elsevier Science B.V., Vol. 39, pp. 15-25.

Giraud (1984), "Generalized active Compliance for Part Mating with Assembly Robots", *Robotic Research. The First International Symposium.* Eds. M. Brady and R. Paul. The MIT Press, 1985, pp. 949-960.

GRIC (2005), Cognitive Engineering Research Group, http://www.irit.fr/GRIC, Accessed 7 Feb 2005

Hannoun, M., Boissier, O., Sichman, J. S. and Sayettat, C. (2000), "MOISE: An Organizational Model for Multi-Agent Systems", *Proc. Advances in Artificial Intelligence: International Joint Conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI, IBERAMIA-SBIA 2000,* Atibaia, SP, Brazil, November 2000, Lecture Notes in Computer Science, Springer-Verlag, Vol. 1952, p 156.

Hoare, C. A. R. (1985), *Communicating Sequential Processes,* Prentice Hall.

Hogg, L. M. J. and Jennings, N. R. (2001), "Socially Intelligent Reasoning for Autonomous Agents", *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans,* Vol. 31, No. 5, pp. 381-393.

Holsapple, C. W. and Joshi K. D. (2002), "A Collaborative Approach to Ontology Design", In: *Communications of the ACM,* February 2002, vol. 45, No. 2, pp. 42-47.

Huberman, B. A. and Glance, N. S. (1998), "Fluctuating Efforts and Sustainable Cooperation", *Simulating Organizations: computational models of institutions and groups,* eds M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp. 89-103.

IPF (2003), *Best Value Theme*, http://ipf.co.uk/bestvalue/, Accessed 7 Feb 2005

ISACA (2005) *Cobit*, Information systems Audit and Control Association, http://www.isaca.org/cobit,htm, Accessed 18 February 2005.

ISO (2000), *ISO/IEC 17799:2000 Information Technology - Code of practice for information security management*, http://www.iso.org/iso/en/prods-services/popstds/informationsecurity.html, Accessed 18 February, 2005.

ISO (2003), *ISO 9000*, International Organization for Standardization, http://www.iso.org/iso/en/iso9000-14000/iso9000/iso9000index.html, Accessed 24 November 2003.

Iwazume, M., Kato, Y. and Kanai, A. (2001), "A multi-agent based model for analyzing human cognitive process of advertising information", *The 5th World Multi-Conference on Systems, Cybernetics and Informatics,* Vol XIV.

Kang, M., Waisel, L. B. and Wallace, W. A. (1998), "Team Soar: A Model for Team Decision Making", *Simulating Organizations: computational models of institutions and groups,* eds M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp 23-45.

Kaplan, D. J. and Carley, K. M. (1998), "An Approach to Modeling Communication and Information Technology in Organizations", *Simulating Organizations: computational models of institutions and groups,* eds. M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp. 169-189.

Kaplan, R. S. and Norton, D. P. (1996), *The Balanced Scorecard: From Strategy to Action*, Harward Business School Press, Harward.

Kido, T. (2001), "Internet Agent for Community Organization", International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, L'Aquila, Italy, August 06-12, 2001.

Kinny, D. and Georgeff, M. (1997), "Modelling and Design of Multi-agent systems", *Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages (ATAL-96)*, LNAI, Springer-Verlag, Vol. 1193.

Klein, M. and Dellarocas, C. (2000), "A Knowledge-Based Approach to Handling Exceptions in Workflow Systems", *Journal of Computer-Supproted Collaborative Work*, Special issue on Adapive Workflow Systems, Kluwer, Vol. 9, No. 3-4, pp. 399-412.

Kleppe, A., Warmer, J. and Bast, W. (2003), *MDA Explained: The model Driven Architecture: Practice and Promise,* Addison-Wesley.

Klusch, M.s (ed.) (1998), *Intelligent Information Agents,* Springer.

Lawrence, P. R. and Lorsch, J. W. (1967), *Organization and Environment: Managing Differentation and Integration,* Harvard Business School Press, Boston, USA.

Lee, J. H. M. and Zhao, L. (2002), "A Real-Time Agent Architecture: Design, Implementation and Evaluation", *LNCS,* Springer-Verlag, Vol. 2413, p. 18.

Levitt, R. E., Cohen, P. G., Kunz, J. C., Nass, C., Christiansen, T. and Jin, Y. (1994), "The Virtual Design Team: Simulating How Organizational Structure and Communication Tools Affect Team Performance", *Computational Organization Theory,* eds. K. M. Carley and M. J. Prietula. Hillsdale, N. J.: Lawrence Erlbaum Associates.

Lin, Z. (1998), "The Choice Between Accuracy and Errors. A Contingency Analysis of External Conditions and Organizational Decision Making Performance", *Simulating Organizations: computational models of institutions and groups,* eds M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp 67-87.

Maes, P. (1997), "General Tutorial on Software Agents".

Malone, T. W. et al. (1999), "Tools for inventing organisations: Toward a handbook of organisational processes" *Management Science*, Vol. 45, No. 3, pp. 425-443.

Meriste, M. and Motus, L. (2002), "On Models for Time-Sensitive Interactive Computing", *International Conference on Computational Science, Amsterdam*, April 21-24, 2002.

Meriste, M. and Penjam, J. (1995), "Attributed models of computing", *Proc. Estonian Academy of Sciences, series Engineering*, Vol. 1, No. 2, pp. 139-157.

MGHD (1984), *McGraw-Hill Dictionary of Scientific and Technical Terms,* Third Edition, ed in Chief S. P. Parker, McGraw-Hill Book Company, New York, St Louis, San Francisco.

MGHE (1987). *McGraw-Hill Encyclopedia of Science & Technology*, 6th Edition, McGraw-Hill Book Company, New York, St. Louis, San Francisco.

MIT SAG (2001), MIT software Agent Group. http://agents.media.mit.edu/. Accessed 13 February 2004.

MindTools (2003a), *Critical Path Analysis & PERT Charts*, http://www.mindtools.com/cripath.html, Accessed 7 July 2003.

MindTools (2003b), *SWOT Analysis*, http://www.mindtools.com/, Accessed 8 July 2003.

Montangero, C., Derniame, J.-C., Kaba A. and Warboys, B. (1999), "The software Process: Modelling and Technology", In: *Software Process: Principles, Methodology, Technology,* eds. J.-C. Derniame, A. Badara Kaba, and D. Wastell, Springer Verlag, pp. 1-13.

Moraitis, P., Petraki, E., Spanoudakis, N. (2002), "Engineering JADE Agents with the Gaia Methodology", *Agent Technologies, Infrastructures, Tools and Applications for e-Services*, eds. R. Kowalszyk, J. Müller, H. Tianfield, R. Unland, Lecture Notes in Computer Science, Springer-Verlag, Vol. 2592, pp. 77-91.

Mortensen, K. H. (2003), Coloured Petri Nets, web page, University of Aarhus, Denmark, http://www.daimi.au.dk/CPnets/intro/, Accessed 26 November 2003.

Mortensen, K. H. and Rölke, H. (2003), "Introductions to Petri Nets. Literature", Web page, University of Aarhus, Denmark, http://www.daimi.au.dk/PetriNets/introductions/, Accessed 26 November 2003.

Motus L. (2003), "Modeling metric time", *UML for real: Design of Embedded real-time systems*, eds. B. Selic, L. Lavagno, G. Martin, Kluwer Academic Publishers, Norwell, pp. 205-220.

Motus L., Meriste M., Kelder T., Helekivi, J. (2002), "An Architecture for a Multi-Agent System Test-Bed", IFAC Congress, Barcelona, 2002.

Motus, L. and Meriste, M. (2002), "Some remarks on time modelling in interactive computing systems", *Int. Conf. on advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine on the Internet, August 2002.*

Motus, L. and Meriste, M. (2001), "Towards Self-Organising Time-Sensitive Control Systems Software", *Proc., IFAC Conference on New Technologies in Computer Control,* November 2001, Hong Kong, pp 236-241.

Motus L. and Naks T. (1998), "Formal timing analysis of OMT designs using LIMITS", *Computer Systems Science and Engineering*, Vol. 13, No. 3, pp. 161-170.

Motus, L. and Rodd, M. G. (1994), *Timing Analysis of Real-Time Software*, Elsevier Science Ltd, Pergamon, Oxford, Great Britain.

Motus, L., Vingerhoeds, R. A. and Meriste, M. (2005), „Challenges for Real-time Systems Engineering: Part 2. Towards time-aware technology", *Proc. Estonian Academy of Sciences, Engineering*, Vol. 11, No. 1, pp. 19-30.

Mylopoulos, J., Kolp, M. and Castro, J. (2001), "UML for Agent-Oriented Software Development: The Tropos Proposal", *Proc. UML 2001 – The Unified modelling Language. Modelling Languages, Concepts and Tools: 4th International Conference, Toronto, Canada, October 1-5, 2001,* eds. M. Gogolla and C. Kobryn, LNAI, Springer-Verlag, Vol. 2185, pp. 422-441.

Naks T. and Motus L. (2001), "Handling Timing in a Time-critical Reasoning System – a case study", *Annual Reviews in Control*, Vol. 25, pp. 157-168

NetMBA (2003), PEST Analysis, http://www.netmba.com/strategy/pest/, Accessed 24 November 2003.

Newell, A. (1990), *Unified Theories of Cognition*, Harvard University Press, Cambridge, Massachusetts, USA.

Nguyen, T. (1997), "A Multi-Agent Architecture for Situation Awareness", *Proc. of First Int Conf on Knowledge-Based Intelligent Electronic Systems,* Mai 21-23, Adelaide, Australia, ed. L. C. Jain, IEEE, pp. 502-507.

Nikander, P. (1999), *An Architecture for Autjorization and Delegation in Distributed Object-Oriented Agent Systems*, PhD Dissertation, Helsinki University of Technology, ISBN 952-91-0786-2.

Nwana, H., Ndumu, D., Lee, L. and Collis, J. (1999), "ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems", *Applied Artificial Intelligence Journal*, Vol. 13, No. 1, pp. 129-186.

Odell, James J., Parunak, H. Van Dyke and Bauer, Bernhard (2000), "Representing Agent Interaction Protocols in UML", *First International Workshop, AOSE 2000 Limerick, Ireland, June 10, 2000. Revised Papers*, eds. P. Ciancarini and M. J. Wooldridge, LNCS, Springer-Verlag, Vol. 1957, pp. 121-140.

OMG (2002a), "Enhanced view on time", version 1.1, document formal/02/05/07.

OMG (2003a), OMG Unified Modeling Language Specification, Version 1.5, An Adopted Formal Specification of the Object Management Group, Inc, formal/03-03-01, March 2003, http://www.omg.org/docs/formal/03-03-01.pdf, Accessed 3 September 2003.

OMG (2003b), UML Profile for Schedulability, Performance and Time Specification, Draft Available Specification version 1.0 ptc/2003-03-02, http://www.omg.org/docs/ptc/03-03-02.pdf, Accessed 3 September 2003.

OMG (2003c), Business Process Definition Metamodel, Request for Proposal, bei/2003-01-06. http://www.omg.org/docs/bei/03-01-06.pdf, Accessed 3 September 2003.

OMG (2003d), UML 2.0 Infrastructure Final Adopted Specifcation, ptc/03-09-15, http://www.omg.org/cgi-bin/doc?ptc/03-09-15.pdf, Accessed 18 February 2005.

OMG (2005), Object Management Group, http://www.omg.org, Accessed 10 May 2005.
Ortega, D. and Soriano, A. (2001), "WUC: A System for Workflow Specification and Verification", *Proc of Conference on Information Systems Development,* Florida.
Parunak, H. van Dyke and Odell, James J. (2001), Representing Social Structures in UML. http://www.jamesodell.com/. Accessed 26 August 2003.
Prietula, M. J., Carley, K. M. and Gasser, L. (Eds) (1998), *Simulating Organizations: Computational Models for Institutions and Groups*, The AAAI Press / The MIT Press, Menlo Park, Massachusetts.
Quirk, W. and Gilbert, R. (1977). "The formal specification of the requirements of complex real-time system", AERE, Harwell, rep. no. 8602.
Raistrick, C., Francis, P. and Wright, J. (2004), *Model Driven Architecture with Executable UML*, Cambridge University Press.
Rakitin, S. R. (2001), *Software Verification and Validation for Practioners and Managers*, Seconn Edition, Artech House, Norwood Massachussetts.
Rao, A. S. and Georgeff, M. P. (1995), "BDI Agents: From Theory to Practice", *Proc. of the First Intl. Conference on Multiagent Systems*.
Rational (2003), IBM Rational Software homepage, http://www.rational.com/. Accessed 4 September, 2003.
Rencken, W. D. et al. (1993), "Autonomous Low-Cost mobile Robot for Complex Non-Production Environments", In: *Intelligent Autonomous Vehicles,* Proc. First IFAC International Conference, pp. 31-36.
Riley, G. (1999), "Clips – A Tool for Building Expert Systems", http://www.ghg.net/clips/, Accessed 28 August 1999.
Rognin, L. (1997), *Coordination Theory,* http://www-sv.cict.fr/cotcos/pjs/TheoreticalApproaches/Coordination/Coordinationpaper Rognin.htm, Accessed 18 August 2003.
Sacile, R., Montaldo, E., Coccoli, M., Paolucci, M. and Boccalatte, A. (2001), "Agent-based architectures for workflow management in manufacturing", *International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, L'Aquila, Italy, August 06-12, 2001, 5 pp.
Salmela, H. (1997), "From information systems quality to sustainable business quality", *Information and Software Technology*, Elsevier Science B.V., Vol. 39, pp. 819-825.
Sandoe, K. (1998), "Organizaional Mnemonics: Exploring the Role of Information Theory in Collective Remembering and Forgetting" *Simulating Organizations: computational models of institutions and groups,* eds. M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp. 191-213.
Savimaa, R. (2002a), "On modelling emerging behaviour of multifunctional non-profit organisations", *Information Systems Development, Advances in Methodologies, Components and Managements,* eds. M. Kirikova et al. Kluwer Academic / Plenum Publishers, New York, pp. 203-214.

Savimaa, R. (2002b), "Modelling of Organisations using agents", *Student talks and abstracts,* 7th Estonian Winter School in Computer Science (EWSCS), VII Eesti Arvutiteaduse Talvekool (EATTK), Palmse, Estonia March 3 - 8, 2002, http://greta.cs.ioc.ee/yik/schools/win2002/talks.php, Accessed 9 March 2004.
Savimaa, R. (2003a), *Registering of vehicle thefts and updating of related information in the police: an example of implementation of a methodology for modelling multifunctional organisations*, in Estonian, original title: *Sõidukivarguste registreerimine ja info kaasajastamine politseis: näide multifunktsionaalsete organisatsioonide modelleerimise metoodika rakendamisest*, Tallinn, not published, available at Estonian Police Board.
Savimaa, R. (2003b), Modelling of co-operation and personal interests in a hierarchical organisation, *Poster session,* 2nd Estonian Summer School on Computer and Systems Science (ESSCaSS'03), II Eesti Arvuti- ja Süsteemiteaduse Suvekool (EASTS'03), Taagepera Castle, August 10-14, 2003.
Savimaa, R. (2004a), "Integrating UML, the Q-model and a Multi-Agent Approach in Process and Behaviour Models of Organisations", *Proc The International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2004) jointly with The 10th International Conference on Information Systems Analysis and Synthesis (SAS 2004)*, July 21-25, 2004 Orlando, Florida, USA, Vol. I, pp. 167-171.
Savimaa, R. (2004b), "A methodology for Modelling of Modifications in Multifunctional Human Organisations", *Proc The 8th World Multi-Conference on Systemcs, Cybernetics and Informatics (SCI 2004)*, July 18-21, 2004 Orlando, Florida, USA, Vol. X, pp. 265-269.
Savimaa, R. (2004c), "Composition of Organisational Process Models for Supproting Information Systems Design", *Proc. 11th Doctoral Consortium on Advanced Information Systems Engineering (CaiSE 2004 DC)*, June 7-8, 2004 Riga, Latvia, pp. 59-70.
Savimaa, R. (2005), "Using agent and UML technologies in organisation modelling: a vehicle theft example", *Proc. Estonian Academy of Sciences, series Engineering*, Vol. 11, No. 1, pp. 31-45.
Scacchi, W. (1998), "Modeling, Simulating and Enacting Complex Organizational Processes. A Life Cycle Approach", *Simulating Organizations: computational models of institutions and groups,* eds. M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp. 153-167.
Schmidt, K. (1994), *The Organization of Cooperative Work. Beyond the 'Leviathan' conception of the organization of co-operative work CSCW'94*, Chapel Hill, North Carolina, October 24-26.
Schmidt, K. and Simone, C. (2000), *Mind the gap! Towards a unified view of CSCW. Proc. COOP2000,* Sophia, Antipolis, France, 23-26 May 2000.
Scott, R. W. (1992), *Organizations: Rational, Natural and Open Systems,* third edition, Prentice-Hall, Englewood Cliffs, N. J., USA.

SD (1978), *IEEE Standard Dictionary of Electrical and Electronic Terms,* Second Edition, ed in Chief F. Jay, An American National Standard, approved May 12, 1978.
SEI (2001), *People Capability Maturity Model (P-CMM) home page*, Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/cmm-p/version2/, Accessed 17 September 2003.
Selic, B. (2003), "Recent developments in model-driven development methods and the unified modelling language", lecture material in 2nd Estonian Summer School on Computer and Systems Science (ESSCaSS'03), Taagepera Castle, Estonia, August 10-14, 2003, http://greta.cs.ioc.ee/yik/schools/sum2003/, Accessed 3 September 2003.
Selic, B. and Motus, L. (2003), "Using models in real-time software design", *IEEE Control Systems Magazine*, Vol.23, No.3, pp. 31-42.
Shen, W. and Norrie, D. H. (1999), "Agent Based System for Intelligent Manufacturing: A State of the Art Survey", *Knowledge and Information systems,* Vol. 1, No. 2, pp. 129-156.
SNPB (1999), *Police Management System Seminar*, course material, Swedish National Police Board, Tallinn, October 1999, organised by Estonian Police Board and Swedish National Police Board.
So, Y. and Durfee, E. H. (1998), "Designing Organizations for Computational Agents", *Simulating Organizations: computational models of institutions and groups,* eds. by M. Prietula, K. Carley and L. Gasser, AAAI Press / The MIT Press, Menlo Park, California, USA, pp. 47-64.
SPC (1996), *Management action cycle, training and informative material*, The Scottish Police College, Tulliallan Castle, Kincardine, Scotland, UK.
Taveter, K. and Wagner, G. (2001), "Combining AOR Diagrams and Ross Business Rules' Diagrams for Enterprise Modelling", *Proc 2nd International Bi-conference Workshop on Agent-Oriented Information Systems,* Sweden / USA, Texas.
Taveter, K. (1999), "Business Rules' Approach to the Modelling, Design and Implementation of Agent-Oriented Information Systems", *Proc Of the First Int. Workshop on Agent-Oriented Information Systems,* 1 May 1999, Seattle, USA and 14-15 June 1999, Heidelberg, Germany.
The Omega Group (2001), *Agent Based Reasoning,* http://www.ags.uni-sb.de/~omega/research/agentsTP/, Accessed 18 August 2003.
Thompson, L. F. and Coovert, M. D. (2001), "From Doing to Delegating: Knowledge Workers and the Adoption of Intelligent Agent Technology", *Int Conf on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, L'Aquila, Italy, August 06-12, 2001, 12 pp.
Thrampoulidis, K. X., Goumopoulos, C. and Housos, E. (1997), "Rule handling in day-to-day resource management problem: an object-oriented approach", *Information and Software Technology*, Elsevier, Vol. 39, pp. 185-193.
Trist, E. L. (1981), "The Evolution of Sociotechnical Systems as a Conceptual Framework and as an Action Research Program", *Perspectives on*

Organization Design and Behavior, eds. A. H. Van de Ven and W. F. Joyce, John Viley, Viley-Interscience, New York, USA, pp. 19-75.

Van Aeken, F. and Demazeau, Y. (1998). *Minimal Multi-agent Systems.*

Wagner, G. (2002), "A UML Profile for External AOR Models", *Agent-Oriented Software Engineering III: Third International Workshop, AOSE 2002, Bologna, Italy, July 15, 2002. Revised Papers and Invited Contributions*, eds. F. Giunchiglia, J. Odell and G. Weiß, Lecture Notes in Computer Science, 2003, Springer-Verlag, Vol. 2585, pp. 138-149.

Wang, Huaimin and Wu, Quanyuan (2000), "A formal framework of multi-agent systems with requirement/service cooperative style", *Journal of Computer Science and Technology,* Science Press, Beijing, China and Allerton Press, Inc, New York, USA, 15, 2, March 2000, pp. 107-115.

Wang, J. (1998), *Timed Petri Nets, Theory and Application,* Kluwer Academic Publishers, Boston.

Wang, W. and Zhong, C. (2000), "The distributed workflow management system – Flow Agent", *Journal of Computer Science and Technology,* Science Press, Beijing, China and Allerton Press, Inc, New York, USA, Vol. 15, No. 4, pp. 376-382.

Wangler, B., Persson, A. and Söderström, E. (2001), "Enterprise Modeling for B2B Integration", *Int Conf on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, L'Aquila, Italy, August 06-12, 2001, 5 pp.

Wegner, P. (1998), "Interactive foundations of computing", *Theoretical Computer Science*, Vol. 192, pp. 315-351.

Wegner P. (1997), "Why Interaction is More Powerful than Algorithms", *Comm. of ACM,* Vol.40, No.5, pp.80-91.

Wegner P. and Eberbach E. (2004), "New models of computation", *The Computer Journal*, Vol. 47, No.1, pp. 4-9.

Wegner, P. and Goldin, D. (2000), "Coinductive models of finite computing agents", *Electronic Notes in Theoretical Computer Science*, Vol.19 (June 10, 2000); www.elsevier.nl/locate/entcs. Accessed 10 August 2002.

WfMC (1995), "The Workflow Reference model", *http://www.aiim.org/wfmc/mainframe.htm*. Accessed 21 November 2003. The Workflow Management Coalition Specification, issue 1.1

Wille, C., Dumke, R. R. and Stojanov, S. (2002), *Quality Assurance in Agent-Based Systems, Current State and Open Problems*. Otto-von-Guericke-Universität Magdeburg, Preprint Nr. 4, 2002.

Williams A. B. (2003), *A Hybrid Cognitive and Behaviour-Based Multi-Agent System for Simulated Robot Soccer*, Project 55:185. http://css.engineering.uiowa.edu/~agents/Assignments/Project1-SoccerBots_Project_Description.pdf, Accessed 18 August 2003.

Wood, M. F. and DeLoach, S. A. (2000), "An Overview of the Multiagent Systems Engineering Methodology", *Agent-Oriented Software Engineering: Proceedings of the First International Workshop on Agent-Oriented Software Engineering,* 10th June 2000, Limerick, Ireland, eds. P Ciancarini,

M. Wooldridge, Lecture Notes in Computer Science, Springer Verlag, Vol. 1957, pp. 207 – 221.

Wooldridge, M., Jennings, N. R. and Kinny, D. (2000), "The Gaia Methodology for Agent-Oriented Analysis and Design", *Autonomous Agents and Multi-Agent Systems*, Vol. 3, pp. 285-312.

Wooldridge, M. (2002), *An Introduction to Multi-Agent Systems*, John Wiley & Sons, West Sussex.

Xu, X., Ye, D. and Li, Q. (2000), "Dynamic Organsiation and mothodology for agile virtual enterprises", *Journal of Computer Science and Technology,* Science Press, Beijing, China and Allerton Press, Inc, New York, USA, Vol. 15, No. 4, pp. 368-375.

Yu, E. and Mylopoulos, J. (1993), "An Actor Dependency Model of Organizational Work – With Application to Business Process Reengineering", *Proceedings of the Conference on Organizational Computing Systems (COOCS 93),* November 1-4, 1993, Milpitas, California, USA, pp. 258-268.

Zimmermann, A. (2005), "Petri Net", web page, Technische Universität Berlin. http://pdv.cs.tu-berlin.de/~azi/petri.html, Accessed 10 June 2005.

Zhou, M. (1995), *Petri Nets in Flexible and Agile Automation,* Kluwer Academic Publisher.

Zou, T., Wang, J. and Zhang, F. (2000), "Information service model with mobile agent supported", *Journal of Computer Science and Technology,* Science Press, Beijing, China and Allerton Press, Inc, New York, USA, Vol. 15, No. 2, pp. 150-157.

ANNEX 1. CURRICULUM VITAE

1. Personal Data
   Name: Raul Savimaa
   Date and place of birth: February 12, 1970, Tallinn
   Citizenship: Estonian
   Marriage status: married
   Chindren: two children (age 5 and 3)

2. Contact Information
   Address: Ümera 36-12 Tallinn 13816
   Phone: 509 0797
   E-mail address: raul.savimaa@dcc.ttu.ee

3. Education

Institution	Year of graduation	Education (degree)
Tallinn Technical University	1995	Master of Science in Engineering (computer and systems engineering)
Tallinn Secondary School No 32	1988	Secondary Education

4. Language Skills

Estonian	Mother Language
English	Advanced
Russian	Advanced
Swedish	Basic

5. Special Courses

1994 February – July Lund Institute of Technology, research and master studies

6. Professional Empolyment

Since 1993	Estonian Police Board, Information systems department
1991-1992	National Library, Department of information systems, engineer
1987-1991	Information and Computation Centre of Ministry of Communication, Department of Control Systems, senior technician and engineer

7. Scientific Work
   - Doctorate studies and research on mission-critical process specification and organisation modelling
   - Publications:
   1) "On modelling emerging behaviour of multifunctional non-profit organisations." In *Information Systems Development, Advances in Methodologies, Components and Managements,* edited by M. Kirikova et al. Kluwer Academic / Plenum Publishers, New York, 2002, pp. 203-214.
   2) "Integrating UML, the Q-model and a Multi-Agent Approach in Process and Behaviour Models of Organisations", *Proc. The International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2004) jointly with The 10th International Conference on Information Systems Analysis and Synthesis (ISAS 2004)*, July 21-25, 2004 Orlando, Florida, USA, IIIS 2004, vol. I, pp. 167-171.
   3) "A methodology for Modelling of Modifications in Multifunctional Human Organisations", *Proc. The 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004),* July 18-21, 2004 Orlando, Florida, USA, IIIS, 2004, vol. X, pp. 265-269.
   4) "Composition of Organisational Process Models for Supporting Information Systems Design", *Proc. 11th Doctoral Consortium on Advanced Information Systems Engineering*, June 7-8, 2004 Riga, Latvia, NTNU 2004, pp. 59-70.
   5) "Using agent and UML technologies in organisation modelling: a vehicle theft example", *Proc. Estonian Academy of Sciences, series Engineering*, 2005, vol. 11, no. 1, pp. 31-45.

8. Defended theses
   MSc Thesis "Layered Control Architectures in Flexible Manufacture", Tallinn Technical University, 1995.

9. Main research directions
   Research on multi-agent systems and their components and modelling of behaviour of multifunctional organisations.

10. Other research projects
   Multi-agent systems, real-time systems, flexible manufacture.

04.05.2005

ANNEX 2. ELULOOKIRJELDUS

1. Isikuandmed
   Ees- ja perekonnanimi: Raul Savimaa
   Sünniaeg ja –koht: 12. veebruar 1970, Tallinn
   Kodakondsus: Eesti
   Perekonnaseis: abielus
   Lapsed: poeg (5 a.) ja tütar (3 a.)

2. Kontaktandmed
   Aadress: Ümera 36-12 Tallinn 13816
   Telefon: 509 0797
   E-posti aadress: raul.savimaa@dcc.ttu.ee

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	Lõpetamise aeg	Haridus (eriala/kraad)
Tallinna Tehnikaülikool	1995	Tehnikateaduste magister, arvuti- ja süsteemitehnika
Tallinna 32. Keskkool	1988	Keskharidus

4. Keelteoskus (alg-, kesk- või kõrgtase)

Eesti keel	emakeel
Inglise keel	kesktase
Vene keel	kesktase
Rootsi keel	algtase

5. Täiendõpe

   1994 veebruar – juuli Lundi Tehnikaülikool, magistriõpingud

6. Teenistuskäik

Alates 1993	Politseiamet, infosüsteemide osakond
1991-1992	Rahvusraamatukogu, infosüsteemide osakond, insener
1987-1991	Sideministeeriumi Vabariiklik Info- ja Arvutuskeskus, automatiseeritud juhtimissüsteemide osakond, vanemtehnik ja insener

7. Teadustegevus
   - Alates 01. märts 2005, Tallinna Tehnikaülikooli Automaatikainstituudi erakorraline teadur

- Doktoriõpe, ajakriitiliste protsesside spetsifitseerimine ja organisatsioonide modelleerimine
- Publikatsioonid:

1) "On modelling emerging behaviour of multifunctional non-profit organisations," *Information Systems Development, Advances in Methodologies, Components and Managements,* M. Kirikova et al. (eds.), Kluwer Academic / Plenum Publishers, New York, 2002, pp. 203-214.
2) "Integrating UML, the Q-model and a Multi-Agent Approach in Process and Behaviour Models of Organisations", *Proc. The International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2004) jointly with The 10th International Conference on Information Systems Analysis and Synthesis (ISAS 2004)*, July 21-25, 2004 Orlando, Florida, USA, IIIS, 2004, vol. I, pp. 167-171.
3) "A methodology for Modelling of Modifications in Multifunctional Human Organisations", *Proc. The 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004),* July 18-21, 2004 Orlando, Florida, USA, IIIS, 2004, vol. X, pp. 265-269.
4) "Composition of Organisational Process Models for Supporting Information Systems Design", *Proc. 11th Doctoral Consortium on Advanced Information Systems Engineering*, June 7-8, 2004 Riga, Latvia, NTNU 2004, pp. 59-70.
5) "Using agent and UML technologies in organisation modelling: a vehicle theft example", *Proc. Estonian Academy of Sciences, series Engineering*, 2005, vol. 11, no. 1, pp. 31-45.

8. Kaitstud lõputööd
   Magistritöö "Layered Control Architectures in Flexible Manufacture", Tallinna Tehnikaülikool, 1995.

9. Teadustöö põhisuunad
   Arukate komponentide (multiagentsüsteemid) ja nende käitumise uurimine, multifunktsionaalsete organisatsioonide käitumise modelleerimine

10. Teised uurimisprojektid
    Multiagentsüsteemid, reaalajasüsteemid, paindtootmine (flexible manufacture)

04.05.2005