Integrating UML, the Q-model and a Multi-Agent Approach in Process Specifications and Behavioural Models of Organisations

Raul SAVIMAA
Department of Computer Control,
Tallinn University of Technology
Ehitajate tee 5, 19086 Tallinn, Estonia

ABSTRACT

Efficient estimation and representation of an organisation's behaviour requires specification of business processes and modelling of actors' behaviour. Therefore the existing classical approaches that concentrate only on planned processes are not suitable and an approach that integrates process specifications with behavioural models of actors should be used instead. The present research indicates that a suitable approach should be based on interactive computing. This paper examines the integration of UML diagrams for process specifications, the Q-model specifications for modelling timing criteria of existing and planned processes and a multi-agent approach for simulating non-deterministic behaviour of human actors in an organisation. The corresponding original methodology is introduced and some of its applications as case studies are reviewed.

Keywords: Emergent Behaviour, Modelling of Organisations, Unified Modelling Language, Q-Model, Multi-Agent Systems and Simulation.

1. INTRODUCTION

Modelling of business processes in an organisation is important for prognosticating reliable results of the organisation's activities. Such results are, for example, correctness of processes, efficiency, fulfilment of goals, throughput and quality of service. Modelling is also an unavoidable step during the planning of changes in processes and in designing an information system that supports decision-making and continuity of business processes.

Modelling is often seen purely as a specification of existing or planned work processes. This is a simplified approach that cannot give reliable estimations on performance in time-sensitive systems in real organisations. The actual behaviour of an organisation and its human actors does not depend only on planned actions but also on emerging situations and reactions to those situations. Neither can behaviour be fully predicted since actors are persons with free will, motivations and intentions.

While behaviour is considered as a series of actions performed by an actor, emergent behaviour is considered here as an operational behaviour of an actor that emerges as a result of the actor's intentions, environmental situations and actor's reactions of situations. The expected emergent behaviour is actually a result of a number of different, concurrently and sometimes quite independently appearing factors and conditions.

In order to capture emergent behaviour of an organisation, specification of business processes should be extended by estimations of probable behaviour of different actors. As a result, efficient modelling of human organisations should consider both aspects of modelling – business processes and behaviour of human actors.

We can never describe expected emergent behaviour entirely (i.e. all possible behavioural patterns in all emerging situations). The behaviour of employees as well as that of a whole organisation is ongoing, determined by interactions, goals, existing knowledge and the ability to acquire new knowledge. Therefore modelling that meets the above-given requirements and considers time-dependent behaviour of actors cannot be based on the conventional algorithmic approach; such programs can be described and analysed by using interaction-centred models of computation [1].

Models of interactive computing [2] - and agent technology in particular - seem to be most suitable for this application. Characteristic to interactive computing is the phenomenon of emergent behaviour of the modelled system which is most similar to the actual behaviour in real human organisations [3]. It is generally accepted to regard behaviour in such circumstances as a set of interactive, repeatedly activated and terminating programs [3], rather than one non-terminating algorithm.

The current paper reviews suitable approaches for modelling planned processes along with verifying temporal requirements and simulating interactions of multiple actors. A unique modelling methodology is introduced that integrates both of the above-mentioned aspects in the modelling of organisations' behaviour.

The paper is organised as follows. Modelling of processes is reviewed in the next section. It is shown that for analysing and verifying timing criteria of processes, UML should be supported by a model processor. A suitable model processor – the Q-model is introduced and its implementation together with UML is illustrated.

A multi-agent representation of an organisation is examined in Section 3. It is shown that UML should be extended to enable a suitable description of an organisation and its actors. Section 4 presents how emergent behaviour of actors can be simulated by using the suggested methodology.

Section 5 integrates both aspects and gives an overview of a suggested original model of an organisation. Implementation examples are presented in Section 6 and conclusions are drawn in Section 7.

2. MODELLING OF PROCESSES AND TIMING CRITERIA: UML AND THE Q-MODEL

General approaches for process modelling

A number of methods are elaborated for the specification of processes in an organisation. Process modelling can be supported by a variety of tools and techniques. The choice of the methodology and tools to be used are often related to other activities in the organisation. For example, one or another methodology is either connected to quality management program or to information systems development process. Therefore the current approach does not prefer any specific approach and, as the main objective, only stresses the importance of using a systematic and generally accepted tool or method, if possible, or a combination of such tools and methods, if justified. For being more specific in the current paper, a widely used approach - the Enterprise Modelling [4] - is taken here as a basis for comparison.

Enterprise Modelling (EM) is a structured technique for describing major aspects of an enterprise [4]. According to this view an enterprise is an organisation, consisting of multiple interacting workflows. A workflow is a sequence of business processes dedicated to a certain result (e.g. production of a piece of hardware or performing of a request and answering to a query).

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, EM is a well-organised approach for modelling the current or future (desired) state in a large variety of organisations. The enterprise model consists of sub-models, such as goals model, business rule model, concepts model, business process model, actors and resources model and technical components and resources model [4].

However, in this paper the Enterprise Modelling approach is used as an illustration and is not unavoidably related to the introduced methodology. The EM is a general framework for modelling whereas the suggested approach concentrates on the modelling of processes in detail – what specific techniques and languages are used for the detailed modelling of processes

Specification of processes using UML

A suitable modelling language for describing processes in detail is the Unified Modelling Language UML [5]. It is a graphic language accepted by Object Management Group [6], suitable for specifying and documenting the artefacts of a software-intensive system [5]. It is widely used in industry, academic domains and by major software companies - mainly for modelling information systems. The UML is also one of the models that are able to encapsulate interaction-centred computations. UML describes different parts of the model in a single language. UML is not sufficiently formal for handling quantitative properties of behaviour. In some cases (e.g. formal verification or modelling very specific systems) specific languages are needed (e.g. the Q-model [7]). For that purpose the concept of model processors has been developed and a possibility to use specific languages is included in UML specifications. For instance, specific modelling of resources and time is included in UML profile for Performance, Schedulability and Time [8]. Conversion from UML to another language and back can be implemented semi-automatically, if such a tool will be elaborated. Currently the latest version of UML (2.0) supports modelling of organisational processes at the required level. Still a specialised modelling approach for specified and detailed analysis of temporal criteria of processes is useful.

Since UML 2.0 has not yet become a widely used standard and many computer-aided engineering software packages still support earlier versions of UML, the use if UML version 1.4 (as the ISO and industry standard [9]) is justified in the current approach. Differences between the standards do not affect the principal issues of the suggested approach.

UML is considered as the best suitable modelling notation for this approach because of its characteristics, extended availability and popularity. UML meets the requirements set for this approach: it allows modelling different aspects and perspectives of the system, it is logical and visual, therefore allowing also manual interpreting of diagrams. It has also several software tools that support sophisticated modelling (e.g. [10, 11]) or provide illustrative diagrams only. Since UML is widely used already for modelling information systems and related processes, it is recommended to use the same tools for multiple purposes, if possible.

In the introduced methodology the specification of business processes is done as follows. There is always a modelling task that determines what the key problems and areas are in a modelled organisation. Specification of processes starts with the general description of the organisation, presented in a natural language. This description consists of a goals model and a business rule model, as used in Enterprise Modelling [4]. The exact content and syntax of those models may vary, depending on the chosen technique and tool.

The methodology concentrates then on the key processes (that are related to the modelling and analysis task) and related actors. 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 Enterprise Modelling [4]. This specification of processes begins with the description of aims, tasks and essential processes actor by actor. When using some Enterprise modelling 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 of work processes (called *the process table*) in order to simplify the analysis and its later description in other notations (an example of the table is presented hereinafter in Section 4). Each process together with its characteristics is described in a row of the table. Both planned and real work processes of units and employees are described (stored in different sub-models, if necessary).

It is important to devise a detailed specification with explicitly determined relations between processes. Therefore different UML diagrams are integrated for a detailed specification of processes and their interconnections. Use case diagrams are applied to describe interactions of actors and activities. Activity diagrams are used for modelling activities and their sequence. Sequence and interaction diagrams can be used for a more detailed specification of interactions.

Use case and activity diagrams together with the *process table* should be well correlated during the whole modelling process, supporting thus the coherence of the description. For a more detailed specification, interaction diagrams and statecharts can be used. UML profile for Performance, Schedulability and Time allows to add timing tags to processes and activities (described in [12] in more detail). At the same time the UML profile itself cannot verify any models and suitable computerised tools should be looked for.

Verification of timing criteria

The Q-model [7] implements effectively the approach of interactive computing. The Q-model is one of view formalisms that handles time-sensitive interaction-centred computation (another is, for example, presented by Caspi and Halbwachs [13]). Therefore it can be used for modelling timing parameters of processes and interactions. Possible alternatives can be CSP or coloured or timed Petri Nets, but those alternatives cannot represent multiple time concepts in one system.

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. This corresponds to interaction-centred computations approach.

The Q-model is mostly used for process control environments for industrial domains e.g. chemical batch processes, water engineering, etc. The Q-model, although still not used, can also be efficiently used for describing time-sensitive processes in other domains and support a detailed analysis of the timing criteria of processes.

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 in some processes are executed in parallel in several copies or if the execution of some processes is suppresses. A timing diagram can be analysed by a human user in order to evaluate the results of process behaviour and detect collisions.

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 be also applied at the design and implementation stages. The Q-model in the current approach is used for a more precise modelling of timing characteristics of inter-process interaction.

The Q-model allows verification of timing criteria for processes and interactions. In this regard the Q-model can be seen as a RT UML model processor. UML models (a set of integrated use case, activity, interaction diagrams and statecharts with time tags) 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 since the Q-model forms a basis for the above-mentioned UML profile. Depending on which UML tool is used, there may be a need to convert the models manually since a corresponding computerised tool may not be available yet. For example, this kind of automatic conversion is in principle done for Artisan RT Studio [11], but not for Rational Rose RT.

According to the proposed methodology, modelling in the Q-model notation starts with elementary processes that are presented in the *process table*. As all real actions in the organisation can be successful or unsuccessful, it is also 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. 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. The Q-model demands a very detailed specification of process characteristics, forcing thus the designer to plan and analyse all related issues. In addition, the Q-model also supports hierarchical decomposition of processes.

In this approach the Q-model analysis is implemented and stored in Limits PC CASE Tool [14] projects. It enables the implementing of all necessary timing characteristics [15]. As a result of that, the application of the Q-model modelling tool often shows existing gaps in processes and interrelations between processes. Some misbehaviour can already be understood from process specifications. The main analysis concentrates on the analysis of timing diagrams.

The Q-model as the only modelling tool is not sufficient for multi-functional organisations. The model of real organisations in the actual 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.

3. MULTI-AGENT APPROACH

Multi-agent systems

The methods and tools considered so far in the present paper have focused on specifying the actual work routines and planned 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. At the same time 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 and 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. Multiple studies have been conducted on models for interactive computations in the agents' community, e.g. [16] and [17]. Agent-based system architectures are modular and redundant, agents can be easily reused and decision-making responsibilities can be distributed in modular way [18]. The principles of agent behaviour are quite similar to the principles governing human behaviour. Additionally an information system can be built to comprise software agents – computed systems to which one can delegate tasks [18].

There exist several systems designed for the modelling of MAS (e.g. reviews in [17]). Researches on using the UML in the modelling of multi-agent systems have been fruitful in recent years (e.g. [19], [20] and [21]) and an extension of the UML, called Agent UML, is being elaborated [22]. It is popular to follow FIPA [23] standards for modelling such agents that have to consider limitations of real world environments. At the same time there is the disadvantage that FIPA is still working on defining temporal constraints for agents [24]. Also the standards differ from UML profile for Schedulability, Performance and Time ([8]). Therefore the ideal approach in the future should integrate positive aspects of UML, AUML and FIPA standards. Promising steps towards it are first attempts to build agent-based systems with the latest UML version – UML 2.0 [25].

A socially intelligent agent, suitable to be used for the modelling of human organisations, should express human-style social intelligence, i.e.: be pro-active, achieve its objectives by choosing activities and interacting with other entities [26], balance individual and social concerns, operate in a resource-bounded manner, understand sophisticated communication, distinguish between partners and enemies [27], and be able to change behaviour according to dynamic changes in the environment [28].

Human organisation as a multi-agent system

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 [16]. 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. The influence of collective opinions and knowledge is important to consider if units have to co-operate as teams. This is easy to model since agent-based systems architectures offer a modular distribution of decision-making responsibilities [18].

To fulfil its tasks, organisations often perform a fixed, hierarchical predefined structure where 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. Organisations in MAS can be modelled as populations on interacting agents in a given structure [27]. 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 [29, 16].

Multiple roles of actors as well as formal and informal communication should be considered at different levels (level of individuals, groups and global societies) of an organisation and modelled in the MAS.

Modelling of motivation, co-operation, and communication

Each employee in an organisation is a person with free will and his/her own interests, beliefs, desires, habits. Motivations of employees influence the choice of a particular action when tasks are loosely prioritised or there are multiple concurrent tasks to choose from. The representation of motivations in multi-agent systems is quite similar to that of real organisations [16].

The motivations of agents can be expressed in class models of an agent (e.g. in some 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 used for an agent, as suggested, should be BDI-agent [30], the specific mapping of motivations that are desired to be expressed in 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 ideas of the designer.

Beliefs, desires, and other characteristics of human behaviour are more difficult to express in relation to pragmatic modelling of business processes. For that reason motivation is hereby considered as the only characteristic of the internal states of an agent (an actor).

Co-operation can be defined as the behaviour of an individual in performing his/her activities in the way that is useful for himself/herself and for a specific target group in order to produce common good and to fulfil necessary tasks. In a formal organisation co-operation means co-ordination of actions (i.e. actually performed work processes) and collaboration by sharing tasks and resources. An efficient way to increase contribution from each employee is to assign to persons the sub-goals of the organisation as their own goals [29]. Each individual faces the dilemma either to concentrate more on the satisfaction of his/her own interests or to contribute to the production of common good [31].

The key problem in implementing actual co-operation in MAS is co-ordination. Efficient co-ordination even improves the production of common good in a measurable way [32]. Different schemas can be used for co-ordination like co-ordination through partial global planning, co-ordination through joint intentions, co-ordination my 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.

Communication is a way for interaction between friendly, neutral or non-friendly agents or groups. Communication is often an essential method of 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 lead to choices in co-operation and may cause problems in fulfilling the expected timelines).

Communication in MAS is normally expressed by 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. FIPA [23] standards also define the suitable performatives for agent interactions. The FIPA agent communication language (ACL) is quite similar to other agent communication languages like KIF or KQML [16], e.g. the structure of the messages is the same and the message attribute fields are also very similar. The comparison can be found in [17]. 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 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). 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. Considering the fact that the modelling objective in the current approach is to model the behaviour of a human organisation, FIPA performatives meet this aim. The FIPA agent communication language is hereby chosen as the most suitable one.

4. REPRESENTATION AND SIMULATION OF PROACTIVE COMPONENTS

Modelling of actors and their behaviour

Implementation of the introduced methodology continues with modelling the organisation form actors' viewpoints. The purpose of this step is to capture emergent behaviour of actors and the organisation, therefore actors' choices and interactions between different actors are analysed and simulated. As described in the previous section, the whole organisation is seen as a multi-agent system (MAS). It is justified to view an organisation as MAS since there are many acting entities with their own objectives and capabilities inside an organisation. The behaviour of selected key actors related mainly to goal-achieving (e.g. choosing from multiple tasks, information capturing, processing and exchange) and interactions between actors are described as the behaviour of collaborating agents in MAS.

In general, all related 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 previously in use case diagrams should be specified as agents. Additionally, the described artificial actors may improve the completeness and trustworthiness of the model (i.e. used for representing all the most important interactions). In reality it is too voluminous to describe the whole system (organisation) and a suitable set of agents and their characteristics can be chosen for each concrete case only, depending on the aim of the specific modelling (e.g. analysis of information exchange or co-operative activities).

Multi-agent representation of the organisation starts with determination and specification of agent classes. Agent class diagrams are developed from UML use case diagrams. The developing of agent class diagrams starts from the determination of agent classes. The modelling task determines what structural units or aspects (i.e. views on the behaviour and performance) of the organisation are to be modelled or focused on more deeply.

The agent class model should be able to represent both aspects – organisational and human subjective aspect. Each agent applies methods that can be seen as work processes of the actor that the agent represents. A separate class is created for every actor possessing different characteristics from others. An agent is thus an instantiation of this agent class and corresponds to the respective actor in the class model.

Work methods and routines of actors 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 to distinguish different, more specific tasks and roles. 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 endure 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 such cases it can 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 of broad re-structuring of the organisation.

The introduced 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 in the present methodology is seen in inheriting some of the agent properties from an upper class that represents human characteristics (e.g. the class Person). The inheritance of human characteristics can be also implemented hierarchically. In the current research a specific set of characteristics is chosen, some of which are more important only for this modelling approach (e.g. attributes like preferences, priority tasks and the list of actor's own priorities or methods, like selecting of an activity).

Although actual behaviour of an actor depends on different personal characteristics like motives, intentions, interests (e.g. career interests), ethical and moral norms (versus the goals of the organisation) or sympathies and antipathies between employees, it is very difficult to model these characteristics in relation to work processes. Therefore the current methodology considers only issues of co-operation, competition of different tasks with similar priority level and selecting of activities if personal priorities do not correspond to official priorities. These personal characteristics are easy to include into agents' methods and behaviours as additional internal modules for delays or selecting activities according to dynamic priorities.

As a result, according to the currently suggested methodology, the agent class methods for each agent that represents a human actor in the system constitute usually an individual combination of properties inherited from two hierarchies of classes – one for work methods and another for personal characteristics. In this way it is possible to simulate decision-making in an actor that takes into account organisational goals and personal interests.

Internal description of an agent (e.g. its methods) can be presented in UML and analysed again in the Q-model [33]. Use case, activity and sequence diagrams are used for specifying the dynamic behaviour of agents.

The exact 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. For a more exact simulation of human behaviour BDI (belief-desire-intention) architecture would be used (e.g. [34]). Otherwise cognitive agents with teleonomic behaviour (for classification see [16]) are most suitable for modelling [27]. If models are simplified and only determined work processes are modelled, internal structure of an agent and its behavioural description can be simplified and other types of behaviour used. For example, the architecture used for the illustration in the current paper is combined with competitive task architecture (the behaviour is chosen on the basis of the message received) and modular horizontal architecture. In general, an agent should meet the FIPA standards and recommendations as well as correspond to the recommended generic structure of an agent [35].

Multi-agent simulation

In addition to agent models, a subset of the system may be implemented as executable agent simulation. This is seen as a complementary (additional) measure for allowing analysis and dynamic visualisation of important parts of the systems behaviour, thus helping both designers and system owners to understand the system better.

Simulation model is a simplified model of the organisational model. The simulation model serves for devising a multi-agent simulation that is the most useful one for representing the key aspects of emergent behaviour from multiple actors' viewpoint. The complete simulation model consists of simplified UML use case, activity, sequence and class diagrams. Actually implemented agents in the simulation may act in a simpler way than actors described in the model and the simulation may cover only a part of the model (for example some specific interactions or information exchange).

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 – as one agent or as specific routines in modelled agents.

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 with 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.

Currently (as it is used so far) an agent meets the FIPA standards and recommendations. Sequence and interaction diagrams are used to model the application aspects of inter-agent communication protocols.

Agent modelling software tool JADE [36] has been selected as a starting point for agents' development in agent simulation. Communication (message exchange) between agents in JADE is FIPA compliant. Cooperation, any kind of control and storing of data in the log file results from inter-agent communication. JADE is not the only available environment for developing agents for simulation model. Other possible environments can be, for instance, AgentTool [37] and Zeus [38].

5. INTEGRATION: THE MODEL OF AN ORGANISATION

The introduced modelling methodology describes a model of the organisation that will be composed as a result of the specification of business processes (described in Section 2 of the current paper) and simulation of actors in a multi-agent system (described in Section 3). As a result, the model describes the estimated dynamic behaviour of the organisation from four viewpoints:

1) Planned business processes are described by use case and activity diagrams in UML. Use cases describe static relationships between processes and actors, activity diagrams are used to express dynamic relationships between processes.
2) Timing properties of processes and their interactions are described (and verified) by using the Q-model. The process specifications for the Q-model are derived from the *process table* and UML use case and activity diagrams.
3) Principally non-deterministic behaviour of employees as actors is described by representing the organisation as a multi-agent system. In this system human actors (and important artificial actors) are described as agents. The agents are described by using UML class diagrams. The behaviour of agents is described by UML sequence and interaction diagrams.
4) A selected part of the system, its behaviour and interactions between agents are simulated as a multi-agent simulation, developed in JADE. The UML use case, activity and sequence diagrams, used for the specification of the simulation are devised on the basis of UML diagrams representing the whole modelled organisation and the Q-model process diagrams.

The model of the organisation is a semiformal description of the organisation, corresponding to that organisation at the moment of modelling. The model allows:

1) Analysis of processes and their compatibility with the goals set for different actors, described in UML and in common language.
2) Verification and simulation of timing parameters of processes and their interactions (in the Q-model).
3) Consideration of multiple roles and tasks of actors as well as usage of common resources (in UML).
4) Visual simulation of probable behaviour and interactions in different circumstances (in agent simulation model).

Model analysis and organisation analysis that are derived from it constitute a deep, complex and time-consuming process. Its outcome is a common understanding of organisational processes and a reliable model of the organisation that can describe the organisation's reaction to dynamic inputs from the environment.

As no unified simulation environment exists currently for UML, the Q-model and agents, the model actually consists of four different components:

1) A document that comprises a general description of the organisation together with its structural model, the table of work processes and a description of processes and behaviour in the form of various diagrams and explanations.
2) UML model (that actually includes use case, activity, class and interaction diagrams).

3) The process model in the Q-model simulation environment for analysing and simulating timing properties and process interactions.
4) A multi-agent simulation model.

The validation of the model of the organisation is checking the correspondence of the model to the modelled organisation (or its component) i.e. checking whether the model exhibits the important organisational characteristics correctly and represents the behaviour of the organisation adequately. The validation can be later confirmed or rejected by experiments (simulation).

The verification of the model is formal proving that the model represents the organisation. Verification of different parts of the current model and ensuring their correspondence is quite difficult and almost possible. Also there does not exist sound theory for verification of actors' behaviour in multi-agent systems.

6. IMPLEMENTATION

The described approach as a modelling methodology was first implemented in the Estonian police in 2003 for the analysis of information exchange and the efficiency of existing information systems in regard to stolen vehicles. The modelling task concentrated on the police activities in registering and exchanging information in cases of vehicle theft. In that example the aim of the modelling task was also to give recommendations for modifying information systems and relevant work processes. Related work regulations of three police officials (a duty officer in local command and control centre, a duty officer in the central command and control centre, and a patrolling officer) and two officials in the vehicle registration office and in a border crossing point were studied and modelled by using UML use case and activity diagrams and timing criteria in the Q-model Limits PC CASE Tool [14]. Part of the related behaviour was also modelled by using agent simulation in JADE. The multi-agent system (MAS) involved 12 agents. The presented recommendations were supported by illustrations and experiments based on the model.

It was suggested that the implementation of given proposals should cause only one-time modification and there should be no need to increase manpower in everyday performance. The recommendations given in the analysis document were implemented in the police organisation.

Excellent characteristics of an emergent organisation and an existing need to model complex tasks in the particular organisation were the main reasons for choosing the police as a modelling example. The handling of similar issues in other researches (e.g. [39]) gives a good opportunity for comparing the results.

In that case study the behaviour of the systems was modelled without personal interests and depends directly on the messages received: e.g. each input message received by an agent causes specific response (a message or activity). The behaviour of the system under such circumstances is still emerging since some events happen on a random basis. 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.

That case study was later in 2004 extended with modelling of motivation and co-operation of some key actors. The aim of the extended case study was to proceed from the initial task, given by the organisation, 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 2005 the methodology was used for case study on task decomposition and co-operation of structural units and for analysing of activities of a project team. In all case studies the intention was to follow the methodology as much as possible.

Those case studies indicated the implementability of the suggested methodology also for other modelling tasks. It still can be concluded that the methodology is much suitable for modelling emergent behaviour in time-sensitive environments where the benefit of time-consuming Q-model and MAS analysis is the biggest.

7. CONCLUSIONS

In this paper an original approach is presented on integrating the modelling of business processes and actors' behaviour in an organisation. The methodology combines UML and the Q-model for modelling processes and timing criteria and a multi-agent approach for representing actors' behaviour, choices and interactions.

The main conclusion of the research is: time-sensitive emergent operational behaviour of a multi-functional organisation can be efficiently modelled. A suitable solution is simultaneous modelling from the viewpoints of processes and actors, using a model that combines UML, the Q-model and agent technologies.

Two additional conclusions can be drawn. First, since the Q-model corresponds to UML profile of Schedulability, Performance and Time and a suitable computerised tool exists for the analysis and verification of timing criteria of process models, the Q-model can be used as a model processor for analysing and verifying timing criteria and interactions of business processes in an organisation. It can likewise be applied for analysing possible interactions of a multi-agent system that represents actors in this organisation. Moreover, it means that the same tool can be used for the analysis of both planned processes and agents' behaviour.

Second, an extension of UML that could be fully suitable for modelling organisations according to the aforesaid principle should integrate timing and performance criteria along with agent modelling; therefore a suitable integration could be UML 2.0 with Agent UML.

The suggestions made regarding the modelling emergent behaviour are not final. Further work, related to the proposed methodology, would comprise the following activities.

First, 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.

Second, devising 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. The elaboration is related to the development of agent environments.

ACKNOWLEDGEMENT

This research is partially financed by the Estonian Science Foundation grant ETF 4860 and the Estonian Ministry of Education grant 0142509s03.

REFERENCES

[1] P. Wegner, "Why Interaction is More Powerful than Algorithms", **Communications of ACM**, Vol. 40, No. 5, 1997, pp.80-91.
[2] P. Wegner, "Interactive Foundations of Computing", **Theoretical Computer Science**, Vol. 192, 1998, pp. 315-351.
[3] M. Meriste and L. Motus, "On Models for Time-Sensitive Interactive Computing", **Lecture Notes in Computer Science**, No. 2329, 2002, pp. 156-165.
[4] J.A. Bubenko jr, A. Persson and J. Stirna, **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, projext No IST-2000-28401, Appendix B: EKD User Guide, Stockholm: Royal institute of Technology, 2001.
[5] G. Booch, J. Rumbaugh and I. Jacobson, **The Unified Modelling Language. User Guide**, Reading, Massachusetts: Addison Wesley Longman, Inc., 1999.
[6] Object Management Group, http://www.omg.org. Accessed 7 February 2006.
[7] L. Motus and M.G. Rodd, **Timing Analysis of Real-Time Software**, Oxford: Pergamon, Elsevier Science Ltd, 1994.
[8] OMG, **UML^{TM} Profile for Schedulability, Performance, and Time Specification**, v.1.1, http://www.omg.org/cgi-bin/doc?formal/2005-01-02. Accessed 7 February 2006.
[9] International Organization for Standardization, ISO/IEC 19501:2005 Information technology – Open Distributed Processing – Unified modelling Language (UML) Version 1.4.2, http://www.iso.org. Accessed 7 February 2006.
[10] IBM Rational Software, **Rational Rose**, IBM Rational Software homepage. http://www-306.ibm.com/software/rational/. Accessed 10 February 2006.
[11] ARTiSAN Software Tools, Inc., **Artisan Real-time Studio Professional**, http://www.artisansw.com. Accessed 10 February 2006.
[12] B. Selic and L. Motus, "Modelling of Real-Time Software with UML", **IEEE Control Systems Magazine**, Vol. 23, No. 3, 2003, pp. 31-42.
[13] P. Caspy and N. Halbwachs, "A Functional Model for Describing and Reasoning About Time Behaviour of Computing Systems", **Acta Informatica**, Vol. 22, Nr. 6, Springer-Verlag NY Inc, 1986, pp. 595-627.
[14] **Limits-PC User Manual, version 1.1**, Institute of Automation, Tallinn Technical University, Tallinn, 1998.
[15] L. Motus and T. Naks, "Formal Timing Analysis of OMT Designs using LIMITS", **Computer Systems Science and Engineering**, Vol. 13, No. 3, 1998, pp. 161-170.
[16] J. Ferber, **Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence**, Harlow, London: Addison-Wesley, 1999.
[17] M.J. Wooldridge, **An Introduction to Multi-Agent Systems**, West Sussex, UK: John Wiley & Sons Ltd, Chichester, 2002.
[18] K.S. Barber, A. Goel, D. Han, J. Kim, T.H. Liu, C.E. Martin and R. McKay, "Problem-Solving Frameworks for Sensible Agents in an Electronic Market", Proc. International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, **Lecture Notes in Computer Science**, Vol. 1611, 2004, pp. 470-479.
[19] S. Flake, C. Geiger and J.M. Küster, "Towards UML-based Analysis and Design of Multi-Agent Systems", **Proc of International NAISO Congress Information Science Innovations**, Dubai: Naiso Academic Press, 2001, pp.695-701.
[20] F. Bergenti and A. Poggi, "Exploiting UML in the Design of Multi-agent Systems", Engineering Societes in the Agents World: First International workshop ESAW 2000, **Lecture Notes in Computer Science**, Vol. 1972, 2000, pp. 106-113.
[21] B. Bauer and J.P. Müller, "Using UML in the Context of Agent-Oriented Software Engineering: State of the Art", **Agent-Oriented Software Engineering IV**, edited by G. Giorgioni, J.P. Müller and J. Odell, **Lecture Notes in Computer Science**, Springer-Verlag, 2003, Vol. 2935, pp. 1-24.
[22] B. Bauer, J.P. Mueller and J. Odell, "An Extension of UML by Protocols of Multiagent Interaction", **Proc. Fourth Int. Conference on Multi-Agent Systems**, Boston, Massachusetts, USA, 2000, pp. 207-214.
[23] Federation of Intelligent Physical Agents. http://www.fipa.org. Accessed 8 February 2006.
[24] FIPA, **FIPA Modelling Area: Temporal Constraints**, Working Draft, version 1.0, 2003-04-18, http://www.auml.org/auml/documents/Temporal-Constraints.pdf. Accessed 8 February 2006.
[25] B. Bauer and J. Odell, "UML 2.0 and Agents: How to Build Agent-Based Systems With the New UML Standard", **Engineering Applications of Artificial Intelligence**, Vol. 18, 2005, pp. 141-157.
[26] L.M.J. Hogg and N.R. Jennings, "Socially intelligent Reasoning for Autonomous Agents", **IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans**, Vol. 31, No. 5, 2001, pp. 381-393.
[27] B. Edmonds, "Modelling Socially Intelligent Agents", **Applied Artificial Intelligence**, Vol. 12, No.7-8, 1998, pp. 677-699.
[28] M. Armin and D. Ballard, "Defining New Markets for Intelligent Agents", **IT Professional**, Vol. 2, No. 4, 2000, pp. 29-35.
[29] M.S. Fox, M. Barbuceanu, M. Gruninger and J. Lin, "An Organizational Ontology for Enterprise Modeling", **Simulating Organizations: computational models of institutions and groups**, edited by M. Prietula, K. Carley

and L. Gasser, Menlo Park, California: AAAI Press / The MIT Press, 1998, pp 131-152.

[30] A.S Rao and M.P. Georgeff, **"**BDI Agents: From Theory to Practice." **Proc. of the First Intl. Conference on Multiagent Systems,** 1995.

[31] M.C. Kang, L.B. Waisel and W.A. Wallace, "Team Soar: A Model for Team Decision Making", **Simulating Organizations: computational models of institutions and groups**, edited by M. Prietula, K. Carley and L. Gasser, Menlo Park, California: AAAI Press / The MIT Press, 1998, pp 23-45.

[32] B.A. Huberman and N.S. Glance, "Fluctuating Efforts and Sustainable Cooperation", **Simulating Organizations: computational models of institutions and groups**, edited by M. Prietula, K. Carley and L. Gasser, Menlo Park, California: AAAI Press / The MIT Press, 1998, pp 89-103.

[33] R. Savimaa, "On Modelling Emerging Behaviour of Multifunctional Non-Profit Organisations", **Information Systems Development, Advances in Methodologies, Components, and Managements**, edited by M. Kirikova et al., New York: Kluwer Academic / Plenum Publishers, 2002, pp. 203-214.

[34] D. Kinny and M. Georgeff, "Modelling and Design of Multi-agent systems", **Intelligent Agents III: Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), Lecture Notes in Artificial Intelligence**, Vol. 1193, 1996.

[35] L. Motus, M. Meriste, T. Kelder and J. Helekivi, "An Architecture for a Multi-Agent System Test-Bed", **Proc. Of the 15th IFAC World Congress**, Vol. L, Elsevier Science Publ., 2002, 6pp.

[36] F. Bellifemine, A. Poggi and G. Rimassa, "JADE – A FIPA-Compliant Agent Framework", **Proc of 4th Int. Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents PAAM'99**, London, 1999, pp. 97-108.

[37] S.A. Deloach and M. Wood, "Developing Multiagent Systems with agentTool", **Lecture Notes in Computer Science,** Vol. 1986, 2000, pp. 46-60.

[38] H. Nwana, D. Ndumu, L. Lee and J. Collins, "ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems", **Applied Artificial Intelligence Journal**, Vol. 13, No. 1, 1999, pp. 129-186.

[39] J. Dugdale, B. Pavard, J.L.Soubie, "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)**, Amsterdam: IOS Press, Vol. 58, 2000.