A Flexible Agent-Based Framework for Process Management

Gregor Joeris, Christoph Klauck, Holger Wache

University of Bremen
FB 3, TZI / AG KI
P.O. Box 330 440
D-28334 Bremen
Germany
joeris|klauck|wache@informatik.uni-bremen.de

ABSTRACT

Information technology support for complex, dynamic, and distributed business processes as they occur in engineering domains requires an advanced process management system which enhances currently available workflow management services with respect to integration, flexibility, and adaptation. We present an uniform and flexible framework for advanced process management on an abstract level which uses and adapts agent technology from distributed artificial intelligence for both modelling and enacting of processes. We identify two different frameworks for applying agent technology to process management: First, as a multi-agent system with the domain of process management. Second, as a key infrastructure technology for building a process management system. We will then follow the latter approach and introduce different agent types for managing activities, products, and resources which capture specific views on the process.

1. Introduction

One of the innovative developments in enterprises over the past years was the introduction of computer assisted handling procedures in various problem groups. This innovation step makes it easier for the individual enterprises to respond to the market's requirements for shorter delivery times, growing product complexity in customised varieties - e.g. one of a kind manufacturing - and highest quality. The growing complexity of the ways of looking at problems in enterprises, e.g. due to an increasing complexity of the product or distributed manufacturing within a consortium, implicates a more complex business process and thereby a growing demand for the necessary software. Problem-oriented, customer-specific software solutions, which can be adapted to new, similar problems without incurring greater costs and integrated easily into a distributed and dynamic business process, are required. Within our research in the project MOKASSIN we attempt to come closer to this goal using a flexible agent-based framework as described below.

A process-oriented view has become established to cope with the above mentioned challenges. The principal idea is to explicitly grasp all process-related information, i.e. activities, artefacts, resources, and organisational units as well as their interdependencies, and to describe them in a model. This makes it possible to deal methodically with the single processes. By analysing and evaluating these processes, reengineering and continuous improvement of the processes will be achieved. All this is summarised by business process (re-)engineering (cf. [HaCh93]). Moreover, the formal description is the basis for the execution of the process specifications and thus for the direct support of the actors who perform a specific task. This enactment support is provided by workflow management systems (WFMS) (cf. [JaBu96]). However, the application of WFMSs is currently restricted to well structured and a priori defined processes. Neither a participated and distributed modelling with integration of the process models in a logical, overall view is possible nor mechanisms for the description of engineering and management processes are available (cf. [GHS95], [Amb96]). Engineering processes have a very dynamic nature which means that they cannot be planned in advance and are under change during execution. Thus, advanced process management technology is needed to support complex, dynamic and distributed processes. To cope with these challenges, we propose an agent-based approach to process management.

In the following sections we first describe a real world scenario in section 2 in order to motivate the requirements of an advanced process management system, as described in section 3. In section 4, we then explain how agent technology may be applied to process management, and identify two different approaches: First, as a multi-agent system with the domain of process management. Second, as a key infrastructure technology for building a process management system. We will then follow the latter approach and introduce in section 5 a flexible framework for modelling and enacting of processes which is based on five agent types and interrelationships between them. Furthermore, we explain why we see agents as a key infrastructure technology which best meets the requirements of advanced process management. Since there is no generally accepted definition of an agent, we outline the main characteristics of our agents and relate them to different agent definitions and to distributed object management. Section 6 gives the conclusion to our paper and describes forthcoming research.

2. Scenario

The real world scenario described in this section is coming up from acquisition activities within our project MOKASSIN, where we analyse today's situation of distributed and dynamic business processes in 10 enterprises. From these scenarios we derive the requirements for our flexible agent-based framework, as described in section 3.

In our example two enterprises - enterprise D (3500 employees) and M (25 employees) - are involved. The larger's one product is manufactured individually for every customer. This is the situation e.g. in shipyard industry, space industry and software-engineering. The common situation in these industries is, that manufacturing is started before the detailed design is finished. One manager of enterprise D summarises the situation by the statement: "Our business is changes!".

The business process of D is document centred, i.e., changes in the documents describing the product and the state of manufacturing etc. trigger new activities. As a result, the business process is not a priori determined, i.e., the business process evolves dynamically.

In the cooperation between D and M, this situation becomes more difficult, the main problem being M's business process. Because of security reasons, these are not known to D, i.e. an overall integrated model is not possible. Only the interface process describing the cooperation between D and M is accessible by both.

If any change should arise in D's process affecting M, e.g. regarding material and measurements, M will have to be informed as soon as possible. But, what kind of information will be sent to M? And how will it integrated into their running process? A helpful system in this respect would be a system determining which of the changes are important with regard to M. These would then be sent electronically to M and integrated into their product data, thus updating M's process by informing the respective persons or systems.

3. Advanced Process Management

With process modelling the knowledge of processes becomes explicit. This is the foundation for the analysis and improvement of processes in business process reengineering as well as for enactment support provided by workflow management systems ([JaBu96]) or process-centred software engineering environments ([FKN94]). WFMSs are currently restricted to support well-structured processes which have been designed and improved by business process (re)engineering activities. Furthermore, the design of a business process, its formalization to a workflow specification and the execution of this specification are totally separated leading to a prescriptive and inflexible work environment. In the following, we use the notion (advanced) process management rather than workflow management as a general term

a) to denote an integrated approach for modelling and enacting of processes which includes (1) analysis, simulation, assessment, and improvement of processes, (2) enactment support like guidance, enforcement, and automation ([DoFe94]), (3) monitoring, measurement, and traceability;
b) to emphasize support for a wide range of processes - from low-structured and dynamic processes (e.g. engineering processes) to highly automated business transactions (e.g. transactional workflows [GHS95]);
c) and to avoid the restricted focus of application which is usually associated with workflow management, but to point out a comprehensive support for disparate processes and collaborative work.

Throughout this paper, we concentrate on the core representational and functional requirements of process management, namely modelling and enacting of processes, and disregard outer core functionality like analysis, improvement, monitoring, or measurement. The following list outlines the main modelling and enacting requirements for an advanced process management:

Modelling:

1. *Support for different process modelling scenarios*: Both, centralised as well as participatory modelling of processes should be supported (cf. [Amb96]). The latter requires mechanisms to integrate different process model fragments into a global view of the process and to specify global constraints and business rules which ensure the definition of valid process descriptions.

2. *Customisation of processes*: Process descriptions should be adaptable to personal/project's needs. Furthermore, 'what is to be done' should be separated from 'how has it to be done'. Then, the actor may decide which process is best to realise the task in a specific situation.
3. *Modelling and integration of enterprise-wide processes*: The capabilities for distributed modelling and integrating of process models have to be enhanced and adopted to be applicable for enterprise overlapping processes.
4. *Reflexivity*: It should be possible to model and enact the process of process management like any other process (cf. [CFF93]).

Enacting:

1. *Process evolution*: In order to support dynamic and low-structured processes (e.g. like engineering processes), mechanisms for flexible planning and modification of processes have to be provided to reflect the real world processes. This requires techniques for scheme evolution, on-the-fly modifications of enacting process instances, and exception handling (cf. [CFF93], [BDF94]).
2. *Mechanisms to flexibly control the flow and assignment of work*: This should include actor-driven, document-based and process-based concepts and in particular mechanisms for handling product changes and feedbacks in the process (cf. [HJKW96]).
3. *Transactional support*: In order to co-ordinate concurrent execution of both manual and automatic tasks and to support automation of process steps, relaxed and adaptable transaction protocols have to be provided and integrated with the execution behaviour specification (cf. [GHS95])
4. *Integration of version and work space control capabilities* into the process management system to support versioning, change propagation and co-operative work. This requirement is particularly important for engineering processes.

Furthermore, we focus on the system architecture of a process management system and on the integration of heterogeneous application systems. The main requirements are as follows:

1. *Flexible architecture*: The architecture of the process management system should be scaleable and extensible and has to support physically distributed, intra- and inter-enterprise process execution.
2. *Security for data interchange*: Process relevant data and product data interchange between enterprises require secure communication and authentication protocols.
3. *Interoperability with application systems*: System integration should be realised at least on the basis of control integration. Especially, transaction management capabilities of the application system have to be integrated into the transaction protocol of the process engine.
4. *Data Integration*: Tight integration of application systems has to take the integration of the application system data into account. This requires an uniform way to access and interchange data, mechanisms for syntactic and semantic data transformation, and management of data dependencies.

To sum up, integration as well as flexibility and adaptation are the recurrent problems and the main challenges in process management: A uniform process-centred framework is required for (1) integrating workflows, products, resources, and organisational units, for (2) integrating processes of different projects and enterprises, for (3) integrating manual and automatic processes, for (4) integrating heterogeneous systems, and for (4) integrating user-specific adaptation and deviations of processes into one global logical model. Flexibility is needed for both, the process model and the virtual work environment in which processes are performed, to handle with any kind of changing situations.

4. Agents meet Process Management

Agent-based computing is a emerging field in software development. But, there is no clear origin of the concept agent and no agreement of what an agent is. The general usage of the notion 'agent' may be distinguished respectively into a weak and strong one [WoJe95]. We prefer the weak and relatively uncontentious notion of an agent which is defined as a hardware or software system that exhibits the properties of autonomy, reactivity, pro-activeness, and social ability. Furthermore, our view on (software) agents is based on the concurrent Actor model of Hewitt [Hew77] which may be seen as one origin of the concept of an agent [Nwa96]. Hewitt defines an actor as a self-contained, interactive and concurrently-executing object which encapsulates some internal state and may interact with other actors by message-passing. Moreover, the usage of an agent as a generic term may be refined by more accurate definitions for specific agent types (cf. [Nwa96]). When introducing our agent-based framework for process management, we will identify various agent types for process management and discuss their characteristics in more detail.

To clarify the objectives of our framework, we first explain how agents can be applied to process management. In general, there are two different ways to use agent technology for this purpose:[1]

(1) First, as a multi-agent system with the domain of process management: Different agent types are built which perform different types of tasks and therefore fulfill different *roles* (cf. [Kir96]). Thus, there will be specialised agents e.g. for risk analysis in the field of claim processing which perform or assist in performing a specific task. Based on an agent architecture designed for managing business processes, agencies have to be designed for and adapted to different application areas where the designed agents will help to solve the problem. This approach reflects directly the organisational structure where a business process takes place. It is characterised by the role-based design of different agent types. Correctly speaking, in this approach (the domain of) process management is used as an application of agent technologies, and not vice versa. Moreover, the problem solving process is encapsulated in the domain-specific application knowledge and the general problem solving strategies implemented in the agents. Therefore, the process does not become explicit. Thus, this approach is by no means process-oriented. ADEPT [Jen+97] and PEACE+ [ALO96] follow this approach.

(2) Second, to use agents as a modelling technique for the representation of process elements and as a key infrastructure technology for building a flexible process engine: This approach is a *process-oriented* one which is based on a formalism for describing activities, artefacts, resources, organisational units as well as their relationships. As one may specify all these entities in a uniform object-oriented formalism, the next step goes towards modelling and enacting these entities as agents. This is what we call agent-based framework for process modelling and enacting. Thus, this approach follows the principal idea of business process engineering: It extracts all process-relevant knowledge from the actors of a process and depicts it explicitly in a process model (instead of the implicit definition in the role-based approach). Furthermore, the modelled information is used then in turn to create and derive agents which support enacting of processes and constitute a flexible execution architecture. Different agent types for activity or task management, product or data management and resource management may be identified in the field of process management. Thus, agent-based computing is further used as key infrastructure technology like distributed object management.

In both approaches, agents may be seen as encapsulated, distributed, and interacting entities. In the first approach, they are problem-solving entities, whereas in the latter approach they are process service provider. We belief that both approaches have their justification, their strengths and their weaknesses, since they are based on complementary goals. We do not intend to assess these two directions. Our objective is to develop a uniform framework for modelling and enacting of distributed and dynamic processes based on the principal idea of explicitly specifying processes. Therefore, we will follow the second approach. Moreover, we show how specific role-based agents may be embedded in our agent-based framework which is introduced in the next section.

5. Agent-based Framework for Process Management

In this section we introduce a flexible framework for modelling and enacting of processes which is based on five agent types and interrelationships between them. Since there is no generally accepted definition of an agent, we explain the main characteristics of our agents and relate them to different agent definitions and to distributed object management.

5.1 Fundamentals of Process Modelling

Process modelling and management deals not only with the flow of work, rather it focuses on all aspects of performing a task: (1) the functional perspective (the activities and workflows), (2) the informational perspective (the data and information), (3) the organisational perspective (the organisational units), (4) the resource perspective (the actors and any other resources), and (5) the behavioural perspective (execution states, control and flow). A *process model* combines descriptions or sub-models of these different perspectives of a *process* (cf. [CKO92], [JaBu96]). It is expressed by a *process modelling language* (PML) or a process meta model. Moreover, since co-ordination support is based on the relationships between entities rather than on the entities itself, the (intra)relations between the entities

[1] A third approach which is proposed by Chang and Scott [ChSc97] does not directly apply agents for process management, but uses agents to facilitate various parts of existing workflows. Different agents which act on behalf of a user are introduced as a frontend to existing WFMS and interact with the WFMS through the WfMC APIs ([WfMC94]). Thus, the agents encapsulate the WFMS for the user.

of one model and especially the (inter)relations between entities of different models have to be represented. A process model is divided into at least two layers: On an abstract level, *generic processes* (process types or process templates) are specified which define the process skeleton (process definition domain [DoFe94]). On an actual level, the process enactment model domain, instances of these generic process specifications are created and extended by actual process and product relevant data (e.g. deadlines, actual documents to be used or to be produced etc.). Theses *process instances* are executed under the control of a *process engine*.[2] Finally, processes are performed 'in the real world' by humans and tools. This level is denoted as process performance domain.

5.2 Building Blocks of the Framework

Our modelling and enacting framework takes these perspectives and levels into account. It follows the principle of *separation of concerns*, i.e. modelling of the different perspectives of the process is divided into different *spaces* which capture all information which logically belongs together. As illustrated in figure 1, we distinguish between four spaces, namely the organisational space, the resource space, the repository space, which respectively reflect the corresponding perspectives, and the process space which includes the functional and behavioural perspective. Further, these separation supports clear definition of the interrelationships between these spaces leading to a well defined and tightly integrated overall process model.
To liven these spaces up, object-oriented concepts are well suited since they rely on a natural way of identifying and encapsulating existing entities. Figure 1 sketches the building blocks of such a meta model based on objects and illustrates the intra- and interrelationships between objects of one or different spaces (cf. [ArKe94]). Since we do not intend to provide a full meta model or process modelling language in this paper, we have illustrated only the core elements of the meta model. However, these elements are the building blocks of both, the modelling and enacting aspects of our framework.

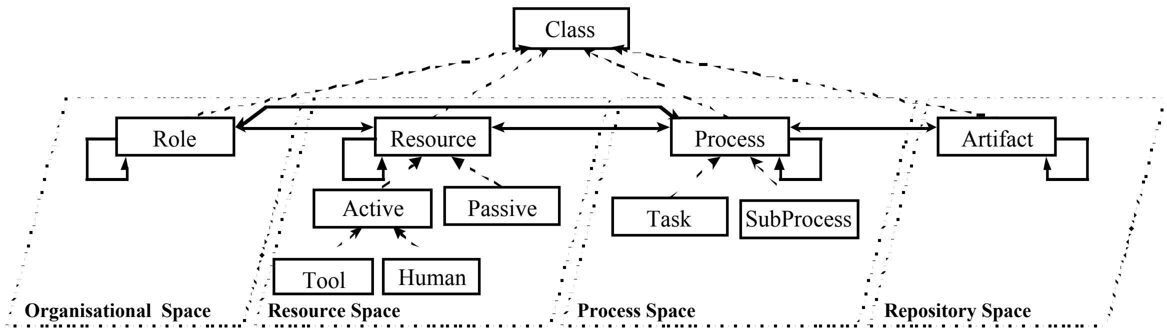

Figure 1: Modelling Spaces and Abstract Process Meta Model

The principal idea to get an integrated view on the modelling and enacting aspects of a process in our framework is now to understand the entities or objects of the model as agents. This is the next step forward: From passive objects to active agents, from modelling to enacting. It leads to an agent-based approach where agents are interactive and concurrent-executing entities which communicate in a common agent communication language (in adoption to [Hew77] and [GeKe94]). For enacting, we derive agents from the objects in the process model. These agents take the process knowledge from the generic process descriptions and may be enriched with additional context-dependent knowledge of the environment in which they act. They maintain all process instance information and handle inconsistencies between the actual process enactment and performance states (cf. [DoFe94]). Thus, the process model is operationalised to a process support agency as illustrated in figure 2. Intra- and interrelationships between the sub-models are used to establish communication channels between the agents. Note, that in general there is no direct transformation, i.e. not every object inevitably corresponds to an agent.

We distinguish agent types for process management (in the narrow sense), data management and resource management which correspond to the process, repository and resource space, respectively. The organisational space is not directly reflected by an agent type but imposes constraints on the actual agent architecture and the agent communication. An brief example which includes all levels is given in figure 3.

[2] [CFF93] denotes the abstract level as template variation and distinguishes on instance level between enactable and enacting variation.

(1) ProcessAgent: A ProcessAgent acts as a distributed process engine (corresponds regarding some aspects to the 'Workflow Engine' in terms of [WfMC94]). It is the basic execution control instance and interacts with both, other ProcessAgents and the various agents which are in charge of handling the repository and resource space. ProcessAgents do not do the work, they just manage the flow of work (i.e. they co-ordinate the various complex and atomic activities within a process and the control and data flow). A ProcessAgent may be responsible for all activities of a complex process or may spawn new (Sub)ProcessAgents and shift the responsibility for sub-processes to them. Further, we distinguish ProcessAgents which manage a (complex) process from those which are associated with an atomic activity or task (i.e. a leaf node of the process hierarchy) and which we denote as TaskAgents.

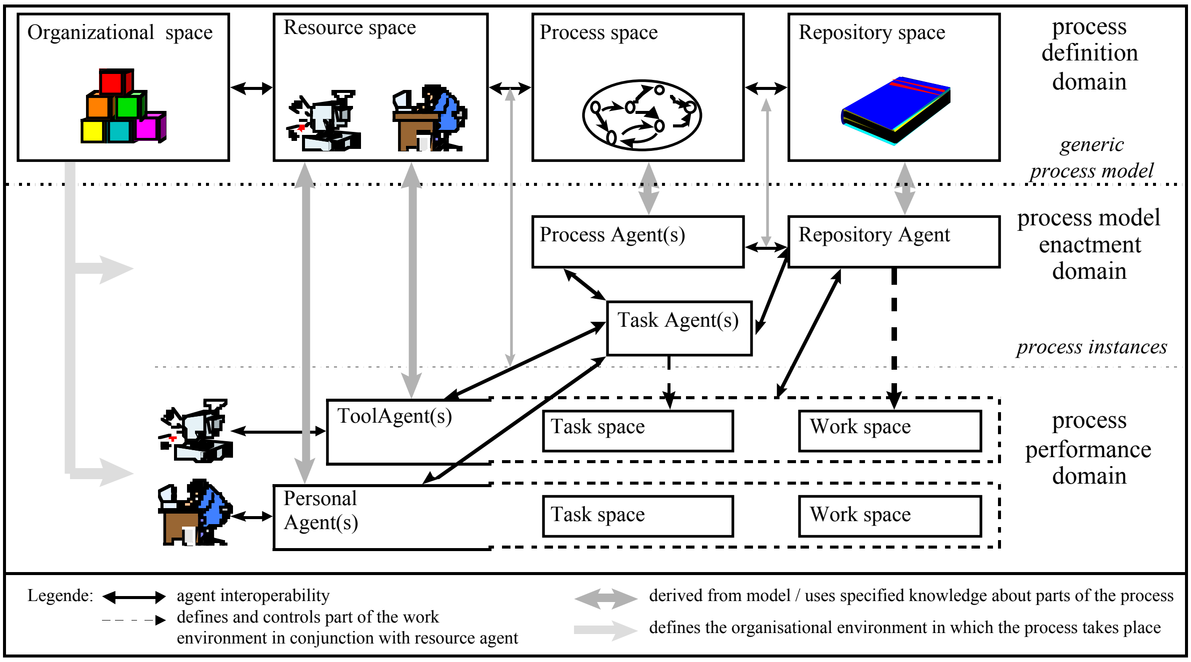

Figure 2: Agent-based Modelling and Enacting of Processes

A *TaskAgent* is in charge of providing process performance support and transforming all performance feedback information into the process execution state. TaskAgents and ActorAgents (see below) provide the interface between the process model enactment domain and the process performance domain and tightly interoperate. The TaskAgent delivers all context information which is needed to perform the task and informs the ActorAgents of changes of this information (e.g. suspension of the task, change of input etc.). Further, it is responsible for resource allocation and scheduling. The ActorAgent acts on behalf of a human or machine and performs or supports performing the desired task. It gives feedback of the actual process performance state and particularly of all process deviations and exceptions (cf. [BDF94]). TaskAgents and ProcessAgents have the knowledge of how to handle these situations with respect to the specified process.

In general, all ProcessAgents react on dynamic changing situations. Thus, they provide the mechanisms for on-the-fly modifications of the underlying enacting process model instance, especially for late refinement and binding of processes.[3] Furthermore, in conjunction with the PersonalAgents (see below) they allow user-specific customisation of processes to avoid totall prescription (cf. [Hei91])

Note, that there is no fixed architecture of the process engine. Rather, depending on the actual organisational environment in which the process takes place several ProcessAgents are created and will interoperate. In particular, the ProcessAgent acts as a process interface of an organisation and hides all internal information. Thus, this agent-based architecture is *scaleable* and *adaptable* to different organisational settings. The architecture may differ from fully centralised (i.e. one ProcessAgent or process engine) to fully distributed (i.e. every process step constitutes a ProcessAgent or TaskAgent).

(2) RepositoryAgent: The RepositoryAgent provides the basic document management functionality. First of all, it provides uniform data retrieval and storage as well as access control for all documents and business objects which are used and furnished in a process. Thus, it acts as an integrated repository which encapsulates different storage systems and, in conjunction with the ActorAgent, provides

[3] Note, that every agent has limited planning capabilities to react on dynamic changing situations. However, general planning and modelling functionality is provided by a modelling component.

the work space for an actor. One or several RepositoryAgents may act as centralised or distributed document management system, respectively. Furthermore, these agents are responsible for version management and maintaining data dependencies to support impact analysis of changes. Moreover, they provide basic co-operation support capabilities like concurrency control, transactional features

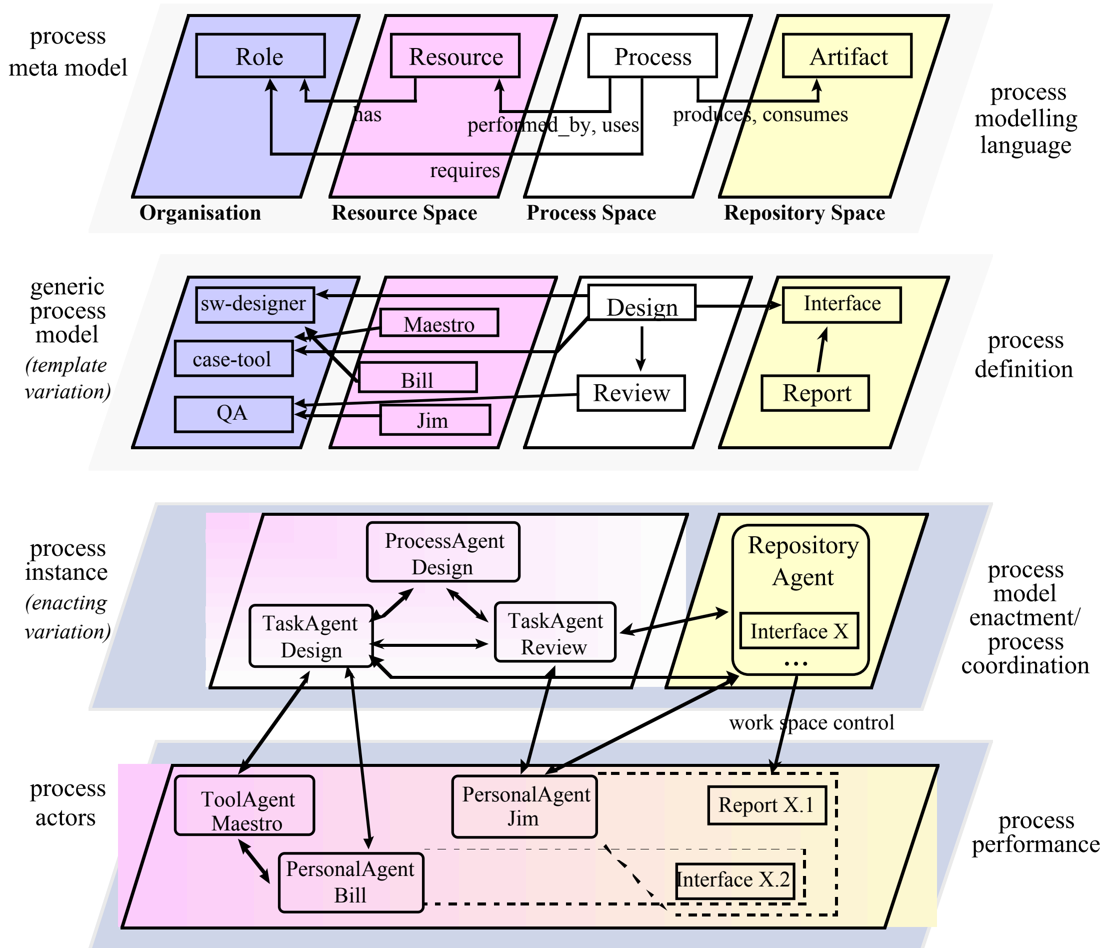

Figure 3: Example of the Process Levels and the Agents within

and change propagation which are of particular interest for advanced process management.

Since process descriptions are documents as well, they are managed in a specific repository - the process knowledge base - by an ProcessRepositoryAgent. Note, that the underlying process knowledge base and ProcessRepositoryAgent which provide the fundamental modelling functionality are not explicitly illustrated in figure 2 for the sake of readability.

(3) ResourceAgents are in charge of managing the resources used in the process. We distinguish between passive resources like time, auxiliaries etc. and active resources like humans and tools which may act as process performers or provide basic process performance support. Active resources are managed by an *ActorAgent* (which corresponds regarding some aspects to the 'Workflow Client Application' in terms of [WfMC94]) This agent type provides the interface between an actor of a task (human or tool) and the process management systems. ActorAgents are divided into PersonalAgents for human assistance and ToolAgents for application system integration:

- A *PersonalAgent* supports a human in performing her or his work. It is that kind of agent which is often referred to as 'intelligent agent that acts on behalf of its user' (cf. [ChSc97]) or shortly interface agent ([Nwa96]). In the field of process management, it provides the work environment or agenda for the user, i.e. it manages the task list, provides the right information at the right time, facilitates tool invocation and supports communication and co-operation with other actors in a controlled way.
- A *ToolAgent* deals with application system integration, especially with heterogeneous and autonomous legacy system. A ToolAgent encapsulates the application-specific knowledge of how to interoperate with a certain application system. Beyond control integration, the ToolAgent may be further enriched with knowledge about the used and produced data of the application system to provide semantic data integration.

The intrarelationships (e.g. process hierarchy) and interrelationships (e.g. i/o-relationship) between the modelling spaces are partially useful to define communication channels and to determine the type of interactivity between the agents. E.g. hierarchical dependencies lead to delegation and reporting, the i/o-relationship may be used for passing the actual input and output objects, and most dependencies for notification of relevant events. The agents may equally co-operate or an agent may have the power to give instructions to other agents.

5.3 The Process Management Agent Properties

To avoid misunderstanding about the usage of the notion agent in our framework, we describe the properties of the above defined agent types in more detail and outline the differences to distributed object management. Let us first remark that agents may be described in an object-oriented manner and therefore inherit the advantages like information hiding, abstraction and flexibility as well as the advantage of an unique modelling framework. Moreover, agent technology relies on the same natural way of identifying and capturing existing entities. Thus, agents provide *encapsulation* and meet the generally accepted requirement of abstraction mechanisms for process management.
But, multi-agent technology go beyond distributed object management with respect to their minimal properties autonomy, social ability, responsiveness, and pro-activeness (cf. [WoJe95]). The following list discusses which properties are hold by the agent types defined above:

- *autonomy* defined as operating without direct human intervention is a very strong property for an agent and may be too strong for a general characteristic as [Mül97] suggested. In a limited way, the above sketched agents may autonomously plan process steps, assign resources, automatically trigger actions and so on. However, this is an optional rather than a mandatory property of a process management agent. Obviously, humans have the overall control of the process!
- *social ability* means interacting with humans and agents. Our understanding of a process is based on this property. Thus, the agents for process management have considerable interoperability capabilities based on a common agent communication language which is an important difference to object-oriented programming (cf. [GeKe94]).
- *reactivity*: To deal with dynamic changing situation, process management agents - in particular the ActorAgents and ProcessAgents - must react on process deviations and process model changes.
- *pro-activeness* means that agents are able to exhibit goal-directed behaviour by taking the initiative. In our framework ProcessAgents may take the initiate e.g. for process refinement, dynamic resource allocation, and automatical execution of process steps. PersonalAgents may provide active guidance, i.e. actively notify the user about significant events, supply warnings, etc. ([DoFe94]).

Despite some restrictions, our application of agents to process management keeps the main properties of an agent. Furthermore, the explicit representation of process knowledge which is used by the agent and the comprehensive communication capabilities based on a common agent communication language are the main characteristics and differ from distributed object management (cf. [GeKe94]). However, agents may be built on top of distributed object management services (cf. [Kir96]).

6. Conclusion

Information-technology support for complex, dynamic and distributed business processes as they occur in several engineering domains requires an advanced process management system which enhances currently available workflow management services with respect to integration, flexibility, and adaptation. Our uniform and flexible framework presented in this paper admits these requirements. It uses and adapts agent technology from distributed artificial intelligence for both modelling and enacting of processes resulting in an advanced process management system.

Encapsulation and comprehensive communication capabilities characterise agents and provide a powerful and uniform concept for *integration*:

- System integration and interoperability is improved since the ActorAgent maintains additional knowledge about the application system
- Manual and automatic process steps may be integrated in a natural way by means of TaskAgents and ToolAgents
- Inter-enterprise process interoperability and distributed process execution is supported by ProcessAgents with specific communication capabilities for inter-enterprise communication
- Workflow and document management are integrated by Process- and RepositoryAgents in order to provide process-based and document-based support for collaborative work.

Moreover, the framework is *flexible* with respect to modelling and enacting. The reactiveness property of agents supports handling of dynamically changing situations which may be reflected in the model as well as in the execution architecture. Furthermore, due to the clear separation the framework is easily *extensible*. Particularly, specific role-based agents may be integrated as specialised ActorAgents and co-ordinate their work using Task- and ProcessAgents.

On the other hand, whether the tight coupling of the process model and the architecture of process enactment and performance services is beneficial in contrast to a standard process engine based on a client/server architecture has still to be proved. Within our project MOKASSIN the next steps are to evaluate the framework, which is still more of an idea, by going into details of its concepts and by implementing several parts resulting in a prototype. Additionally three enterprises of 10 will be involved permanently in our project to evaluate our research from the users' point-of-view.

7. References

[Amb96] Amberg, M.: "Modeling Adaptive Workflows in Distributed Environments". To appear in *Proc. of the 1st Int. Conf. on Practical Aspects of Knowledge Management*, Basel, 30th - 31th Oct., 1996.

[ALO96] Alloui, I.; Latrous, S.; Qquendo, F.: "A Multi-Agent Approach for Modeling, Enacting and Evolving Distributed Cooperative Software Processes". In *Software Process Technology - Fifth European Workshop EWSPT'96*, LNCS 1149, Springer, Berlin, 1996; pages 225-235.

[ArKe94] Armitage, J.W.; Kellner, M.I.: "A Conceptual Schema for Process Definitions and Models". In *Proc. of the 3rd Int. Conf. on the Software Process*. IEEE Computer Society Press, Los Alamitos, CA, 1994; pages 153-166.

[BDF94] Bandinelli, S.; Di Nitto, E.; Fuggetta, A.: "Policies and Mechanisms to Support Process Evolution in PSEEs". In *Proc. of the 3rd Int. Conf. on the Software Process*. IEEE Computer Society Press, Los Alamitos, CA, 1994; pages 9-20.

[ChSc97] Chang, J.W.; Scott, C.T.: "Agent based Workflows: TRP Support Environment (TSE)". To appear in *ECRIM/W4G International Workshop on CSCW and the Web*, 1997.

[CFF93] Conradi, R.; Fernström, C.; Fugetta, A.: "A Conceptual Framework for Evolving Software Processes". In *ACM SIGSOFT Software Engineering Notes*, 18(4), Oct. 1993; pages 26-34.

[CKO92] Curtis, B.; Kellner, M.I.; Over, J.: "Process Modeling". In *Communications of the ACM*, 35(9), Sep. 1992; pages 75-90.

[DoFe94] Dowson, M.; Fernström, Ch.: "Towards Requirements for Enactment Mechanisms". In *Software Process Technology - Third European Workshop EWSPT'94*, LNCS 772, Springer, Berlin, 1994; pages 90-106.

[FKN94] Finkelstein, A.; Kramer, J.; Nuseibeh, B.: "Software Process Modelling and Technology". Research Studies Press (John Wiley), Chichester, 1994.

[GeKe94] Genesereth, M.R.; Ketchpel, S.P.: "Software Agents". In *Communications of the ACM*, 37(7), 1994; pages 48-53.

[GHS95] Georgakopoulos, D.; Hornick, M.; Shet, A.: "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure". In*: Distributed and Parallel Databases*, 3(2), April 1995; pages 119-153.

[HaCh93] Hammer, M.; Champy, J.: “Reegineering the corporation. A manifesto for business revolution”. Harper Business, New York, 1993.

[Hew77] Hewitt, C.:”Viewing Control Structures as Patterns of Passing Messages”. In *Artificial Intelligence*, 8(3), 1977; pages 323-364.

[Hei91] D. Heimbigner: "Proscription versus Prescription in Process-Centered Environments". In Proc. of the 6th Int. Software Process Workshop, IEEE Computer Society Press, Los Alamitos, CA, 1991; pages 99-102.

[HJKW96] Heimann, P.; Joeris, G.; Krapp, C.-A.; Westfechtel, B.: "DYNAMITE: Dynamic Task Nets for Software Process Management". In *Proc. of the 18th Int. Conf. on Software Engineering,* IEEE Computer Society Press, 1996, pages 331-341.

[JaBu96] Jablonski, St.; Bussler, Ch.: “Workflow Management - Modeling Concepts, Architecture and Implementation”. International Thomson Computer Press, London, 1996.

[Jen+97] Jennings, J.R. et al.: “Agent-Based Business Process Management”. To appear in *Int. Journal of Cooperative Information Systems*, 1997.

[Kir96] Kirn, St.: “Kooperativ-Intelligente Softwareagenten”. In *IM - Information Management*, 11(1), 1996; pages 18-28.

[Mül97] Müller, H.J.: “Towards Agent Systems Engineering”. To appear in *International Journal on Data and Knowledge Engineering, Special Issue on Distributed Expertise*, 1997.

[Nwa96] Nwana, H.S.: “Software Agents: An Overview”. In *Knowledge Engineering Review*, 11(3), 1996; pages 1-40.

[WoJe95] Wooldridge, M.; Jenings, N.: “Intelligent Agents: Theory and Practice”. In *Knowledge Engineering Review*, 10(2), 1995; pages 115-152.

[WfMC94] The Workflow Reference Model. Workflow Management Coalition Specification, WFMC-TC-1003, 1994.