Using the Agent Paradigm to Improve Business Process Modelling

Denis Berthier [*], Chantal Morley [**] & Michel Maurice-Demourioux[***]

GET/INT (Groupe des Écoles des Télécommunications)
9 rue Charles Fourier, 91011 Evry Cedex, France

(*) Tél. : +331 60 76 41 22 – email: denis.berthier@int-evry.fr
(**) Tél. : +331 46 21 36 98 (rép.-fax) – email: chantal.morley@int-evry.fr
(***) Tél. : +331 60 76 47 35 – email: michel.maurice-demourioux@int-evry.fr

Abstract: This communication aims at showing how some of the concepts of the agent paradigm can be used to improve business process modelling. In a first part, we recall the fundamental characteristics of the agent paradigm, in particular how it extends the reusability and interoperabilty goals of the object paradigm. In a second part, we propose a unified framefork for the modelling of business processes, by means of a metamodel whose concepts are aligned with the standard process definitions and match the available modelling languages; we explicit the asumptions that support it. Then, we highlight two of its main limitations: re-using dynamic components is difficult and representing processes with emergent structuring is awkward. In a third part, we show how the agent and the goal concepts lead to a richer modelling language for processes and how they open the way to building models with dynamic components: one can both re-use existing activities and build parts with various degrees of structuring. To this end, we propose to extend our first metamodel to one that we name Activity-Actor-Agent. Finally, we hint at directions for our future work, both at the conceptual and at the methodological levels.

Keywords: information system, modelling, business process, reusability, interoperability

Enrichissement de la modélisation des processus métiers par le paradigme agent

Résumé *: Le but de cette communication est de montrer comment certains concepts du paradigme agent permettent d'enrichir la modélisation des processus métiers. Dans une première partie, nous rappelons les caractéristiques fondamentales du paradigme agent, en particulier comment il étend l'objectif de réutilisabilité et d'interopérabilité du paradigme objet. Dans une deuxième partie, nous proposons un cadre unifié pour la modélisation des processus métiers, sous forme d'un métamodèle dont les concepts respectent les définitions normalisées d'un processus et peuvent être mis en correspondance avec ceux des principaux langages de modélisation actuels ; nous indiquons les hypothèses sur lesquelles il repose. Puis, nous mettons en évidence deux principales limitations : la difficulté à réutiliser des composants dynamiques et la lourdeur de représentation de processus à structuration émergente. Dans une troisième partie, nous montrons comment les concepts d'agent et de but conduisent à un langage de modélisation de processus plus riche et ouvrent la voie à la construction de modèles ayant des composants dynamiques : on peut à la fois réutiliser des activités existantes et bâtir des parties avec des degrés variés de structuration. Nous proposons pour cela une évolution vers un métamodèle étendu, nommé Activité-Acteur-Agent. Nous terminons sur les axes de poursuite du travail, au niveau conceptuel et au niveau méthodologique.*

Mots-clés *: système d'information, modélisation, processus métier, agent, réutilisation, interopérabilité*

INTRODUCTION

In the last decades, the process view has been playing an increasing role both in organizational theories and in the information systems (i.s.) area. Process modelling is recognized as a key element when representing an i.s. behaviour [12]. An i.s. process is considered to be an information oriented view of a business process [1], and successful i.s. design starts with business process modelling [8]. However, based on the generic definition of a process [18], most process models give a central place to the concept of an activity. This limits the reuse of process models and it hides the communication dimension which may be essential for some processes.

This paper aims at showing how some concepts of the agent paradigm can be used to improve business process modelling. In a first part, we recall the fundamental characteristics of the agent paradigm. In a second part, we propose a unified framefork for the modelling of business processes, by means of a metamodel whose concepts are aligned with the standard process definitions and match the available modelling languages; we also highlight two main limitations of this metamodel. In a third part, we propose to extend it with three core concepts: Activity, Actor and Agent.

1. THE AGENT PARADIGM

1-1 From reusability to interoperability

In the world of software, the major goal of reusability has been partially achieved in the object oriented approach by the introduction of generic elements that can be specialized. However, in the agent approach, the initial goal of reusability has been rephrased into a more ambitious one of interoperability.

The reusability problem consists in designing software components that can be reused in various applications. In essence, the typed languages, and then the object oriented ones, have been developed to solve this challenge. Notwithstanding its unquestionable success, the object oriented approach faces some limitations: object classes that have been designed independently of each other or written in different programming languages cannot be integrated unmodified into a unique program.

But, in the world of computer and communication systems, heterogeneity is nowadays an inescapable reality. This has led to the notion of interoperability – defined for software components as the capacity to "work" together with no preliminary rewritting, although they were designed independently of each other, programmed in different languages, and although they may run on machines of different makes, with different operating systems, and distant from each other. Interoperability is not an alternative to normalization, which is still necessary to avoid a disorganized and useless multiplication of incompatible systems; on the contrary, it complements it, because the systems that one expects to interoperate are already normalized; it allows a compromise between heterogeneity and evolutivity on the one hand and uniformity and normalization on the other hand [5].

Interoperability concerns data as well as programs: on the data side, one finds the techniques of a) syntactic normalization of query languages and b) semantic integration of conceptual models; on the programs side, one finds the techniques of a) normalized abstract specifications for object behaviours, and b) interpretation of these specifications by a set of system services running on computers grouped into a "platform"; the platform role is to use these abstract descriptions to localize and invoke the corresponding objects, wherever they may reside on its different machines.

The agent approach generalizes the solutions brought to the reusability and communication problems.

1-2 Agents reusability and interoperability

Agents can be understood as a new design and programming paradigm [30]; they are endowed with a "communicational" side, inherited from software engineering, and a "mentalistic" or "cognitive" side, inherited from artificial intelligence (AI).

Concerning the "communicational" side, the multi agent systems (MAS) technology, as it is normalized by FIPA, assumes the interoperability problematics and it aims at pushing further the above faculties. Technically, the agents are particular types of objects, defined by their ability to "communicate" between them through messages in a standardized format; each message is an instance of an abstract, conceptually high-level, "communicative act"; in particular, it abstracts away from any technique and any protocol at the level of telecommunication networks; there is a finite number of such primitive communicative acts, the set of which constitutes a universal language (for instance the FIPA-ACL). This distinguishes the agents from the general objects, for which (even though they are sometimes misnamed "messages") the "methods" are in fact only calls to procedures specific to the relevant object class. Agents allow the reuse of existing software, through a technique named wrapping: every information source (database, XML page, etc.), every object, every program, etc., is wrapped into an interface that makes it appear from the outside as if it were an agent. Moreover, the SMA paradigm permits the design of system architectures that are *a priori* much more varied, flexible and robust than the classical client-server.

As for the "mentalistic" or "cognitive" or "intentional" side, the agents comply at least with Newell's definition [24]: an agent is governed by "goals", it possesses "knowledge", which it puts to rational use in order to reach its goals. In this domain, where AI has not always been very careful, much attention must be accorded to vocabulary: here, the "knowledge" is formal symbolic knowledge, a man-machine hybrid, that can be understood by a human being who knows nought about the computing subtleties of a machine that is able to make use of it, and that can be used by a computer that "knows" nought about the psychic subtleties of a human being who is able to understand it [5]. In practice, this knowledge often appears as a set of rules, expressing, in the vocabulary of the application domain, the expertise necessary to treat the problems the agent is supposed to solve; and the word "rational" means that the system is able to exploit these rules by formal inference. One can also find other "mentalistic" components: "intentions", "beliefs", "commitments"... that can lead to refined use of the rules.

These two faces of the agents, although they are conceptually and technically distinct, are actually tightly bound by the conceptual level at which they acquire their full meaning. Thus, every message expresses an "intention of communication", in the sense of Austin and Searle's speech acts theory, as it has been formalized subsequently [31]; it refers to a particular (applicative) ontology and to a precise content language (in which the ontology and the content are expressed) and the content of every message, which is the most specific part of it, refers to the agent's "knowledge", expressed in the very vocabulary of the application.

In the same way as the object approach has been used to enrich the conceptual representation of the information systems, in particular the static part with the introduction of the generalization / specialization relation, we propose to take the agent approach as a starting point to overcome the limitations of the dynamic representation, more precisely of business process modelling – while making sure, of course, not to mix up the different levels of modelling.

2. BUSINESS PROCESS MODELLING

2-1 Concepts for business process modelling

There is quite a number of business processes modelling languages, with rather similar competing notations. Trying to match these formalisms, we have previously built a core metamodel with the main concepts and their relationships [22, 23]. Without attempting to cover all the concepts of the available modelling languages, this minimum set is sufficient for representing most process configurations (Fig. 1).

Our metamodel is based on two assumptions. First, the underlying modelling approach is a top-down one. A Process is initially defined by the Purpose assigned to it. It can be described at several levels of granularity, but the last level is the only one to be detailed. This is particularly useful when drawing a cartography of all the enterprise business processes. Then, the notion of Activity is a central concept, due to the influence of the standard process definitions, especially in ISO9000:2000 [18] and in the ENV12204 standard which is dedicated to enterprise modelling [10]. The starting point is to model the Activities and then to define the suitable organization with Roles and Actors. Besides its conceptual content, the metamodel provides an aid for business analysts to design processes.

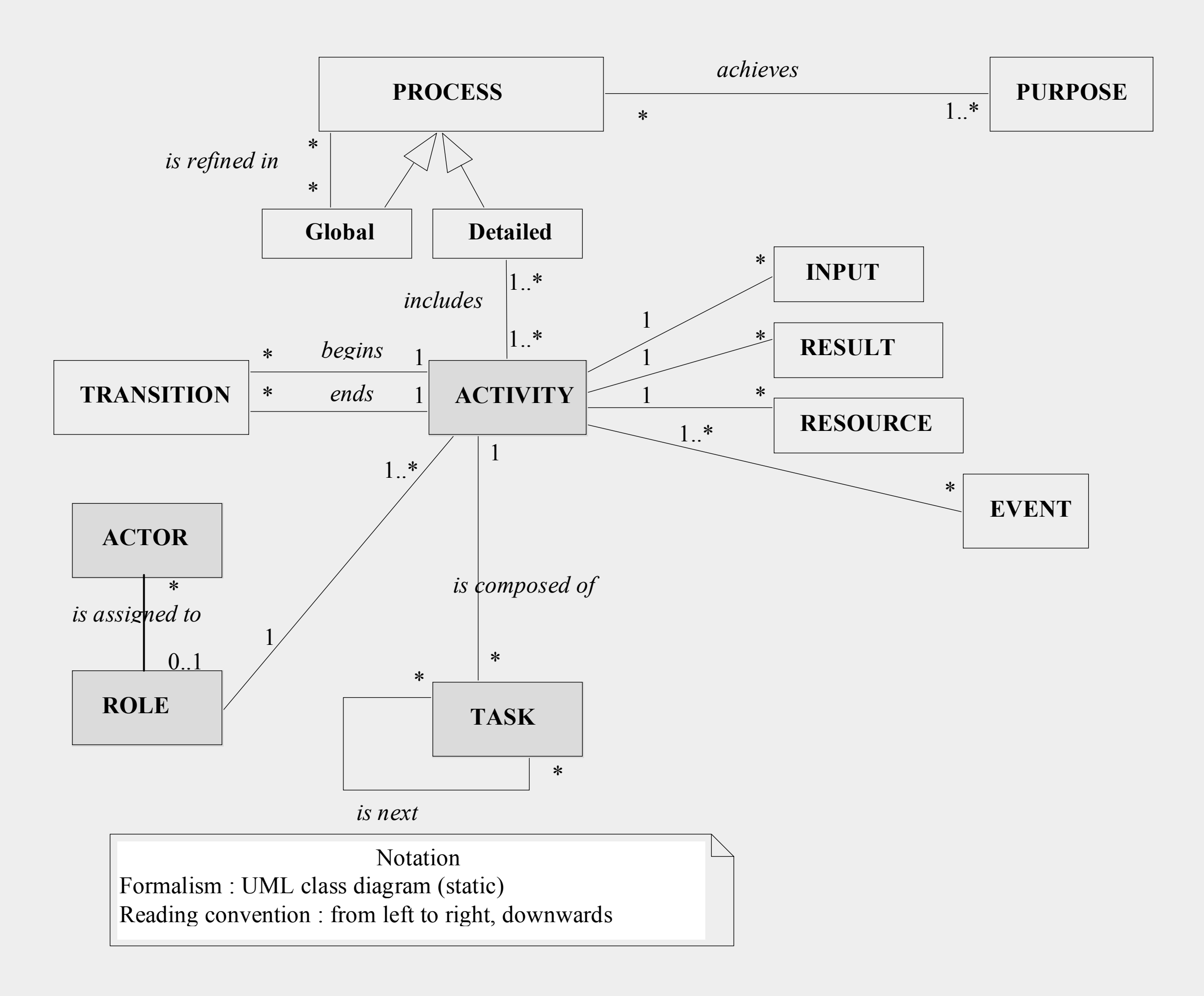

Figure 1: Core process metamodel

A business process is thus viewed as a set of Activities, undertaken with a precise Purpose, and assigned to human, institutional or automatized Actors.

A process is generally implemented within an organizational context; i.e. groups of activities – corresponding to the notion of Role – are supposed to be assigned to a unique Actor. More precisely, a Role is an expected behaviour during process execution.

A Role intended for a computer represents a software specification, whereas a Role assigned to a human Actor corresponds to a specific responsibility for achieving properly a number of Activities and producing the corresponding deliverables. A human Actor is generally entitled to make decisions at the activity level.

When executing, a Process involves resources and may generate or be affected by internal or external Events. The structure of the Process corresponds to the decomposition into Activities and their relationships.

2-2 Structuring business processes

Based on the metamodel, a process can be defined with an explicit and precise ordering of the activities, whereas for others, the links can hardly be expressed because of the variety of possibilities when executing the process. Thus, several approaches in structuring a process have been distinguished [32, 20].

The first approach is often called "mechanistic", as the process model aims at indicating precisely the sequence of activities that must be executed and their content. This is supposed to increase efficiency (optimization of the means) and effectiveness (reaching the objective). Most modelling techniques fall into this category. Links between activities are represented by transitions. They act as milestones when considering a business process as a system transformation. The result of an activity represents an entry element for the subsequent activity.

In the second approach, often called "systemic", activities are viewed as components interacting and reacting to events. Activities are linked by means of results, which act as events : the result of an activity represents a triggering event for another activity or for the same activity in case of a feedback loop. The model includes a number of possible paths for the process execution.

In the third approach, called "enacting" or "social construct", the analyst chooses not to establish precise paths between the activities. The sequence of activities is constructed by the actors during the execution of the process. Every activity can be triggered, interrupted or modified according to specific events. An event may be a temporal event or may originate in another actor, internal or external. This category of representation corresponds to processes which are not fully defined in advance, for example processes with a unique occurrence.

2-3 Limitations of the core metamodel

The above metamodel suffers from two major limitations.

The first limitation has to do with structuring approaches. The metamodel is suited to representing mechanistic processes, as well as systemic processes provided they have a reasonable degree of complexity (i.e. limited numbers of possible links between the activities). On the other hand, when representing "social construct" processes, the analyst can only introduce a limited flexibility by using conditions – for a task, a transition or an event – and event-based activity triggering.

When decisions and inter-actors communications play a key role, the corresponding process model is generally not very user-friendly. An example is given on figure 2, which represents a collaborative process involving a number of team members, managed by a leader who is responsible for delivering the final product within time constraints (a single generic team member has been represented). The role of every team member includes two activities: a production activity aimed at delivering a specific result, and a communication activity, enabling him to communicate with the other team members and with the leader. The process is initiated by a starting event and is completed when the final deliverable has been produced by the leader's control activity. Each process execution depends on team members decisions (for example: to solicit another actor or to answer a solicitation), on resources availability (which can modify the collaboration requirements) and on the team members involvement.

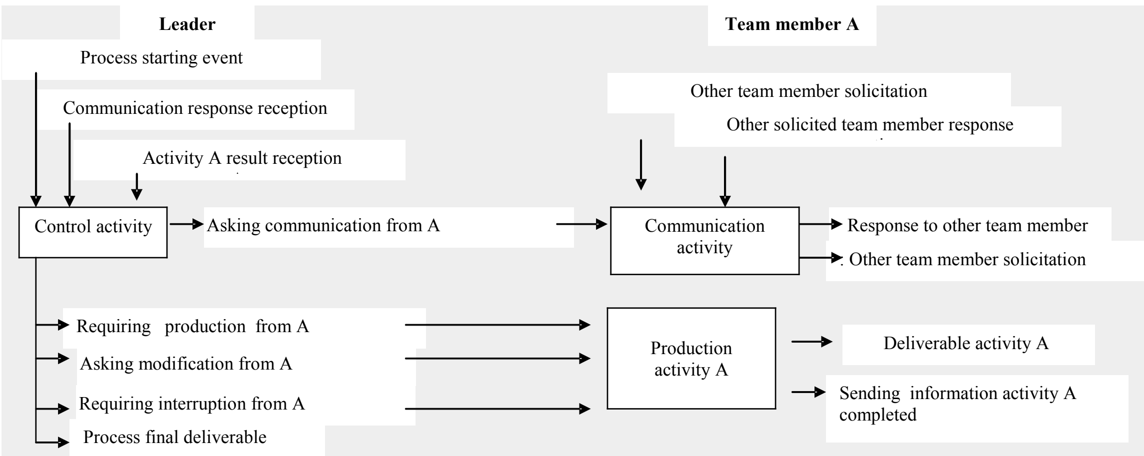

Figure 2: Social construct process

The second limitation of the metamodel results from the underlying design approach. The object-oriented approach is of no help in the design of reusable process components. Moreover, the definition of activities in a specific organizational context is a strong barrier to reuse. In fact, one can notice that the current modelling techniques (IDEF, UML, Ossad, Merise, Petri nets...) do not mention a component-based design.

These two limitations can be related to the two issues of agents communication and software components reuse. This is the reason why we have enhanced our business process core metamodel by introducing concepts inspired from the MAS paradigm.

3. BUSINESS PROCESS MODELLING ENHANCEMENT

3-1 Previous works

Each one with specific purposes, a number of researchers have tried to bring closer MAS modelling and information systems modelling. As reviewed by Kishore & al. [19], many authors have proposed a model for representing a multi-agent system. From this point of view, intelligent agents are the only actors involved in the process; human actors are generally considered as system users. Some authors have used CASE modelling languages, notably UML, for representing such systems [26, 4].

Other researchers have focused on inter-organizational communication modelling. They have proposed to formalize communications, notably by using the concept of a contract [34, 16]. Some others have proposed a business process view either as a "conversation" [2] or as a succession of communication loops between customer and supplier [21], based on the speech acts theory.

It seems that few authors have tried to enhance existing concepts for representing system dynamics at the business process level. Wagner [33] starts from the static vision of an information system (entity-relashionship modelling) and proposes a metamodel for representing business processes. He distinguishes between active agents and passive objects. The proposed formalism is close to UML diagrams, with the addition of an agent interaction language.

Instead of shifting too much from the current paradigm, it seems to be highly interesting to enhance our core metamodel, notably to overcome some of its limitations. This is Kishore's approach [19]; he has proposed a synthetic framework both from MAS modelling and from integrated information systems modelling. We adopt a similar approach for enhancing detailed process modelling.

3-2 Evolution toward an Activity-Actor-Agent metamodel

We propose to introduce two new concepts: Agent and Goal. Figure 3 shows the evolution from figure 1.

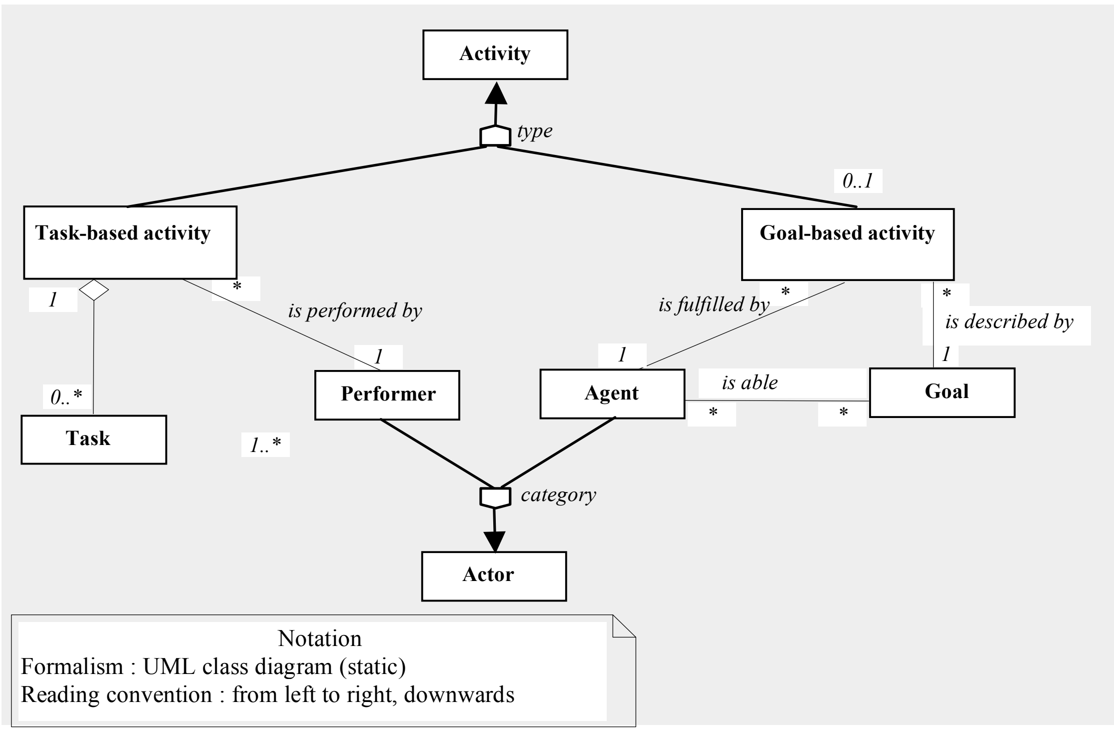

Figure 3: Activity-Actor-Agent metamodel

The concept of an Actor is defined in the following way.

An Actor is an active element (human, organizational entity or machine), involved in the process Activities. An Actor can be internal or external, and a Process can be executed by several Actors.

The concept of an Agent allows to specialize the concept of an Actor. An Agent is defined as an Actor able to achieve a Goal in an autonomous way. On the other hand, a Performer is an Actor who performs precise Tasks.

Thus, we have specialized the concept of an Activity. The subtype "task-based activity" means that the Activity is described by the corresponding Tasks to be performed, and will be assigned to a Performer. The subtype "goal-based activity" is described by the Goal to be achieved, and can only be assigned to an Agent with adequate competency.

The concept of a Goal corresponds to a limited objective, whereas the concept of a Purpose pertains to the characteristics of a Process. A Goal can characterize an Activity ("an activity is designed to achieve a specific goal") as well as an Agent ("an agent has competency for achieving a specific goal"). Unlike other authors [Rolland, 1998], we do not consider that a Purpose is a composition of Goals, because the two concepts are not at the same semantic level. At the Process level, a Purpose is more global and should have direct correspondance with the enterprise strategic orientations. A Goal is at the functionality level, it corresponds to a deliverable (product or service). Different detailed Processes, thus different Goals, can correspond to the same Purpose.

3-3 Contributions of the metamodel

The contributions of a metamodel are both theoretical and practical. A metamodel is a representation of the elements of a modelling language (theoretical side), in words that have a meaning in the application domain – here, for an organization in which an information system is going to operate. From that point of view, it defines the top level ontology that will be used to describe this organization – here, the concepts Activity, Agent, Goal, etc.[1]

But a metamodel is also a guide for the analyst when he tries to represent (or even to re-enginner) a business process (practical side). Our proposal concerns this second side as well. The fewer new concepts introduced, the more efficient this guide. However, since we were inspired by an existing technical paradigm, there were two major pitfalls, that we tried to avoid: that of an uncontrolled proliferation of new concepts and that of a confusion between the two levels: technical and organizational.

From the technical MAS paradigm we have retained only the two concepts that we consider as the most fundamental at the organizational level – which is the level of the metamodel: Agent and Goal, as well as their associations. In particular, note that we have not retained the concept of a Message, although it is central in the MAS paradigm, because it is doutfull that this concept, as such, means anything at the organizational level.

In concrete terms, with our metamodel, one of the first decisions that must be taken by the analyst, for every Activity, is: will it be described in a classical procedural way, like a strictly organized set of Tasks, or, in a more flexible way, by its Goal?

The extension of our first metamodel based upon these only two concepts, Agent and Goal, introduces several new facilities for the modelling of processes.

1. *Using dynamic components in process modelling*

If one relies on the distinction between Agent and Actor and on the autonomy of the Agents, one can model Processes using some Activities which are solely defined by their Goal and are placed under an Agent's responsibility (human or software) whose detailed behaviour is concealed. This brings a solution to the question of reuse. Indeed, such an Activity will be potentially reusable by several processes ; therefore, it can be considered as a component for the dynamics.

2. *Inter-organizational process modelling*

One can also consider that an Agent is (or belongs to) an external i.s.; one can thus represent Processes resorting to Agents that are external to the enterprise information system. An Activity entrusted to an external Agent will not be described as a defined Task, but it will appear only in terms of its Goal and of the result it has to transmit to a dependent Activity.

3. *Modelling systemic processes*

The autonomy of the Agents allows the representation of Processes some parts of which are structured with precise Tasks to accomplish (for example, the beginning and the end of a collaborative process), and other parts

[1] Of course, the metamodel cannot define this ontology without relying itself on a meta-ontology (what one sometimes calls the epistemological or meta-structural level) – in our case the concepts Class and Association (in this paper, one can find them in the UML class diagram, in Fig. 1 and 3.).

of which are not, but where the Agents can communicate and cooperate freely within the limits of their only constraints: their Goal and the result of their activities.

3.4 Illustration

An illustration of the Activity-Actor-Agent metamodel is given in figure 4.

The model represents a project which has been structured in the following way. Two Actors (project office and project manager) are Performers and have precise Tasks to accomplish. The project office is supposed to initiate the project by developing a project charter. Then the project manager has to plan the project and to contract with two service providers for delivering specific results. The providers are Agent-type actors, who are assigned a Goal-based activity. After completion of activity A and activity B, the project manager has to synthesize the results and prepare a presentation document. In the end, the project office has to perform project closure by archiving the informations of the project.

In this example, we have both: 1°) detailed Activities, precisely structuring the work to be performed, and 2°) Activities which appear as black boxes, the Agents being free to accomplish their work in their own way, as long as they achieve the assigned goal.

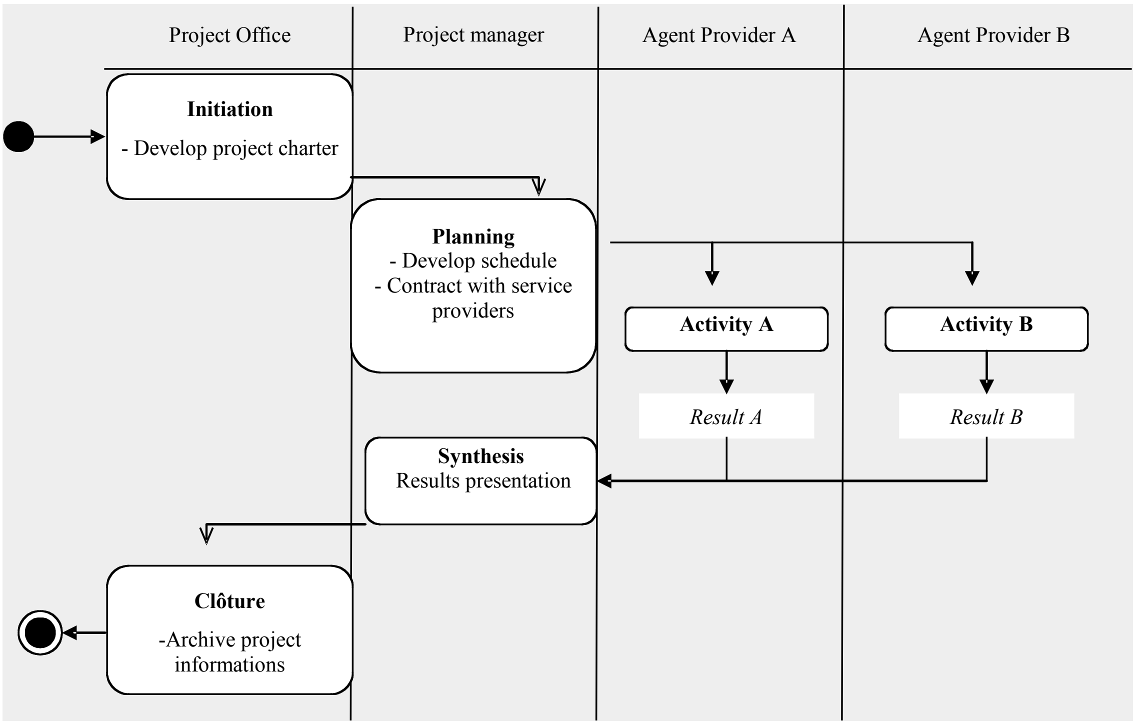

Figure 4: Activity-actor-agent based process

CONCLUSION

The MAS paradigm seems likely to lead to some enrichments to business process modelling. However, a lot of work remains to be done to turn it into a unifying framework upon which one could build a method for the conceptual analysis and the development of an i.s. dynamics. Two directions can be indicated for further work. On the conceptual side, one can study whether introducing other concepts from the agent paradigm is relevant; this applies in particular to the Message concept and to the reference to a standardized communication language, as it has already been done for workflow modelling [21]; but, according to us, it seems that the question should rather be: what should we substitute for them that would be meaningful at the organizational level to which the metamodel belongs ? On the methodological side, we should express the Goal and Agent concepts into one or several modelling languages (for example UML, as it is used for representating organizational information systems) and to highlight the resulting possibilities for structuring collaborative processes.

REFERENCES

1] S. Alter : *Information Systems : A Management Perspective*, 3e éd., Addison-Wesley, 1999.

[2] E. Aurimaki, E. Lehtinen E. & K. Lyytinen, A speech-act-based office modeling approach, *ACM transaction on Office Information Systems*, vol. 6, n° 2, 1988.

[3] J. Austin : *How to do things with words*, Clarendon Press, Oxford, 1956.

[4] B. Bauer & J. Odell : UML 2.0 and agents : how to build agent-based systems with the new UML standard, *Engineering Applications of Artificial Intelligence*, 2004.

[5] D. Berthier : *Le savoir et l'ordinateur*, L'Harmattan, Paris, 2002.

[6] A.T. Berztiss : Domain Analysis for Business Software Systems, *Information Systems*, vol. 24, n° 7, pp. 555-568, 1999.

[7] J. Bradshaw ed. : *Software Agents*, AAAI Press / MIT Press, Cambridge, Mass., 1997.

[8] K.A. Butler, C. Esposito & R. Hebron : Connecting the design of software to the design of work, *Communications of the ACM*, vol. 42, n° 1, pp. 39-46, 1999.

[9] J. Castro, M. Kolp & J. Mylopoulos : Towards requirements-driven information systems engineering : the Tropos project, *Information Systems*, vol. 27, pp. 365-389, 2002.

[10] CEN/CENELEC : ENV12204, Advanced Manufacturing Technology – Systems Architecture – Constructs for enterprise modelling, 1995.

[11] P. Cohen & H. Levesque : Communicative Actions for Artificial Agents, *in* [7].

[12] B. Curtis, M.I. Kellner & J. Over : Process modeling, *Communications of the ACM*, vol 35, n° 9, pp. 75-90, 1992.

[13] T. Finin, Y. Labrou & J. Mayfield : KQML as an Agent Communication Language, *in* [7].

[14] FIPA : http://www.fipa.org

[15] P. Green & M. Rosemann : Integrated process modeling : an ontological evaluation, *Information Systems*, vol. 25, n° 2, pp. 73-87, 2000.

[16] Y. Hoffner, H. Lwdwig, C. Gulcu & P. Grefen : An architecture for cross-organisational business processes, *Advanced Issues of E-Commerce and Web-Based Information Systems*. WECWIS, 2000.

[17] I. Horrocks, P. Patel-Schneider & F. van Harmelen : From SHIQ and RDF to OWL : the making of a Web Ontology Language, *Web Semantics : Science, Services and Agents on the World Wide Web*, vol. 1, n° 1, pp. 7-26, Elsevier, Décembre 2003.

[18] ISO9000 , « Qualité et systèmes de management ISO 9000 », Afnor, 2000.

[19] R. Kishore, H. Zhang & R. Ramesh : Enterprise integration using the agent paradigm : foundations of multi-agent-based integrative business information systems, *Decison Support Systems*, sept. 2004.

[20] N. Melão & M. Pidd : A conceptual framework for understanding business process and business process modeling, *Information Systems Journal*, vol. 10, pp. 105-129, 2000.

[21] G. Mentzas, C. Halaris & S. Kavadias : Modelling business processes with workflow systems : an evaluation of alternative approaches, *International Journal of Information Management*, 21, 123-135, 2001.

[22] C. Morley, J. Hugues, B. Leblanc & O. Hugues : *Processus métiers et systèmes d'information*, Dunod, 2005.

[23] C. Morley : Un cadre unificateur pour la représentation des processus, Pre-ICIS, déc. 2004.

[24] A. Newell : The Knowledge Level, *Artificial Intelligence*, vol 59, pp 87-127, 1982.

[25] E. O'Donnell & J. David : How information systems influence user decisions : a research framework and literature review, *International Journal of Accounting Information Systems*, vol. 1, pp. 178-203, 2002.

[26] I. Paik, S. Takami & F. Watanabe / Intelligent agent to support design in supply chain based on semantic web services, *Proceedings of the 4th International Conference on Hybrid Intelligent Systems*, 2004.

[27] C. Rolland, S. Nurcan & G. Grosz : A unified framework for modeling cooperative design processes and cooperative business processes, *IEEE proc. 31st annual Hawaii International Conference on System Sciences*, 1998.

[28] F. Rowe (éd.) : *Faire de la recherche en systèmes d'information*, Vuibert – Fnege, 2002.

[29] J. Searle : *Speech Acts : an Essay in the Philosophy of Language*, Cambridge University Press, Cambridge, U.K., 1969.

[30] Y. Shoham : Agent-oriented programming, *Artificial Intelligence*, vol. 60, pp. 51-92, 1993.

[31] M. Vanderveken : *Meaning and Speech Acts*, Vol. 1 : *Principles of language use* ; Vol. 2 : *Formal semantics of success and satisfaction*, Cambridge Univesity Press, Cambridge, Masss., 1990.

[32] P. Vidal & S. Nurcan : Coordination des actions organisationnelles et modélisation des processus, *in* [28]

[33] G. Wagner : The Agent-Object-Relationship Metamodel : towards a unified view of state and behavior, *Information Systems*, vol. 28, pp. 475-504, 2003.

[34] H. Weigand & W.J. van den Heuvel : Cross-organizational workflow integration using contracts, *Decision Support Systems*, vol. 33, n° 3, pp. 247-265, juillet 2002.

[35] M. Wooldridge : Semantic Issues in the Verification of Agent Communication Languages, *Autonomous Agents and Multi-Agent Systems*, vol. 3, n° 1, pp. 9-31, 2000.