A METHOD BASED A ONTOLOGY FOR MODELLING BUSINESS PROCESSES

Camelia ŞTEFĂNESCU*, **George MANIU***

*"Spiru Haret" University, Faculty of Management

***Abstract**: In the last years modelling business processes have been playing more and more important for Organization theory and for design the Information Systems. The modelling of the business processes must to take inspiration in the multi agent systems (MAS). This paper describes how to develop a ontology for modelling business processes using the UML metamodel. On the theoretical side, it defines the top level elements of a modelling language, in words that have a meaning for the organization and its managers. On the practical side, since this ontology becomes a guide for the analyst when he tries to model a business process.*

***Keywords**: modelling business processes, modelling language, ontology, organization, strategy, structure.*

1. INTRODUCTION

In the general context of globalisation and in a resulting highly competitive and rapidly changing world, information systems (IS) are facing two major tightly interconnected challenges: interoperability (IS of different institutions, or separately developed parts of the IS of the same institution, must communicate or even cooperate) and flexibility (IS must be able to be easily adapted to the frequent changes a reactive institution must make in its business processes).

In the last decades, the process view has been playing an increasing role both in organizational theories and in the IS area. Process modelling is recognized as a key element when representing the behaviour of an IS. An IS process considered to be an information oriented view of a business process and successful IS design starts with business process modelling. But, this dimension tends to become more and more important as one needs to model ore and more complex processes, which are also more and more interactive (project management, decision processes, innovation processes). When one wants to facilitate the modelling of business processes better fitted to the two challenges mentioned at the start, it is natural to take inspiration in the multi agent systems (MAS) paradigm. In this paper we developed the first sketch of ontology, represented as an UML metamodel, coherent with classical approaches.

2. GRAPHICAL REPRESENTATION OF OUR ONTOLOGY

In our ontology, the notion of a Process occupies the highest level. We consider it as a coordinated set of interoperating Activities, which are assigned to Actors; a Process is motivated by a Purpose, the full meaning of which resides at the organization level, where it corresponds to strategic orientations of the institution. An Actor is an active element (human being, organizational entity or software component) involved in some of the Activities of a Process (we therefore assume from the start a distinction between active elements and passive ones – that can be input or output to processes, resources, etc.). An Actor can be internal or external to the institution and a Process can be executed by one or several Actors.

We distinguish three different types of Activities, with three corresponding different Statuses for the Actors involved in them. The Status is related to the kind of autonomy expected from the Actor while he performs this Activity; it should not be confused with the much more specific notion of the Role(s) an Actor can have in some Activity (ies) (see later). A given Actor can have multiple Statuses if he participates in Activities of various types: this should allow for more flexibility in the organization of the institution. Finally, a special fourth Status for an Actor is introduced: Pilot, a Process has a unique Pilot, in charge of its management. Before entering into more details, let us display our ontology in the UML "metamodel".

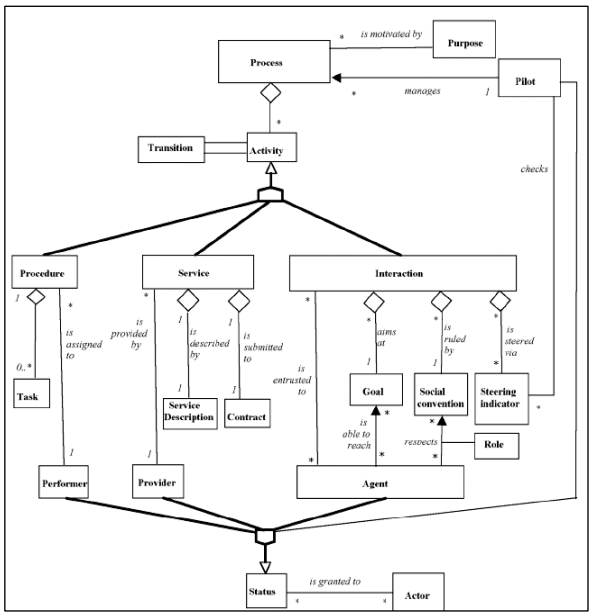

Fig. 1 UML representation of our ontology

3. ACTIVITIES AND ACTORS

We define tree type of an Activity*: Procedure, Service or Interaction.*

3.1. PROCEDURE

A *Procedure* is a kind of Activity defined by the Tasks it is composed of. Generally, it is an ad-hoc Activity, designed for a specific Process, and it has therefore very low reusability. The Tasks sequence (including possible predefined variants) of a Procedure may be specified by an UML sequence diagram or by a task graph or by a Petri net. An Actor to whom a Procedure is assigned has (in the context of this Activity) the Status of a mere Performer: he has no autonomy for modifying this Procedure. Procedures are adapted for the modelling of operational Processes; they are not adapted for more complex Processes.

3.2. SERVICE

An Activity of type Service is described by a *Service Description.* An Actor who provides a Service has (in the context of this Activity) the Status of a *Provider*. In the context of the Process being modelled, an Activity of type Service is not supposed to be described with more detail than: 1) what is specified by its Service Description; 2) the designation of the Provider responsible for providing it (he can be internal or external to the institution); 3) the Contract to which this Provider is submitted.

The Service Description enunciates fixed constraints, which are definitory of the Service. The Contract defines constraints specific to a given instance of the Service and the Provider, that can vary from one instance to another (delays for deliverables, prices, etc.). A typical example of a Service, in case it is a software component, might be a Web service (in which case, the Service Description might ultimately, at the technical level, be formalized in the WSDL language). Notice that, even though he has no option for modifying the Service Description, a Provider has much more autonomy than a Performer: he is totally responsible for the means and the methods he uses to provide the Service and satisfy his Contract.

3.3. INTERACTION

An *Interaction* is an Activity defined by its *unique Goa*, this unique Goal can be satisfied only with the participation of several Actors, whose Statuses are then that of Agents. Although an Interaction is not specified procedurally by rigid Tasks that would define it in detail, it is nevertheless constrained by an organizational framework, i.e. the actions of the various participating Agents are regulated,

both externally and internally: since the Interaction occurs, as any other type of Activity, in the framework of a given Process, it is steered via Steering Indicators (for instance: a planning of deliverables, indices of quality for the output of the Interaction etc.), that will be checked by the Pilot of the Process.

The concept of a Goal allows describing some finality, more limited than the finality attached to the concept of a Purpose (at the level of the Process). Given the Purpose of the Process, the various Activities that compose it and the various Goals associated to Interactions in it correspond to a first level of choice in the way the Process definition can be elaborated so as to reach this Purpose.

The Goal of an Interaction defines fixed general constraints common to any instance of the given Interaction. Nevertheless, contrary to a Service Description that strictly defines the expected result, a Goal can be much less precise. For instance: the Goal of Interaction "prepare an answer to a call for proposal" is not of the same nature as the Service Description "manage invoices"; it leaves open lots of possibilities and it grants the participating Agents much more autonomy (both individual and collective) than a Provider can have.

A Goal characterizes both an Interaction (an Interaction aims at reaching a Goal) and the participating Agents (a group of Agents is collectively able to reach a Goal). It should be noted that, since an Interaction is entrusted to a group of Agents, the (unique) Goal that such an Activity aims at is the Goal common to this group as a whole.

The way this common Goal is decomposed into subgoals, the way those are assigned to the participating Agents and the way all this is coordinated by the group itself depends on the Social Convention ruling the Interaction A group of Agents having a common goal means much more than each Agent having this goal as its individual goal.

In particular, having a common goal supposes that various mutual beliefs must be held and that various types of communication, coordination and cooperation must take place for the group to reach the goal.

3.4. STATUS OF AN ACTOR

As in the classical conception, an *Actor* is an active element that may play a role in the definition and the unfolding of a Process. Once he Activities of a Process has been defined, they are assigned to Actors. An Actor can be internal or external to the institution, and a Process can thus be executed by several cooperating partners.

To the three types of Activities correspond three different possible Statuses for an Actor: Performer, Provider or Agent. Whatever his Status, an Actor can be either a human person, a group of such persons, an organizational entity or a software component; our description at the organization level makes no difference as to this point.

Hereafter, we give some typical examples of how an Actor (in the Process model at the organization level) with a given Status can be implemented at the technical level (if it has been decided that it should be a software component):

- *a Performer* is defined as an Actor with no autonomy: he is expected to accomplish the Tasks assigned to him in total conformance with their procedural description. He necessarily belongs to the institution (because an institution is not supposed to have control on *how* things are done outside).
- *a Provider* is defined as an Actor with total autonomy relative to *how* he manages to provide the Service defined by the Service Description, as far as he also satisfies the further constraints specified in its Contract. His autonomy is autonomy of *means*, not of *goals*. A Provider is typically external to the institution; but it may also be an internal department, providing some peripheral services (mailing department, accounting department etc.).
- *an Agent* is defined as an Actor with the ability to execute Activities in an "autonomous" way, i.e. he is not told how to operate, but only which Goal he must contribute to reach within an Interaction; even his contribution in the Interaction may be formulated in very general terms. Thus, although the autonomy of an Agent is not a full autonomy of its *goals*, it is not only a mere

autonomy of *means*, as is the case for a Provider. An Agent can be internal or external to the institution (since the institution is entitled to have some control on the Interaction via the Steering Indicators and via the final satisfaction of the Goal). Given the autonomy implicit in the notion of an Agent, in case one (or several) of the Agents participating in an Interaction is a software component; its implementation at the technical level may require MAS and/or AI techniques.

A special Status is introduced as a fourth possibility, that of a *Pilot.* Whereas the first three Statuses where attached to Activities, the Status of a Pilot is attached to the Process as a whole. This Actor is unique for each Process; he is in charge of managing the Process (it is highly unlikely that such a Status can be granted to a software component in the near future).

4. SOCIAL CONVENTION

A Social Convention is a set of clauses that regulate the communication, coordination and cooperation among the Agents participating in an Interaction in order to reach its Goal. It is a set of rules and constraints that tie these Agents together in the context of this Interaction – excluding rules specific to each Agent's internal behaviour (such as his reasoning processes, his personal motives). A Social Convention thus defines constraints internal to the Interaction (whereas external constraints are taken care of via the Steering Indicators). Much more flexible than a Procedure, a Social Convention has therefore much more reusability. Moreover, it grants the participating Agents much more autonomy.

The concept of a Social Convention is a very general one, which can have many specializations. What follows has no claim to exhaustively.

4.1. STANDARD CONVERSATION

A seemingly degenerate case of a Social Convention is a *Standard Conversation*. In a Standard Conversation, the communication events between the participating Agents are strongly guided, although the actions they have to accomplish between these events may be very complex and require much autonomy on their part.

This concept is interesting for discussing a potential difficulty of our approach and for precising the limits of our "inspiration from the MAS paradigm". With such inspiration, it might be tempting to rely on the notion of a Standard Conversation as it is standardized by FIPA (Foundation for Intelligent Physical Agents, the international association in charge of standardizing MAS). For FIPA, a standard conversation can be defined starting from the more elementary concept of a message:

- either as an *ad hoc* graph of the possible sequences of messages between agents, using for instance AUML diagrams (Agent UML, an extension of UML);
- or by choosing among a list of standardized communication protocols such as: Contract-net, English-auction, Brokering (where each protocol is predefined with AUML diagrams).

But we consider it would be a major failure for us if we had to define a Standard conversation in this way. The concept of a message, as it appears in so central a position in the MAS paradigm is much too "atomic" to mean anything at the organization level. We want the organization level concepts to appear directly in our ontology with no reference to concepts of lower levels. Concretely, one can specify the notion of a call for proposal (for instance) without having to specify in detail all the possible paths such a conversation can follow (if needed, such detail can be added in subsequent modelling phases). Therefore, we posit in our ontology a Standard Conversation as a subtype of an Interaction. We think that some standard conversations can be abstracted from their technical definition in terms of sequences of messages and asserted in our ontology as specializations of the concept of a Standard Conversation.

Notice that, although a Standard Conversation might look like a Procedure, there is a major difference: a Procedure is decomposed into Tasks that are in turn decomposed into Tasks. In a Standard Conversation, what are defined with some rigidity are only the possible sequences of

communication events between Agents. Neither the internals of each Agent's activities between such events nor the exact content of their outputs is specified.

For instance, in a cfp (call-for-proposal) Standard Conversation, the initiator of the conversation issues a call for proposal. An Agent may answer with a proposal. But, in between, he may have very complex activities, supposing a great deal of autonomy, to elaborate his answer.

4.2. SOCIAL CONVENTIONS BASED ON ROLES

Some Social Conventions can be based on the attribution of specific roles to each participating Agent (one could posit a Role-Based-Social-Convention as a general subtype of Social-Convention).

Consider again the following typical case of an Interaction: "prepare an answer to a call for proposal". Assume the participating Agents are the Commercial Department, the Technical Department and the Legal Department. In a company, when preparing such an answer, each of these (institutional) Agents naturally assumes one or more specific roles in its contribution, corresponding to its competencies.

Moreover, there is naturally a special role for internal coordination of the interaction, which will typically be assumed by the Commercial Department, in addition to its purely commercial role. This last role must be distinguished from the Status of the Pilot of the Process; the coordinator of the Interaction will generally be responsible for interacting with the Pilot of the Process. Defining the associated Social Convention of this example consists in part in formalizing all these natural roles.

Although each participating Agent has a fixed role in it, the Interaction cannot be defined by a fixed sequence of actions from them. This is a main difference with a Standard Conversation.

The actual actions will be determined dynamically from the Goal. For instance, the Technical Department may be confronted to a major difficulty that requires lots of work to assess the feasibility (by its company or by subcontractors) of some parts of the requirements.

4.3. SOCIAL CONVENTIONS IN GENERAL

More generally, a Social Convention can be based on a large panel of social interaction models: negociation, game theory, planification, free collaboration between agents, and remuneration of the agents according to certain economical models. In practice, in an IS, Social Conventions will be more restrictive and more specific than such general models.

In our example "prepare an answer to a call for proposal", in addition to the attribution of a specific role to each agent, a Social Convention might specify that whenever a participating agent finds that he won't be able to fulfil his part of the job in time, he must warn the others immediately (and not let them discover it at the last moment); to whoever this seems too obvious to be explicated: have you have ever worked in collaboration?

A Social Convention also defines, in a more or less direct way, the degree of autonomy of the participating Agents; actually, it is only through this concept that the notion of autonomy, that we have until now used in a very vague manner, can acquire a precise meaning: the autonomy degrees of the various Agents participating in an Interaction are defined by the Social Convention that rules their social behaviour.

There is currently a lot of research activity on social organization of agents in MAS. Just as a brief illustration of the notion, let us cite Jennings GRATE* model (Jennings, N., 1994). Jennings distinguishes several levels on which the various clauses of a Social Convention can bear:

- a set of rules for information communication between the Agents and for assessing the advancement of a common action plan:

1. what type of information each agent must exchange with which other agents, in which conditions (for instance in a reactive or proactive way);

2. what kind of reporting he has to do, to whom, when;
3. what kind of tasks he can delegate, to whom, in which conditions.

- a set of rules defining how the commitments of the participating agents (towards common goals, plans, distribution of tasks, planning) can be taken, re-assessed or dropped.

5. CONCLUSION

In this paper we have proposed general purpose ontology for modelling business processes. On the theoretical side, it defines the top level elements of a modelling language, in words that have a meaning for the organization and its managers. Given the vagueness or the non standardization of the vocabulary in the domain, one should not underestimate the usefulness of having precise definitions for all the terms we use. On the practical side, since this ontology becomes a guide for the analyst when he tries to model (or to re-enginner) a business process. Our ontology introduces several new practical facilities for the modelling of processes: - Inter-organizational processes: one can represent Processes resorting to Services that are external to the corporation. Such an Activity entrusted to an external Provider will not be described by detailed Tasks, but only by its Service Description and by the Contract with the Provider. This case typically includes such examples as Web Services and e-commerce. We have specified only the static part of our ontology. For Interactions, one of the main problems is specifying their dynamics; at the level of detail of this paper, it may seem that our model is of little help and we still have to resort to the classical representations: UML activity. The dynamic side will not be specifying using UML diagram. Nevertheless the concepts of a Goal or a Social Convention may open the door of alternatives to such a semi-graphical representations – which are very cumbersome for complex processes.

BIBLIOGRAPHY

1. Bauer, B., Odell, J., *UML 2.0 and agents: how to build agent-based systems with the new UML standard*, Engineering Applications of Artificial Intelligence, 2004;
2. Filip, F.Gh., *Sisteme suport pentru decizii*, Editura Tehnică, Bucureşti, 2004;
3. Hoffner, Y.H., Lwdwig, C., Grefen, P., *An architecture for cross-organisational business processes*, Advanced Issues of Ecommerce and Web-Based Information Systems, WECWIS, 2000;
4. Jennings, N., *Cooperation in Industrial Multi-Agent Systems*, World Scientific, Singapore, 1994;
5. Kishore, R., Zhang, H., Ramesh, R., *Enterprise integration using the agent paradigm: foundations of multi-agent-based integrative business information systems. Decison Support Systems*, 2004;
6. Mentzas, G., Halaris, C., Kavadias, S., *Modelling business processes with workflow systems: an evaluation of alternative approaches*, International Journal of Information Management, No 21/2001;
7. Paik, I., Takami, S., Watanabe, F., *Intelligent agent to support design in supply chain based on semantic web services*, Proceedings of the 4th International Conference on Hybrid Intelligent Systems, 2004;
8. Wagner, G., *The Agent-Object-Relationship Metamodel: towards a unified view of state and behaviour*, Information Systems, 2003;
9. Weigand, H., Van den Heuvel, W.J., *Cross-organizational workflow integration using contracts*, Decision Support Systems, vol. 33, No3/2002.