Modeling and enacting processes. Some difficulties.

Claude Godart, Pascal Molli, Olivier Perrin

Modeling and enacting processes. Some difficulties.

C. Godart, P. Molli and O. Perrin
LORIA - INRIA
Campus Scientifique, BP 239,
54506 Vandœuvre-lès-Nancy Cedex - FRANCE
E-mail: godart@loria.fr

Abstract— **This paper reports about difficulties when modeling processes. It is based on experience gained in different contexts: virtual teams (software engineering, administration), well established virtual enterprises (embedded system (software and hardware)), and temporary virtual enterprises (building construction). Most of the difficulties are common to most organization types, but are more or less acute depending on this type.**

I. Introduction

This paper reports about difficulties encountered when modeling processes. It is based on experiences gained in different contexts:

- virtual teams in software engineering and administrative applications,
- a well established virtual enterprise producing embedded systems (hardware and system) with a assembler partner and equipment providers,
- temporary virtual enterprises in building construction (we are concerned with middle size constructions implicating several trades which are rather small sized companies).

Most difficulties pointed out in this paper are common to most organization types, but are more or less acute depending on this type.

We organize the discussion considering two dimensions: process definition and process enactment. Finally, we conclude, putting out some key challenges to provide a better support to the modeling of a large scope of processes.

II. Processes definition

When starting process modeling, the first evidence is that, in most organizations, processes are not well defined. There are several reasons for that.

There is a lot of way to model the same process.

The same process can be defined in six steps in one department and in eight steps in another. This can be in two departments of the same company, in the context of a virtual team, or between two departments of two different companies, in the context of a virtual enterprise. Process modeling can be a way to rationalize these variations of the apparently same process. But, things are not so simple, even inside one company, and seem impossible in the context of a virtual enterprise, even if cooperation in this virtual enterprise is well established.

In addition, in most cases, it is not interesting to integrate all the processes that can be used within virtual organizations. It is sufficient to offer to processes the ability to cooperate despite their differences. We argument this position idea in the following.

Processes are often of uncertain development.

In most cases, processes are not defined *a priori* but incrementally, starting from an embryo which evolves depending on its interactions with its environment (clients, resources ...). Of course, the logic of a process can be refined based on previous experience. But the problem is not so simple because of the many variations of the apparently same process, as introduced above, and of the changing environment. We think that this is especially noticeable in temporary virtual enterprises.

We experimented it in the domain of building constructions: there have a lot of (textually formalized) process models to describe how they work, but these processes are abstract fragments of processes and each fragment can be instanciated in a lot of ways, depending on participants involved in the process and on methods used. In addition, how these fragments assemble is also not well defined. Environment can also influence the process: for instance, weather conditions can impose redesign or optimization of existing processes and the definition of new ones. That is also the case with embedded systems where assemblers

make smooth adaptations to their processes in order to correct the problems arising in the lifecycle of the system.

Some tasks can only be completely and definitively defined at runtime and a good approach seems to constantly improve the process by redefining and tuning it (instead of radically redesigning it). This is often the case in building construction, as they proceed with goals rather than with a very well defined sequence of steps. More generally, this seems a good approach when cooperation is not well established, especially in temporary virtual enterprises.

And this problem is still more perceptible when, in the context of a virtual enterprise, no physical enterprise is responsible for driving the main process [1].

Processes must deal with exceptions.

Even in domains where processes seem simple, we must deal with exceptions. This means simply that everything cannot be forecast. We experimented it when modeling the travel reimbursement system of our office (seats theoretically booked which where not, impossibility to respect the legal deadline,...).

In some cases, exception are very numerous. Sometime, we ask us if exception is not the rule in the domain of building construction. Architects often ask us to improve flexibility of predefined workflows in order to reflect the way they work. In fact, they often plan to do things according the process model, but once they use it, they deviate from this model. This can be because of unpredictable events (external events, intervention of new participants, time-out, failure of a participant,...), or simply because a partner wants to personalize its activities regarding its working style. This shows that, small, smooth adaptable([2]) and evolving process models are better adapted to such situations than rigid, well-structured and predefined huge models. And probably that this is a general rule for temporary virtual enterprises.

Processes are often creative activities.

We must constat that administrative and production processes are better defined than design, creative, interactive processes. This is due to the fact that design processes are related to human behaviors and that, in general, a lot of research work has still to be done to model these behaviors. We understood this difficulty first when modeling virtual teams in software processes [3]. We verified later how it is still more crucial in domain of co-design and co-engineering as in building construction. There, the rule is to work following fragments of abstract processes, generally corresponding to the different trades, and when a conflict occurs between two or more fragments, to organize a building site meeting to re-synchronize these fragments. Formalizing these meetings and their impacts on fragments is a big challenge.

More generally, we think that process model must not be too interventionist, but must support activities. The difference is that in one case, the user must follow a given process, and the other, he drives the process, and the workflow helps him in his decision, executing the more routine tasks but also helping him to handle exceptions, rollbacks,... This is also related to the relationship between coordination and communication (developed later) and more generally to the relationship to informal and formal fragments of processes.

Of course, the limit between formal and informal things is always moving. Developing new operators and new concepts as these introduced in [4] and which can apply to many cooperative situation would be very valuable, allowing to formalize behaviors which currently are not.

Processes are the richness of enterprises.

Processes are the richness of enterprises, and as a consequence, some enterprises are reluctant to make their processes visible outside. Especially, (some) small enterprises managers are quite convinced that their head is the better place where to lock their processes.

This explain why, in the context of virtual enterprises, the interface between two partners is only specified by contracts defining qualities of products and responsibilities in case of problems, but nothing explicit about the better way their process interact. However, "organizations interact by connecting their processes" [5], but this is not well formalized. Defining models of process interconnection in this context is a big challenge. Such model must be defined to minimize the impact on existing processes and to allow a partner organization to preserve its capital from outside observation and inquisition. Coming back to our above example in which two organizations model the same think in six and eight steps, the solution is not to choose one of the two process, or to define a new common process, but to find a minimum solution. [6], [7] define concepts and mechanisms to support interconnection of processes, but we think that some work has to be done at the level of process models. Operators of [4] can help. The problem is studied more globally in [8].

Process innovation is faster and faster.

Although initially uncommon and perceived as radical a few year ago, ideas like just-in-time inventory control and concurrent engineering have become accepted as best practice. These innovative practices have clearly been beneficial, but most organizations still need improvement... [9]. This means that in most organization, process modeling is not only an activity of formalization of the current state, but also an activity of process re-engineering. This can explain some resistance, even if process modeling seems a right way to support this acceleration of evolution of enterprise organization.

Formalizing processes is time-consuming

Most of small enterprises do not feel the need to formalize the way they work and do not have the resource to do this job. At the opposite, large enterprises seems ready to do this job for their internal organization.

III. Process enactment

A process engine is an advanced operating system which must synchronize activities accessing to resources by following the logic described in the process model and by incorporating human decisions in its scheduler. Such a system is complex for several reasons.

Actors already use application tools.

Processes are gross grain programs which coordinate elementary activities. These activities execute tools which are the preferred tools of the actors of the processes. A process engine must be able to work with existing tools : an architect which uses Autocad on PC will not easily accept to change its working environment. More generally, a process engine must allow to integrate most important operating systems and tools.

That is the usual problem of integration of legacy systems, data or tools. Problems in this section are dedicated to interoperability issues. Architects we are working with are very interesting in exchange formats of data, and some solutions already exist and are based on Step/Express. However, some work remains. For instance, some systems used in design departments are not those that architects found on building sites. Then, these techniques should also deeply improve informal communication process.

Most processes are distributed in space.

Actors contributing to the achievement of a process can be located on different sites. This implies considerations about security, availability, communication and scalability, as in traditional distributed systems. All these problems are complex and some are not always completely solved in general. We think that the context of process modeling can be used to simplify some of these problems. As example, exploiting some semantics of the process can simplify the management of replica what increases the scalability of the system [10].

From another point of view, if two distributed actors contribute to the achievement of the same process, distribution in space can imply to manage problems related to virtual presence.

Most processes are distributed in time.

Interacting entities, and especially interacting actors, contributing to the achievement of a process are not necessary (always) present at the same time. This is a problem when it is needed to establish a connection between two entities and one of them is not currently present.

Process engines must manage the relationships between coordination and communication.

The two last items indicate a relationship between process modeling and cooperative work : process engines must manage the relationships between coordination and communication. In fact, it is not realistic to believe that everything in a process can be formalized as activities of a workflow. Sometimes, the workflow has to interact with humans and in some situation, computer support can be limited to providing a communication media between actors connected to activities. One can protest that communicating as just defined is not a problem of process. Unfortunately, things are not so simple. Interaction between coordination and communication exists at different points of a communication : a communication can be triggered by a process engine; when communicating, actors can execute activities specified in the process, but without control of the workflow engine; when communicating actors can take decisions which will influence the continuation of the process, even to "abort" the current process fragment and to "start" a new one... This makes again reference to a model of process in which process fragments are assembled during meetings. The challenge here is to define a concept of electronic meeting as a component of a process model. Another example of interaction between coordination and communica-

tion is the usage of coordination information to enhance awareness.

Process engines are non classical transactional systems.

Considering a process as a partially ordered sequence of transactions seems an attractive way to simplify the problem of concurrency (each activity of the process is encapsulated in a transaction). The problem is that not all activities of processes are abortable (as required by the traditional ACID transaction theory), nor compensatable (as required by most advanced models) [11], nor isolated [4](as required by the traditional ACID transaction theory). New theories have been defined : the extended unified theory [11] in the continuation of [12] or COO [4] for non isolated transactions. But new models still request to relax the stringency of most classical correctness criteria and to provide more flexible recovery methods and more flexible interactions between transactions.

Processes can span several organizations.

We underlined previously the problem related to the modeling of process interconnection in the context of virtual enterprises. The problematic related to virtual enterprise is more largely described in [13]. At the system level, the objective is to provide mechanisms, on one hand to connect processes, on the other hand to provide confidentiality and autonomy to partners. Normalization groups contribute to the first point [6], [7] but in the limited context of workflow systems. Concerning the second issue, process engines must be provided with an efficient firewall (not too laxist nor too restrictive). Until such an efficient firewall is not provided, organizations will not accept external entities to work in their limits, even in the case of well-established virtual enterprises. Of cause, this increases also the problem of interoperability introduced above : enactment platforms must be able to interoperate in some way [8].

IV. Conclusion

This paper is a preliminary report based ongoing experiences. Nevertheless, we are convinced that process modeling complexity increases depending on :

- the level of creativity it involves,
- the monolithic nature of the process (depending if the process is defined as a whole or as the result of the cooperation of process fragments),
- in case of multi-fragments processes, the nature of process cooperation (well established or temporary).

The more simple organizations to model are the administrative workflows internal to a company, the more complex are the creative processes of temporary virtual enterprises composed of small enterprises. The key challenges to support a larger set of processes are :

- to monitor the relationships between coordination and communication, as introduced above,
- to develop models for dynamically interconnecting fragments of processes.

V. Acknowledgement

We acknowledge partners of CTI CNET no 98-1B-124 concerning cooperation models in general and their applications to building construction in particular, and people working with us in the context of embedded systems, for their contribution to this work.

References

[1] G. alonso, U. Fiedler, A. Lazcano, H. Schuldt, and N. Weiler, "WISE: Business to Business E-Commerce," in *IEEE RIDE: Research Issues in Data Engneering (Virtual Enterprise)*, Sydney, March 1999.

[2] Amit Sheth Yanbo Han and Christoph Bussler, "A Taxonomy of Adaptive Workflow Management," in *CSCW-98 Workshop, Towards Adaptive Workflow Systems*, 1998.

[3] N. Belkhatir, A. Carzinagan, J. Estublier, E. Di Nitto, C. Godart (ed), J. Janke, P. Lago, W. Shaefer, and H. Skaf, "Cooperation control in PSEE's," in *Software Process: Principles, Methodology, Technology*, J.C. Derniame and B. Warboys editors, Eds. LNCS 1500, 1999.

[4] C. Godart et O. Perrin et H. Skaf, "COO: a workflow operator to improve cooperation modeling in virtual enterprises," in *9th IEEE RIDE, International Workshop on Research Issues in Data Engineering, Sydney*, March 1999.

[5] T Malone, "Managing processes in the networking economy," in *WACC'99, Keynote talk*, 1998, p. .

[6] Workflow Management Coalition, "Workflow Standard - Interoperability," Tech. Rep. WFMC-TC-1012, Version 1.0, Workflow Management Coalition, october 1997.

[7] Object Management Group, "Workflow management Facility," Tech. Rep. bom/98-06-07, Object Mangement Group, july 1998.

[8] CrossFlow, "Esprit Project No 28635," Tech. Rep., http://www.crossflow.org/.

[9] T.W. Malone and al., "Tools for inventing organizations: toward a handbook of organizational processes," rapport technique http://ccs.mit.edu/ccswp198, Center for Coordination Sciences, MIT, 1988.

[10] G. Canals, P. Molli, and C. Godart, "Support for end user participation using replicated versions and group communication," *ACM SIGGROUP Bulletin*, 1999, To appear.

[11] H. Schuldt, G. Alonso, and H-J. Schek, "Concurrency control and recovery in transactional process management," in *ACM Symposium on Principle of Database Systems (PODS'99)*, May 1999, pp. 1–11.

[12] R. Vingralek, H. Hasse-Ye, Y. Breitbart, and H.J. Schek, "Unifying Concurrency Control and recovery of transactions with semantically rich operations," *Theorical Computer Sciences*, vol. 190, pp. 363–396, 1998.

[13] D. Georgakopoulos and L. Maciaszek, *Proceedings of 9th International Workshop on Research Issues in Data Engineering (Virtual Enterprise)*, IEEE Press, 1999.