

Towards an improved steering of design process in architecture

A. Laaroussi & A. Zarli

CSTB, Sophia-Antipolis, France

J.C. Bignon

CRAI UMR MAP, Nancy, France

ABSTRACT: This paper presents a modelling of Design Process in Architecture: DPA. The goal of this modelling is to control the DPA. This modelling of DPA is based on a process-oriented approach of DPA. It is extended by a suggestion of a software architecture to steer DPA. Thus we State the foundations of a future tool, a kind of "manufacture software" able to evolve according to the requirements of each actor in DPA.

1 INTRODUCTION

Nowadays, the actors of the design in architecture are facing the increasing complexity of projects (e.g. shorter delays, increasing number of actors involved in the design process, high quality requirements, variety of technologies, etc). The context of the project leads to this great complexity. To ensure the control of this complexity, it seems important to provide the actors with a software solution able to easily and coherently represent the context of the project.

Moreover, contemporary architecture projects involve a large amount of geographically dispersed enterprises (and especially SMEs) that participate in the elaboration of an unique project. Thus, the integration of the viewpoints of all the actors of the design phase by taking into account the building life-cycle, suggests a certain degree of the parallelism for design activities. This is made possible by taking into account, very early in the process, the constraints and the parameters managed much later in traditional organizations.

Through the analysis of the context and rules described by French regulations, the researchers have defined several "functional" types of activities (Guerriero. 2002). They have also identified the processes at explicit macro-level: the phases are divided into tasks. In this paper, we focus on the modelling of these explicit macro-levels of design in architecture.

The organization of the paper follows. Section 2 introduces the need of understanding a DPA as an unique process that is time-limited. Next, in section 3, we discuss the main concepts used to model a

DPA. Section 4 presents a software architecture to steer a DPA. Finally we conclude with ongoing and future works.

2 DESIGN IN ARCHITECTURE AS UNIQUE AND TIME LIMITED PROCESS

If the first objective of a design process is still to define an answer to an unsatisfied need, Poveda & Thorin (2000), design in architecture underlies a targeted objective, represented in most cases by an object to be conceived, more or less accurate in its initial definition. The evolution of the designed object is sequenced by a whole set of stages and phases, not necessarily linear, and consequently tends to some degrees of certainty which are not absolute.

Whatever the size of the project; any operation of construction is characterized by temporary associations of complementary skills. The aim of the actors is the achievement of the project, but with not very explicit and poorly codified practices. Limited by the one-of-the-kind nature of building projects, current practices are a logical consequence of the diversity of actors, documents and stages of the design - realization process.

In addition, and contrary to the repetitive processes, design in architecture is always to redefine (Hanser et al. 2003). Indeed, the uniqueness of the projects implies an identification of the particular activities of each project. Thus, one of the project manager's tasks is to precisely define the work to be achieved, by taking into account the objectives and constraints. Moreover, the risks are part of any de-

sign project and represent sometimes a raised degree of unpredictability, whereas in repetitive processes, the continuous improvement minimizes uncertainty.

To manage such a "transverse" approach for design in architecture, it seems important to use a process-driven approach. It leads to decrease barriers among actors by unifying their focus on the main objectives of the project. Thus a process-based approach of design in architecture allows to:

- decompartmentalize the businesses and support the co-operation among actors,
- apprehend the design team like a single system and not like an aggregation of functions, services, businesses, responsibilities...
- unify the focus on the main objectives, and
- improve management of the interfaces among business processes

3 A STEERING-DRIVEN METAMODEL OF DESIGN PROCESSES IN ARCHITECTURE

We propose in this section a metamodel allowing to answer to various constraints associated to design in architecture steering. It meets businesses needs of:

- *Coordination*: it will be made by the perception of events generated by communication among actors and the actions relating to the instance of objects environment (e.g. work, document, actor or activity).
- *Communication*: it is about the processes and the building during design, based on associated structured messages. The contextualization of the messages allows each actor to identify the situation of the project at the time of the communication (located action).
- *Decision-making* concerning the processes (e.g. allocation of tasks and roles, costs, deadlines and conflicts management). This decision-making relies on a negotiation model based on typified messages associated to the environment objects.

In the other words it is meant to:

- Allow the whole perception of design processes by ascertaining its dynamics (e.g. the dashboard concept). Thus the project manager will be able to make the adequate decisions to reach the expected performance.
- Memorize and channel the dynamics produced during the design process and which consequently influence the future actions.
- Provide the processes or sub process state at any moment of the project in design.

- Quantify the achievement of the objectives of the architectural project in design.
- Assist the anticipation of the dysfunction related to the processes.

3.1 General sight of a metamodel of DPA modelling.

In this section we present a steering-driven metamodel of DPA (fig. 2). It is based on the co-operation metamodel from CRAI team (Kubicki et al. 2005). The metamodel highlights the main concepts and their relations and aims at modelling the majority of DPA. In addition, our metamodel gives a general framework, leaving to the actors a degree of improvisation and adaptation according to the requirements and businesses.

3.2 Definition of the various concepts

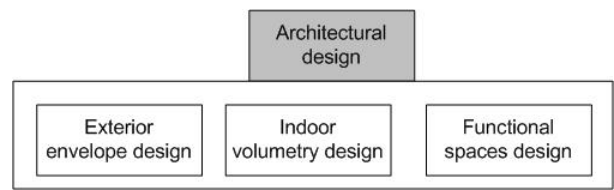
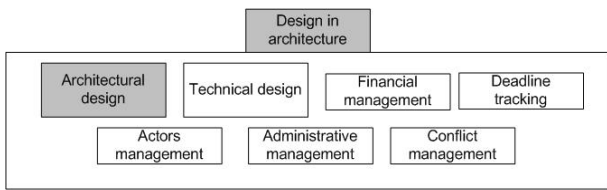
We focus on the definition of concepts that allow to steer the DPA.

The concept of process: it concerns a whole of elements, eventually heterogeneous, which are in interrelated. The metamodel shows the elements setting the system: activities, actors, output data, resources...

The concept of objective: it concerns a goal to achieve or the aims of a process or an activity. It allows to highlight the points to improve for this process or this activity. It is thus the expression of finality.

The global process: it concerns a process representing the objective to achieve and possibly a splitting up into processes. Indeed, sometimes, we are satisfied with identifying a process as a closed box, without its content. For example, at the beginning of a building design, one establishes a cartography of the business processes described only by their objective (e.g. architectural design process, technical design process, financial management process).

Moreover, a global process can be composed of detailed processes. In short, a global process is described by its objective, and sometimes by the processes which make it up, without using other elements of the system. In (fig. 1) we aim at highlighting a second level in the representation of a strategic process. The process of architectural design is thus splitting up into other finer processes.



Process	Objective
Exterior envelope design	To insert building in the site
Indoor volumetry design	To organize indoor volumes according to building envelope
Functional spaces design	To distribute functions according to spaces

Figure 1 example of splitting up of a global process of design in architecture and its objectives

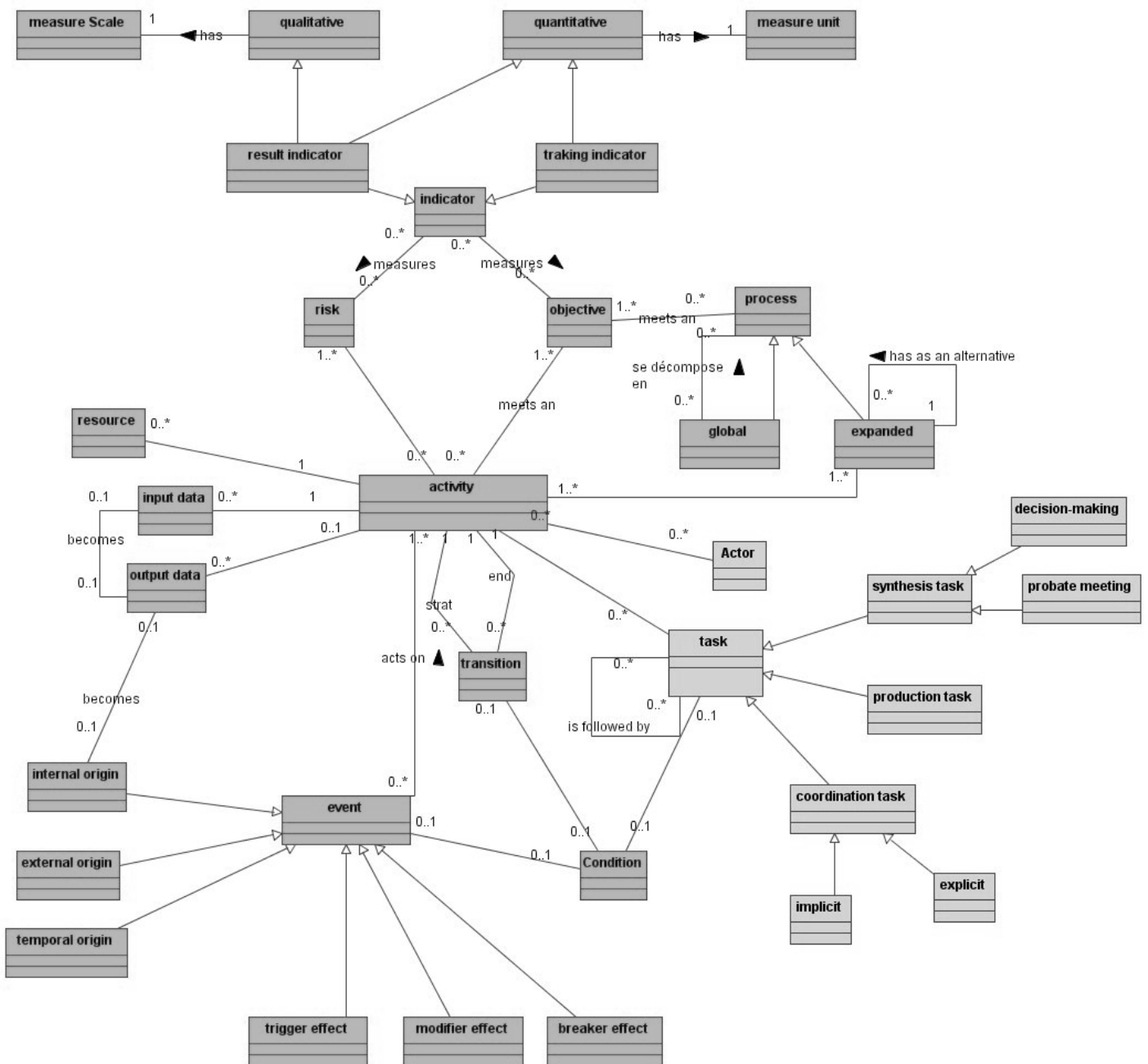


Figure 2 a metamodel for modelling design processes in architecture - UML formalism

The expanded process: is the process represented by its dynamics, in particular its activities and its transitions. Thus one describes the contents of the system by highlighting the detail of its work. Contrary to the global process, the expanded process is described initially by its activities. Thus it can be defined, like a whole of interdependent activities to achieve an objective. The same process instance can be the subject of a double specialization: first like a global process, composing a higher level process, then like an expanded process. To ensure the coherence of the levels of representation, a process being the subject of detailed description should not be split-up any more into a process. Indeed the latter would be on the same level as the activities. Afterwards, we can split-up a global process into several processes. Some of them could be kept at the global level and others be the subject of a detailed representation.

The transition is a concept representing the directed link between two activities. It expresses a sequence constraint and is always associated to two activities. However, the predecessor activity and the successor activity can be the same: a loop is then represented, and the transition has to be supplied with a condition. In addition if the transition is not under a condition, the sequence is mechanical: the end of an activity starts the following one.

The event: concerns a stimulus inducing a reaction in an activity. The actor responsible of the activity has to be informed that the event occurred. This is why we materialize the event by information. Moreover the event doesn't represent work, therefore the arrival event does not consume any resource of the process activity. On the other hand it can have an author, who is at the origin of the event. Thus we distinguish three types of events:

- *Temporal event*: it corresponds to an achievement of a due date (e.g. date, frequency or past delay).
- *Internal event*: it corresponds to a decision taken by an actor. The actor can be the person in charge of the activity or another actor,

playing a role in the same process or in another process.

- *External event*: it can be a decision taken by an external actor or the information that he sends (e.g. a denial of building permit by the mayor, user modification request aborting project). One has little consolidation on an external event

In addition, an event can play various roles towards an activity:

- It can be a *trigger*. An activity can be started by several events, often exclusive, sometimes concomitant (e.g. required data validated, required tasks completed). According to the event, the activity could be performed by specific tasks.
- It can be a *“breaker”* and causes the activity abort (e.g. conflict not solved, ending task)
- It can be a *modifier* and operate on the ongoing activity (e.g. modification request, a technical conflict appearance).

The indicator: it is a measuring instrument using quantitative or qualitative values. It allows the assessment of an objective achievement. It uses:

- A required value to answers to the objective
- An actual value provided from external data or from a calculation (e.g. defined in a rule of calculation).
- An alarm level. It is a value from which the objective must be regarded as unfulfilled

In addition, the indicator can be of two natures:

- A result indicator to measure the final result of the action and to allow establishment of an *a posteriori* report
- A tracking indicator to be used as an action milestone and to allow to measure its progression. Thus the users will be able to anticipate, and react by corrective measures, before the result “is consumed”.

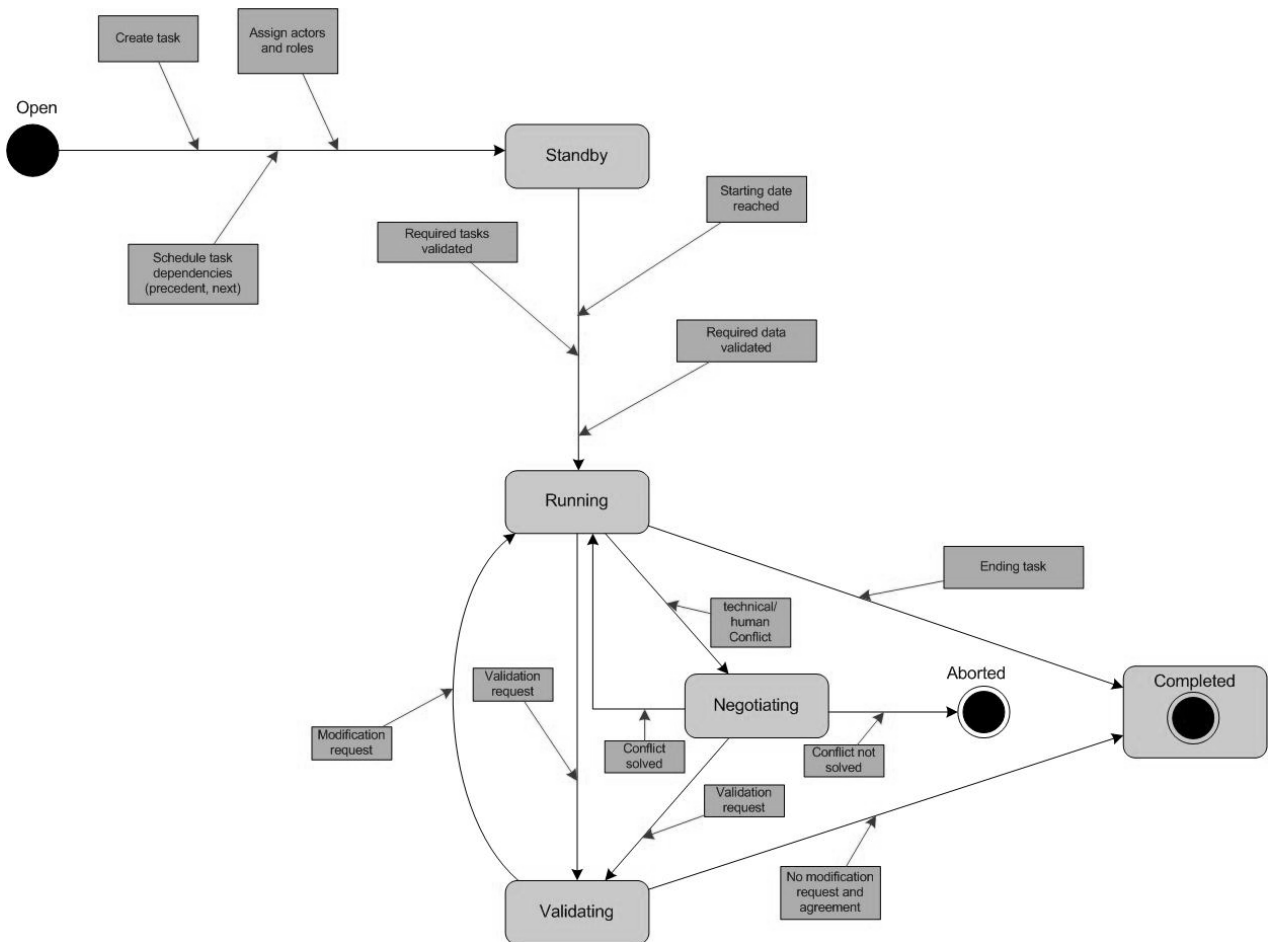


Figure 3 example of a production activity states and events inducing activity state change

3.3 From DPA modelling to its steering

The presented metamodel aims at allowing the DPA formalization. This formalization should enable to react to the DPA constraints. This search for constant improvement results in the steering of the design in architecture. This steering allows the DPA to evolve toward an aim initially fixed, and via various intermediate situations (fig. 4)

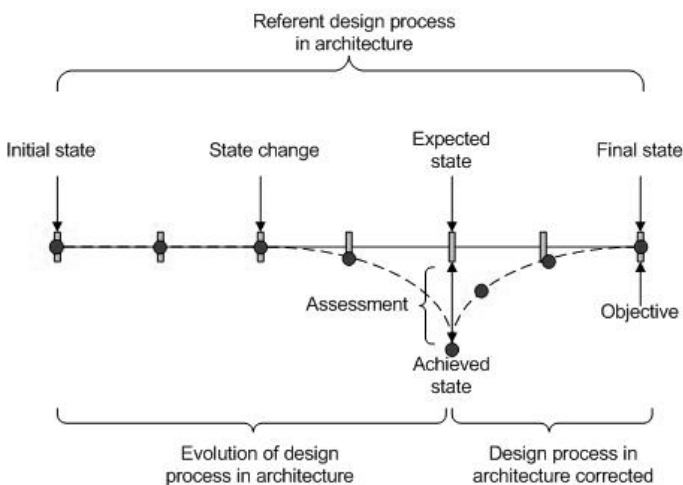


Figure 4: a principle of steering design in architecture

This vision allows to apprehend the steering of the design in architecture according to four stages

implementing various functions (fig. 5). These stages follow:

- *Modelling stage*: it builds the referent process able to achieve the settled goal, by defining the activities and objectives to be achieved.
- *Execution stage*: it executes the referent process and allows the detection of faulty operation based on the achieved result and on the expected objective.
- *Correction stage*: it stops the ongoing process, then, corrects the problems by defining a new referent process
- *Termination stage*: it assesses the process on an overall basis when its final state is reached.

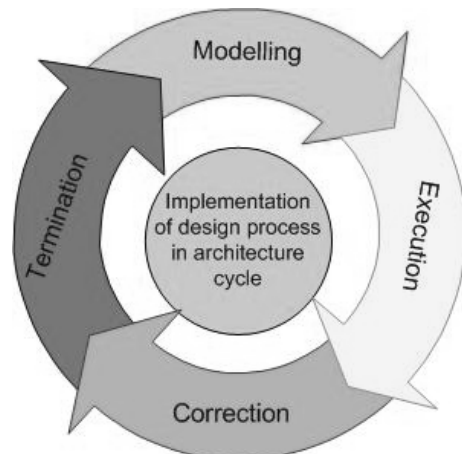


Figure 5 various stages of steering

After the modelling framework presentation, we introduce to the implementation of a software environment driven architecture.

4 A SOFTWARE ARCHITECTURE DRIVEN SERVICES FOR THE DPA STEERING

This research aims at developing an environment able to integrate various software applications. This environment aims to run and to steer the DPA. Thus we propose to base our future tool on a Services Oriented Architecture SOA with three modules (fig. 6).

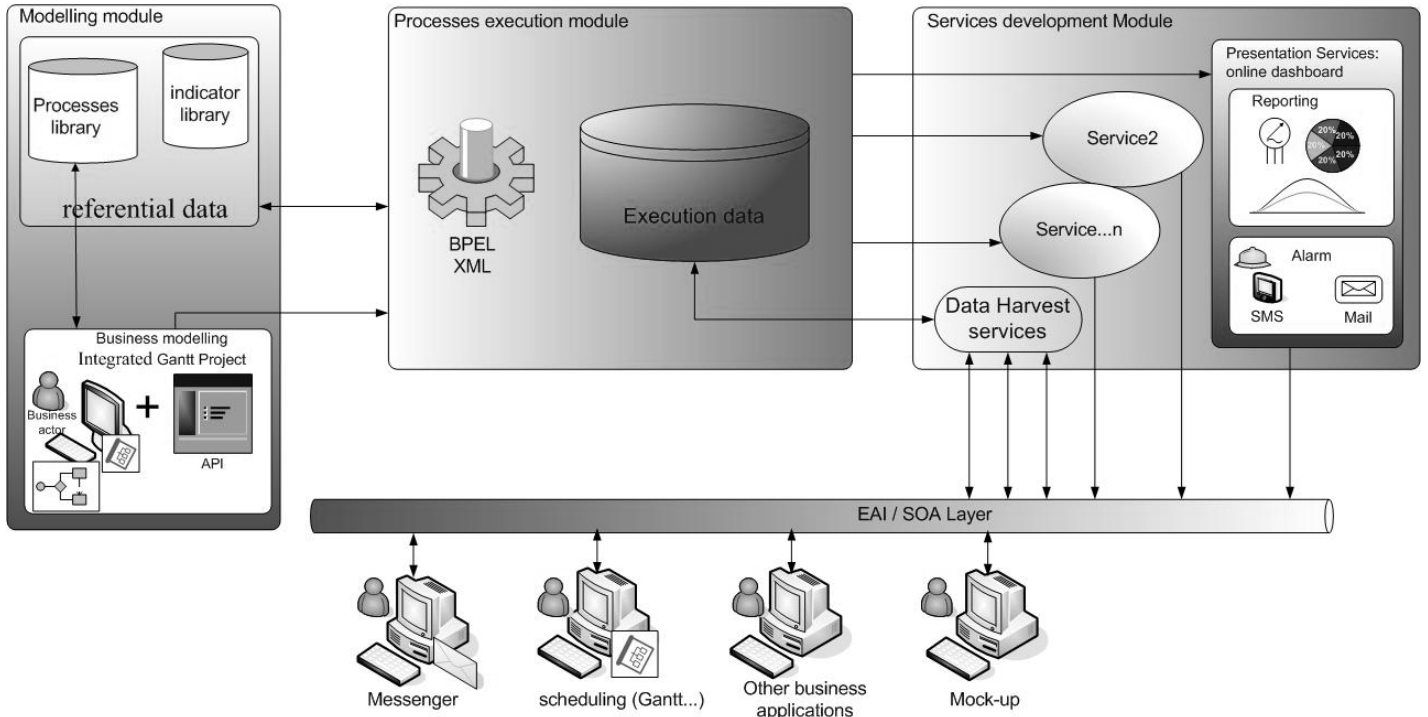


Figure 6: a software architecture to steer a DPA

Modelling module: it is a module based on a graphic tool easy to handle by the business actors. It allows them to get their business processes (e.g. tasks and their sequences, events, constraints, etc.). The modelled business processes will be then coordinated by the DPA manager. To assist the business actors in this modelling, we propose a business processes library based on the experience of the actors. These processes can be customized (e.g. customized transition rules, missions, roles, events, tasks, etc.). Thus the user could draw from the processes libraries in order to customize the ongoing process.

Processes execution module: it is a module allowing the processes transformation towards an executable model, like BPEL (Business Process Language Execution) (Andrews et al. 2003) or XML. This happens with the implementation of an execution engine and extension mechanisms of this engine by using Aspects-Oriented Programming AOP. Bachmendo and Unland (2001). Thus, we propose the installation of a dedicated engine based on a lower layer of events oriented integration (e.g. Event Driven Architecture EDA). This will allow collaboration among various actors. These actors can be, initially, human and in the second time, resources

data-processing (e.g. business applications, type of scheduling, PDM, mockups, etc.)

Services development module: it is a quality and performance supervising service of the DPA (e.g. dashboard, alarm, data harvest services, etc.). It is thus a question of installing the tools for collaborative services generation. These tools for automatic generation will be based on the expertises and already existing tools like “manufacture software”. (Parigot et al. 2002)

5 CONCLUSION

To sum up, we have presented an approach to model a DPA in order to steer it. This approach aims at ensure the well-off development of the DPA model by adding new actions and entities. Then, it will allow the flexible integration of professional tools into the execution engine in the form of services (e.g. visualization, coordination, evaluation tracking, etc.). The variety of modules forming the software architecture can be developed according to the needs of each actor. The inter-connexion with the execution engine has to be independent from the implementation in order to allow the parameterization by final users.

While having this stages ongoing realization, we intend to rely on it in order to allow:

- Modelling a business steering of the design in architecture by taking into account the coordination, communication and decision-making.
- Setting a type MDE design approach allowing a great flexibility of evolution for the design phases and the implementation of the produced applications.
- Validating the business model and MDE approach by a relevant experimentation on selected scenarios.

ACKNOWLEDGMENT

The authors would like to thank Gilles Halin for the fruitful discussions.

REFERENCES

- Andrews et al. 2003. Business Process Execution Language for Web Services version 1.1. Technical report, BEA, IBM, Microsoft, SAP, Siebel Systems.
- Bachmendo and Unland (2001). Aspect-Based Workflow Evolution. In Tutorial and Workshop on Aspect-Oriented Programming and Separation of Concerns. Lancaster, UK.
- Guerriero, A 2002. Étude de la coordination dans la coopération entre acteurs au cours de la conception d'un bâtiment. Mémoire de DEA, Modélisation et Simulation des Espaces Bâties. Université Henri Poincaré, Nancy. Pages : 23-31
- Hanser et al. 2001. Relation based groupware for heterogeneous design teams. In Conference: Education in Computer Aided Architectural Design in Europe. PP. 86-91. Helsinki, Finland.
- Kubicki et al. 2005. Digital assistant for the cooperative construction process in AEC. July, 19-21 2005. CIB W78 Conference. Dresden, Germany.
- Parigot et al. 2002. Aspect and xml-oriented Semantic Framework Generator: Smart Tools. In M. van den Brand and R. Lämmel, editors, ETAPS'2002, LDTA workshop, volume 65 of Electronic Notes in Theoretical Computer Science (ENTCS), Grenoble, France. Elsevier Science.
- Poveda & Thorin, 2000. use of scenario to integrate cooperation in design. COOP2000 Conference. In "the use of theories and models in designing cooperation systems". 23-26 may, Sophia-Antipolis, France