# A Holistic Approach in Enterprise Business Process Modeling

Nancy Alexopoulou
Department of Informatics & Telematics
Harokopio University of Athens
Athens, Greece
nancy@hua.gr

Mara Nikolaidou
Department of Informatics & Telematics
Harokopio University of Athens
Athens, Greece
mara@hua.gr

Dimosthenis Anagnostopoulos
Department of Informatics & Telematics
Harokopio University of Athens
Athens, Greece
dimosthe@hua.gr

***Abstract*—Business process automation is usually engaged to enhance the operation and productivity of an organization. There is a wide variety of methods and tools supporting well-structured business processes, modeled as a sequence of discrete activities. However, an organization may comprise processes of diverse nature, which cannot be efficiently represented by the same process model. Efficient modeling of the overall enterprise functionality, dictates the identification of the appropriate modeling approach for the description of each business process. An integration framework, ensuring a seamless and effective intercommunication between the discrete process models, is also required. In this paper, a categorization of diverse business process modeling paradigms is presented, while an integration framework, constituting a holistic approach in business process modeling, is proposed. The framework is based on the concept of viewing the organization as a system, consisting of a hierarchy of independent sub-systems. Implementation challenges are also discussed.**

***Keywords-*** business process model; business process modeling paradigms; model integration; agile enterprise; system-of-systems enterprise model

## I. Introduction

Nowadays, business process automation is usually accomplished through the utilization of process-aware information systems [1] based on BPMS (Business Process Management System) technology and explicit process models developed according to a suitable modeling language. Most of the currently available BPMSs support the automation of well-structured processes comprising a clear and stable sequence of activities. However, in practice, an organization may comprise processes of diverse nature. A hospital, for example, apart from typical well-structured processes, such as the procedure of patient admission and the procurement of medical stuff, comprises also patient treatment processes. In the latter, the activities performed are strongly based upon human decision influenced by the circumstances as well as unpredicted contingencies. As a result, developing a clear and stable structure for this kind of processes is not a trivial task. In fact, is almost unfeasible to create a traditional model for the description and ultimately the automated execution of such processes. In addition, doctors cannot work efficiently if they are obliged to apply a prescribed treatment procedure; rather they prefer to have the flexibility to adjust the treatment as required, according to their evaluation. It is entailed that the diverse nature of business processes requires the enterprise to be engaged in supporting a different business process modeling approach for each department or procedure.

The deployment, however, of isolated business process models may cause inefficiencies and integration problems in the organization. For an efficient modeling of enterprise functionality which will allow a seamless automated execution of the diverse business processes, what is required is first, the identification of the appropriate modeling approach for the description of a business process and second, an integration framework for a holistic approach in the design and execution of enterprise functionality.

The purpose of this paper is, first, to stress the need for organizations to support multiple, diverse business process models, and second, to propose a framework towards this direction enabling their seamless integration. The concept of treating the organization as a system, further decomposed into a hierarchy of discrete sub-systems is introduced. Such decomposition may consist of either independent units (for example a department) or procedures, better described by a specific process model. More specifically, the need for supporting multiple business process models is elucidated in section 2. Section 3 presents a viewpoint-oriented categorization in an effort to create a clear picture regarding business process modeling paradigms. Section 4 illustrates an integration framework for the diverse enterprise processes, which is based on the concept of system, constituting a holistic approach in business process modeling. The proposed integration framework is explicated in Section 5 through a simple example. Implementation challenges and future work are discussed in section 6. Conclusions lie in section 7.

## II. ENTERPRISE AGILITY

Modern enterprises operate in highly turbulent environments and as such more than ever before they need to be able to efficiently respond to change by adjusting their processes in a timely and cost-effective manner. Apparently, such capability presupposes the adoption of agile business process modeling approaches for the design of the corresponding actual processes. Such approaches would enable the creation of agile models, which can be efficiently modified whenever needed. But what is required for the creation of an agile model? Obviously, as mentioned above, using a typical activity-driven approach to describe processes with unclear structure would produce stiff models, lacking the desired flexibility. On the other hand, a real process comprising well-defined steps can be efficiently described simply using a common activity-driven model. Any effort to use other paradigms (e.g. a data-driven approach) for such a well-structured process may increase complexity in its design and consequently hinder agility, since modifying a complex model is usually time-consuming.

This discussion implies that the crux of the matter is to select the right approach for the creation of a business process model, based on the features of the respective actual process. In particular we state that:

*to create an agile business process model, the selected modeling approach should be in harmonization with the nature of the actual business process to be modeled.*

If such harmonization is not attained, a change in the business process model may become an arduous procedure falling short of the desired flexibility. Thus, it is crucial to adopt the correct approach for the representation of a business process.

Taking into consideration that organizations comprise processes of diverse nature, as explained above using the hospital as an example, what is required in order for the enterprises to be deemed agile is the ability to support multiple business process models. It is this capability that prompts for a holistic approach in business process modeling that will enable the integration of the diverse models. If the process models cannot efficiently intercommunicate, enterprise functionality will be broken into isolated silos causing delays and inefficiencies whenever a process requires feedback provided by other processes.

## III. A VEWPOINT-ORIENTED CATEGORIZATION OF BUSINENES PROCESS MODELING PARADIGMS

Choosing the right modeling paradigm for the design of a business process is not always a trivial task. Often, it requires a deep understanding of the nature of the business process to be modeled, as well as a thorough knowledge of the different paradigms. To contribute in the evolvement of a clear and complete picture regarding the modeling paradigms that would help the interested parties in selecting or developing the appropriate business process modeling approach, we propose in this section a viewpoint-oriented categorization of the existing modeling paradigms.

In the majority of business process modeling approaches found in the literature ([2], [3], [4]), a business process is described through the definition of activities, data (or resources in general), events (often implicitly through the control flow) and roles and, of course, their interrelations. Therefore, we consider *activities*, *data*, *events* and *roles* as the primary structural elements of a business process model. As such, we state that business process modeling approaches can be classified in *activity-, data-, event-* and *role-driven*.

All approaches comprise directly or indirectly information regarding activities, data, events and roles but, as denoted by the suffix 'driven', the respective constituent *drives* the design procedure, so it is regarded a modeling viewpoint. As a rule-of-thumb, consider that an approach is activity-driven if activities are the first entities identified. Likewise if data entities are identified first, the approach is data-driven, and so on. In the following, each modeling paradigm ensuing from the viewpoint-oriented categorization is more analytically discussed. For each paradigm, also a number of representative existing approaches are mentioned.

### *A. Activity-driven Paradigm*

Most business process modeling approaches are activity-driven and concern well-structured business processes. The reason is that well-structured processes (e.g. manufacturing processes) were the first to be automated as they have well-defined steps. Such approaches use UML-like activity diagrams [5] for the representation of business processes. Basically, the objective of activity-driven business process modeling is to identify the actions of a specific functionality context and combine them appropriately in a process graph so that a goal is satisfied. In that sense, emphasis is laid on *how*. Each action is then associated with the corresponding events, roles and data.

A typical example of activity-driven business process modeling is that dictated by Aris framework [3], which explicitly models all aspects of a business process. Events depict the initiating and completion conditions for an action execution. As such, business processes have the form of event-action chains where an action is always followed by an event and vice versa. However, in most approaches such events are omitted as they are regarded an overhead rather than a useful entity that should be explicitly represented. In ADEPT [4], for example, events are implied through the defined control flow. ADEPT is an activity-driven approach that produces executable business process models. Actions correspond to tasks defined in an ADEPT model. BPEL [6] (Business Process Execution Language) and BPMN [2] (Business Process Modeling Notation) are two other popular activity-driven business process modeling approaches. The activity-driven paradigm is mainly eligible for cases where the actors should be enforced to follow a specific flow of steps. This is the case for example with the public processes which comprise specific steps that should be followed by the citizens.

### *B. Data-driven paradigm*

While activity-driven modeling is characterized by a complete and rigid process logic, in data-driven modeling, process logic is more loosely and partially defined. This is why data-driven modeling is more suitable for cases where the business process graph is extremely complicated and thus cannot be easily depicted. Such complexity may stem, for example, from multiple nested conditions or multiple reverses to the same prior actions causing a chaotic structure.

Data-driven approaches focus on identifying initially the data entities managed within a specific functionality context, i.e. emphasis is given on *what*, not on the specific process steps followed. When something happens, it is depicted as change to the current state of one or more data entities. A state of a data entity is usually represented through a set of variable values.

Based on the data entities identified, the definition of actions, events and roles is subsequently conducted. Actions are specified relative to the data entities as procedures that either cause changes to data states or require the presence of specific data objects in order to be executed. Roles are associated with the identified actions. Events correspond to the data states identified. Usually, data-driven processes are represented through state charts [5]. A data-driven approach could be suitable, for example, in case of a process that concerns the design and development of a product. Muller et al. [7] propose a data-driven modeling approach and show how it can be applied in the automotive industry.

### *C. Event-driven Paradigm*

When modeling organizational functionality, there may be cases that an action is not initiated due to data modifications or because an action sequence has to be followed. Rather it is initiated because something happened that needs to be handled somehow. In such cases, the conditions under which an action should be initiated can be expressed in a more abstract manner through events. Such conditions may arise from data modifications, human decisions, timing states or anything that could lead to a situation that should be handled, which can even be of an unknown source. Something that happens signifies an *event*. An event of unknown source may be defined in a model in case it is meaningful for the organization, which means that its occurrence should be handled somehow, e.g. the sudden fall of the stock market may initiate a number of actions despite the fact that what caused the fall may be unknown.

The event-driven modeling procedure starts with the identification of meaningful events that occur in the organizational environment. Apparently, in such a case, events are not defined in a complementary fashion to depict the starting and terminating conditions of an activity but rather are those identified first and determine the actions that should be performed upon their occurrence. In that sense, an event denotes *when* (not necessarily in terms of time) an activity should be initiated. Roles and data are associated with the identified actions. Events have been mainly used in ECA (Event-Condition-Action) model [8], originally used within the active database community [9]. In ECA-based approaches ([10], [11]), design involves the definition of ECA rules. An ECA rule dictates that when an event occurs, a condition is evaluated. If the condition is satisfied, the respective action is executed.

Lastly, it should be noted that the event-driven paradigm inherently supports the description of processes that are affected by unpredicted contingencies. Since contingencies are unpredicted events, following an event-driven approach would better facilitate the incorporation of the new events in the current model.

### *D. Role-driven Paradigm*

While not common, there are also role-driven modeling approaches. In role-driven approaches, modeling begins with the identification of roles i.e. actor categories, involved in a specific functionality context. In that sense, emphasis is laid on *who*. Each role is then associated with the activities that performs, the data that manages, and the events handled or generated by him/her. Interconnections between roles form the broader business process.

A role-driven business process could be described by the UML communication diagram [5], which is used to depict interactions between entities. Obviously, role-driven approaches focus on specifying interactions between roles. As such, they are suitable mainly for modeling communication-based processes, e.g. B2B process.

A role-driven approach has been proposed by Balabko et al. [12]. According to the authors, modeling focused on roles facilitates the modifications of business process models when changes happen in roles as a result of regulatory, technical, or social changes in an organizational environment.

### *E. Categorization Summary*

TABLE I. MODELING PARADIGMS AND THEIR FEATURES

| **Modeling Paradigms** | **Features** |
|---|---|
| Activity-driven | ▪ emphasis is laid on *how*<br>▪ the target of modeling is the process flow<br>▪ modeling begins with the identification of actions<br>▪ mainly suitable for well-structured processes |
| Data-driven | ▪ emphasis is laid on *what*<br>▪ the target of modeling is data management<br>▪ modeling begins with the identification of data entities<br>▪ mainly suitable for processes of vague structure |
| Event-driven | ▪ emphasis is laid on *when*<br>▪ the target of modeling is event handling<br>▪ modeling begins with the identification of meaningful events<br>▪ mainly suitable for dynamic processes that are affected by unpredicted contingencies |
| Role-driven | ▪ emphasis is laid on *who*<br>▪ focus is on interaction between roles<br>▪ modeling begins with the identification of roles<br>▪ mainly suitable for communication-based processes |

Basic characteristics, scope and suitability of the four modeling paradigms are summarized in Table 1. Since each of the identified paradigms focuses on a different process aspect, it is clear that it is best suited for a discrete process category, though it might be used to represent others processes as well. In this case, the resulting process model though accurate, is not the most efficient one, especially when constant adaptation is needed. It should be noted that, theoretically, hybrid approaches may ensue from the combination of paradigms. However, even if two constituents are emphasized, in practice one single would drive the modeling procedure, according to the proposed viewpoint-oriented framework.

## IV. On Establishing a Business Process Model Integration Framework

Most researchers propose models that target a specific business process category, most often using the typical activity-driven paradigm. However, as discussed earlier, an enterprise can gain agility, only if it supports multiple business process modeling paradigms so as to achieve alignment with every different nature of the business processes it possesses. The deployment, however, of isolated business process models may cause inefficiencies and integration problems in the organization. To overcome this problem, following other researchers ([13], [14], [15], [16]), we consider an organization as a system comprising subsystems characterized in an abstract manner as *units*. Units may be a logical grouping of specific functionality or correspond to physical segments of the organization. Each unit can be analyzed into other units and so on. As such, an organization may be ultimately decomposed to a unit hierarchy, which in essence constitutes an allocation of the organizational functionality. The functionality encompassed by atomic units, i.e. units of the lowest level that are not further decomposed, is described through the definition of business process models. A unit, in order to be atomic, should be governed by homogeneous process logic so that its functionality can be effectively described according to a specific business process modeling paradigm, so that flexibility may be achieved. Otherwise, the unit should be divided until homogeneity is reached.

The organization system has well-defined boundaries. These boundaries demarcate the organization within its *ecosystem*, which comprises entities that interact with the organization such as, customers, suppliers, partners, etc. This interaction can be represented through input and output interfaces, called *ports*. As such, the organization may both receive messages from its ecosystem using its input ports, as well as diffuse to it internally generated messages using its output ports, as dictated by a system-of-systems view of the organization and the procedures supported ([16]). Messages coming from the ecosystem are propagated down to the unit hierarchy and are ultimately consumed by the atomic subsystems.

A message may be propagated to more than one units of each level. Conversely, messages that are intended to be diffused in the ecosystem are propagated the other way round. Equivalently, each unit identifies as its ecosystem the other units of the same level, which it can interact with through messages. At the lowest level of the system-unit hierarchy, message passing among units through the corresponding input and output ports indicates the need for information exchange between discrete business process models defined according to different modeling paradigms. It provides the means to integrate models defined within the atomic units, protecting at the same time their atomicity. This implies that messages should be expressed in an approach neutral-format.

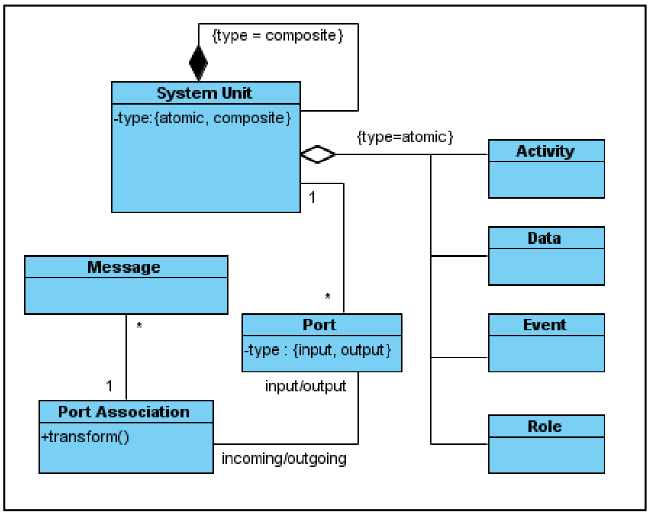


Figure 1. The metamodel of the proposed system-based business process integration framework

Fig. 1 presents the basic metamodel of the proposed business process integration framework. As shown in the metamodel, a system is composed by a hierarchy of system units, either composite or atomic. System units of the same level communicate by exchanging messages. Input and output ports are defined for each system unit for that purpose. Input ports indicate the means to activate the system units, while output ports indicate the means to inform the environment of any kind of information produced or required. The same message may be incoming (input message), when received by a system input port, or outgoing (output message), when released by a system output port, relative to system-unit boundaries. To route such message, an association between the corresponding input and output ports should be defined. A composite system unit is composed of other system units, while the common business process model entities, i.e. activities, data, events and roles, are included in atomic system units. The way these entities are interrelated within a system unit is specified by the modeling logic governing the specific unit.

Messages at the lowest level of the hierarchy convey information related to process institutes, i.e. activities, data, events and roles. Input ports are associated to activation conditions of the respective atomic unit, for example the initiation of a process in an activity-driven unit or the occurrence of an event in an event-driven unit, while output ports are associated to the production of specific knowledge released in the environment, for example the completion of a process in an activity-driven unit or the occurrence of an output event in an event-driven unit. Port definition determines how the input messages will be consumed and how the output messages will be produced. In practice, messages exchanged between different atomic system units constitute the integration mechanism needed to achieve interoperability between the four discrete modeling paradigms.. When an input port is associated with an output port, they exchange messages. In case the respective system units are not of the same type (for example an activity-driven and an event-driven system unit must communicate), a message transformation may be necessary.

This transformation, explicitly defined is related to the corresponding port association defined. The association is responsible for executing the appropriate transformation algorithm, so that messages contain information in well-defined and approach-neutral format.

## V. A Simple Example

The idea of viewing the organization as a system of systems is further elucidated in Fig. 2(a), which illustrates the hospital as a system comprising four subunits, namely the *Inpatient Clinic*, the *Emergency Department*, the *Laboratories* and the *Imaging Department*. The arrival of a patient constitutes an input message for the hospital. When a hospitalized patient is discharged, an output message is generated. Patients usually arrive at the hospital's Emergency Department, where they get examined and it is decided whether they need to be hospitalized. If this is the case, the patients are admitted to the Inpatient Clinic. During the patient's hospitalization, a number of laboratory and imaging examinations are performed. Either a specific modeling paradigm will be employed for each of these four units, or they should be further decomposed to ensure agility. Input and output messages are defined for each unit, as illustrated in Fig. 2(b). Information exchange between units is expressed by the coupling of corresponding input and output messages [16].

As shown in Fig. 2(c), the Inpatient Clinic unit is further decomposed into two other subunits, namely *Patient Admission* and *Patient Treatment.* These subunits are considered atomic. Note that as opposed to the units of the first level, which may exist in reality, the subunits encompassed in the Inpatient Clinic unit constitute a logical functionality grouping. The Patient Admission unit comprises organizational processes, while the Patient Treatment unit includes patient treatment processes, thus they should be described by different modeling paradigms. Fig. 2(d) indicates the adoption of the activity-driven paradigm for the description of Patient Admission, while the event-driven paradigm is suggested [17] for Patient Treatment. The arrival of a patient constitutes an input message for the hospital. When a hospitalized patient is discharged, an output message is generated. Patients usually arrive at the hospital's Emergency Department, where they get examined and it is decided whether they need to be hospitalized. If this is the case, the patients are admitted to the Inpatient Clinic. During the patient's hospitalization, a number of laboratory and imaging examinations are performed. Either a specific modeling paradigm will be employed for each of these four units, or they should be further decomposed to ensure agility. Input and output messages are defined for each unit, as illustrated in Fig. 2(b). Information exchange between units is expressed by the coupling of corresponding input and output messages [16].

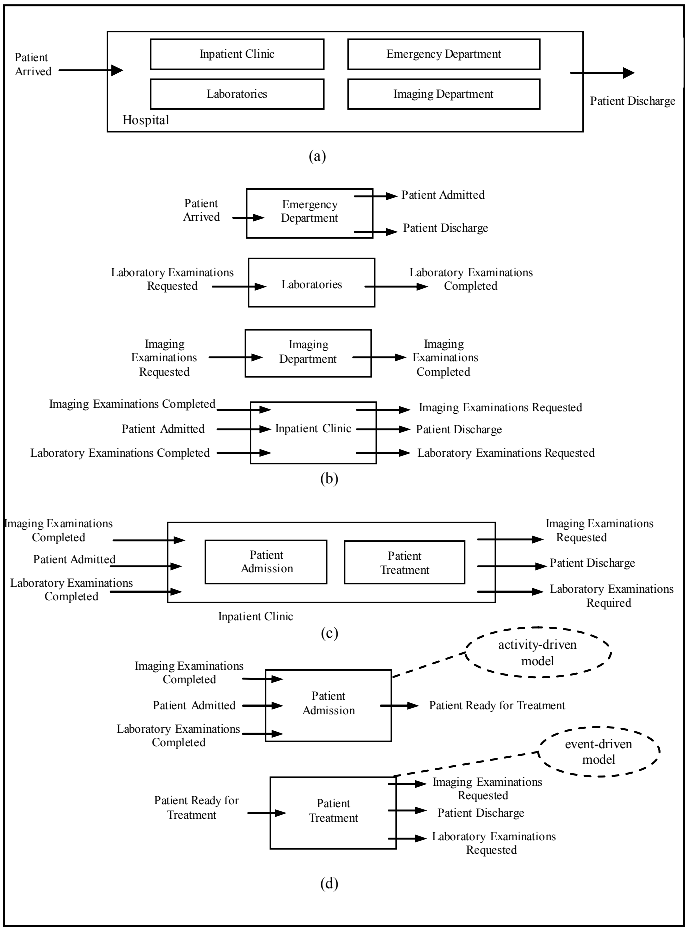


Figure 2. (a) Viewing the Hospital as a system of systems (b) Incoming/outgoing messages of the Hospital's subunits (c) Analysing the *Inpatient Clinic* subsystem (d) Incoming/outgoing messages of the Patient Admission and Patient Treatment atomic units

As shown in Fig. 2(c), the Inpatient Clinic unit is further decomposed into two other subunits, namely *Patient Admission* and *Patient Treatment.* These subunits are considered atomic. Note that as opposed to the units of the first level, which may exist in reality, the subunits encompassed in the Inpatient Clinic unit constitute a logical functionality grouping. The Patient Admission unit comprises organizational processes, while the Patient Treatment unit includes patient treatment processes, thus they should be described by different modeling paradigms. Fig. 2(d) indicates the adoption of the activity-driven paradigm for the description of Patient Admission, while the event-driven paradigm is suggested [17] for Patient Treatment.

Input and output messages for each of these atomic units are depicted in Fig. 2(d). Obvious coupling between messages denotes the way the identified units intercommunicate. In this case, *Patient Ready for Treatment* output message is coupled with the synonymous input message. Although, it might seem that this is the same message, there are actually two different messages having the same name to indicate information exchange. However, output information of Patient Admission process is expressed according to the activity-driven paradigm, while input information of Patient Treatment should be expressed according to the event-driven paradigm. Since the main constituents of all modeling paradigms are the same, appropriate transformation between model entities included in messages (either input or output) can be performed.

## VI. Implementation Issues – Future Work

The basic challenge arising form the automated execution of diverse business process models is the implementation of a Business Process Management System whose engine is capable of executing multiple business process models, as implied in Fig. 3. The issues associated with the implementation of such a BPMS need to be deeply explored so that the best alternative can be unearthed. A basic dilemma, for example, is whether it is more efficient to implement a BPMS, that comprises multiple BPMS engines, as many as the discrete business process models, or it is possible to transform the discrete models into a common one at enactment level, which will then be executed by a single BPMS. We argue that in such a case, the model expressiveness might be diminished. The introduction of MDA (Model Driven Architecture) [18] concepts may contribute towards the integration of discrete BPMS engines and thus is going to be explored.

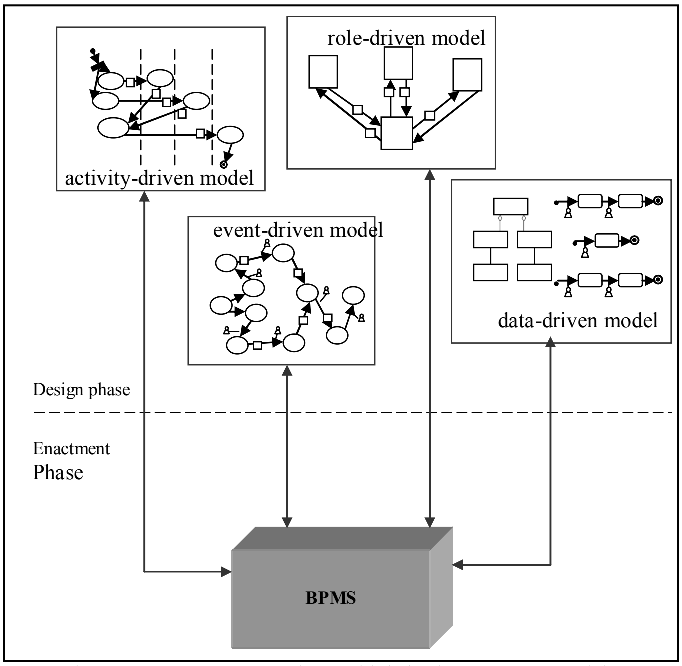


Figure 3. A BPMS executing multiple business process models

## VII. Conclusions

Usually enterprises possess business processes of diverse nature. A common modeling approach for their description does not take into consideration the special characteristics of each process. As a result, the produced models may not all harmonize with the respective actual processes and hence cause inflexibility in their automated execution. It follows that an agile enterprise should support multiple business process models, in a way though that will enable their efficient intercommunication. Such capability will ensure a holistic approach in modeling the enterprise functionality, which may lead to a much more efficient operation of the enterprise. To this end, we proposed in this paper a viewpoint-oriented categorization of business process modeling paradigms to help researchers and business process designers to choose or develop the appropriate modeling approach, based on the nature of the process under design. Furthermore, we proposed a modeling framework, based on the concept of system to achieve the integration of the diverse business process models.

Through the proposed framework, which is going to be further refined and formalized in our future work, it was stressed in this paper that model transformation may prove a promising path towards the attainment of enterprise agility and thus should be adopted by modern enterprises for the development of a holistic BPMS approach

## References


[1] Dumas M., Aalst W., Hofstede A.: Process-Aware Information Systems. John Wiley & Sons INC (2005).

[2] OMG, Business Process Management Notation. Version 1.0,(OMG, 2006).

[3] Scheer A.W., *ARIS-Business Process Modeling*. 2nd ed. Berlin (Springer 1999).

[4] Reichert, M.; Dadam, P., ADEPTflex – supporting dynamic changes of workflows without losing control in *Journal of Intelligent Information Systems*, 1998, 10, pp. 93-129.

[5] OMG Inc: Unified Modeling Language: Superstructure. Version 2.1.1, (OMG, 2007).

[6] Andrews T., Curbera F., Dholakia H., Goland Y., Klein Y., Leymann F., Liu K., Roller D., Smith D., Thatte S., Trickovic I., Weerawarana S., *Business Process Execution Language for Web Services*, Version 1.1. (BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems, 2003).

[7] Muller Dominic, Reichert Manfred and Herbst Joachim, Flexibility of Data-Driven Process Structures**,** in *BPM Workshops*, ***2006,*** LNCS 4103, pp.181-192.

[8] Dayal U., Hsu M., and Ladin R., Organizing Long-Running Activities with Triggers and Transactions in Proceedings of *ACM International Conference on Management of Data*, 1990, pp.204-214.

[9] Widom J. and Ceri S., Active Database Systems: Triggers and Rules For Advanced Database Processing (Morgan Kaufmann, 1996).

[10] Kappel G., Raush-Schott S., Retschitzegger W., Coordination in Workflow Management Systems – A Rule-based Approach in Lecture Notes In *Computer Science*, 1998.

[11] Chen Lin, Li Minglu, Cao Jian, ECA Rule-based Workflow Modeling and Implementation for Service Composition in *IEICE Trans. Inf. & Syst*., 2006, Vol.E89–D, No.2.

[12] Balabko Pavel, Wegmann Alain, Ruppen Alain, Clément Nicolas, The Value of Roles in Modeling Business Processes in *BPMDS* 2004.

[13] Haeckel, S. H.: Adaptive Enterprise Design: The Sense-and-Respond Model. Planning Review, Vol. 23, No. 3, pp. 6-42 (1995).

[14] Desai A.: Adaptive Complex Enterprises. Communications of the ACM, Vol. 48, No. 5, pp.32-35 (2005).

[15] Schulz, S., Ewing T. C., and Rozenblit J. W.: Discrete event system specification (DEVS) and statemate statecharts equivalence for embedded systems modeling. Proc. of 7th IEEE Conf. & Workshop on the Engineering of Computer Based Systems, pp. 308–316 (2000).

[16] OMG: Systems Modeling Language. Version 1.1, OMG (2007).

[17] Alexopoulou Nancy, Nikolaidou Mara, Anagnostopoulos Dimosthenis, Martakos Drakoulis. "An Event-Driven Modeling Approach for Dynamic Human-Intensive Business Processes", edBPM Workshop 2009.

[18] OMG. MDA Guide Version 1.0.1. 12th June 2003