Extending UML Activity Diagrams for Workflow Modeling in Production Systems

Ricardo M. Bastos, Duncan Dubugras A. Ruiz
Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre - RS – Brasil
e-mail: {bastos,duncan}@inf.pucrs.br

Abstract

This paper presents an approach to describe business process in production systems using workflow concepts. This work presents an approach to model workflow applications based on the C-Wf model and UML (*Unified Modeling Language*) Activity Diagrams. The C-Wf model represents the structural and functional enterprise objects involved in the business processes, such as enterprise activities, human resources, machine resources, etc. Through the GWf reference model classes it is possible to build up particular enterprise models that involve the elements required to the production planning process and the basic information for monitoring business processes execution under a workflow point of view. The UML Activity Diagrams displays a sequence of activities including execution alternatives and the involved objects in performing the work. This diagram could de used to describe the workflow of a business process, defining the objects related with the enterprise activities.

1. Introduction

UML has been acquired the standard status recently [11] as a language for modeling of information systems. This status results from the quality and richness of its modeling tools and the integration among them. One of the main advantages provided by the dissemination of UML use is that the software designers, becoming UML experts, can design a large class of problems, knowing (almost) only one modeling language. As a consequence, there are several efforts to employ UML as a modeling language for different problem classes. Specifically concerning the use of UML to describe workflow models, there are some few solutions, e.g. [1] [7] [8] [14] [15].

C-Wf [3] is a model based on the object-oriented approach to model business processes in production systems using workflow concepts. Through the C-Wf reference model classes it is possible to build up particular enterprise models that involve the elements required to the production planning process and the basic information for monitoring business processes execution under a workflow point of view.

This paper presents an approach to describe business process in production systems using workflow concepts. In this sense, we define an extension of the UML Activity Diagram called Workflow Activity Diagram which represents the GWF concepts. By the intensive use of UML use cases, our approach reinforces the usability of UML in the context of business modeling.

The paper is organized as follows. The section 2 presents a briefly description of the C-Wf model. Section 3 discusses how the UML creators [4] [9] [11] suggest the use of UML tools to describe workflow and shows what are the current difficulties to design workflow with UML. The Workflow Activity Diagram, strictly based on UML Activity Diagrams and stereotypes is introduced in details at section 4, and how to apply Workflow Activity Diagrams is discussed at section 5. Section 6 presents some related work and section 7 presents the conclusion and future work.

2. The C-Wf reference model

The C-Wf model (Figure 1) is an object reference model that represents the structural and functional enterprise elements involved in business processes, such as enterprise activities, human resources, machines, etc [3]. The GWf model is based on CIMOSA [16] and WfMC [6] [13] basic concepts, and enables the monitoring and control of domain processes execution by a Workflow Management System (WfMS). The goals are: (1) to capture all the necessary information to describe a Domain Process and (2) to complement the mapping with additional information required.

The basic concepts of the C-Wf model may be divided into two main contexts: (1) Process design and definition, and (2) Process instantiation and control. This division is showed in the Figure 1 where its left side corresponds to

the context (1) and its right side corresponds to the context (2).

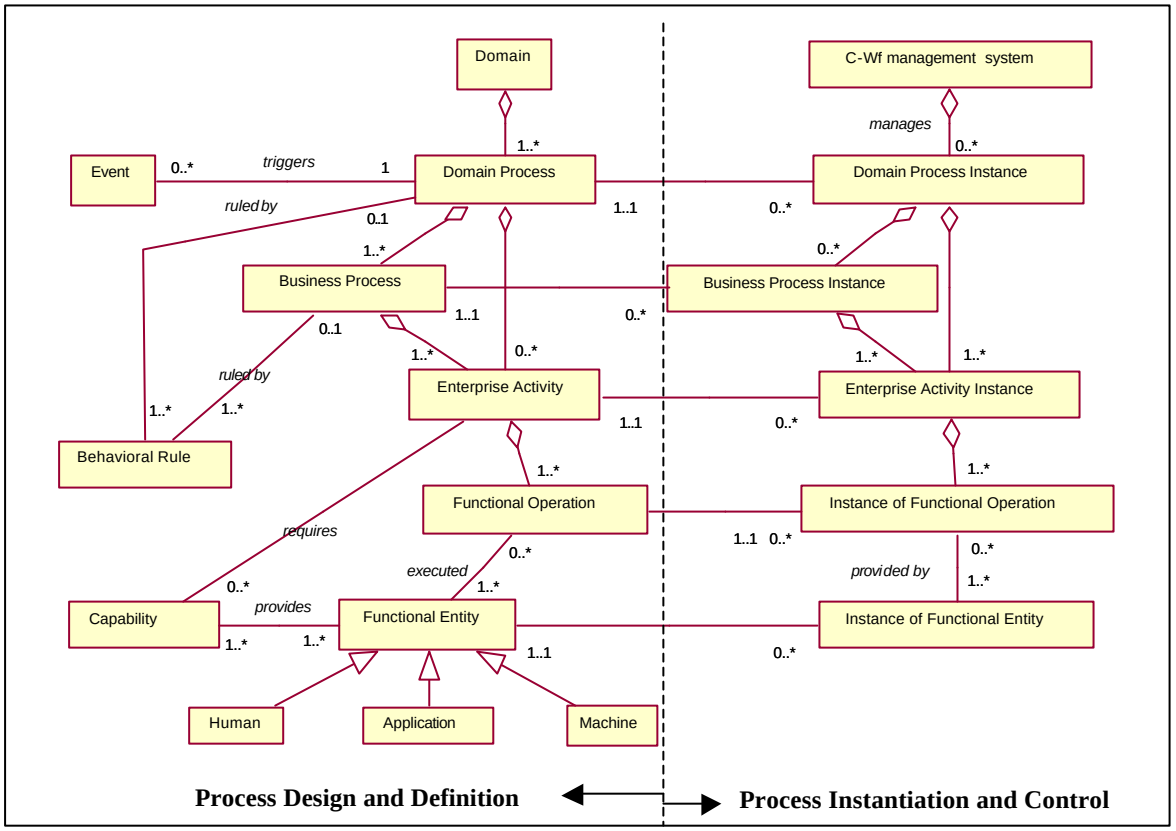


Figure 1. C-Wf UML class diagram

The process design and definition is strongly based on CIMOSA constructs and process instantiation and control is based on WfMC standard. In fact, C-Wf benefits from the richness and diversity of the concepts offered by CIMOSA to design production processes. And C-Wf turns feasible the use of WfMS to manage the execution of production processes because its concepts have direct correlation with WfMC terminology. As a consequence, instances of a production process modeled with C-Wf can be managed by a WfMS in almost all-modeling possibilities.

2.1. Process definition

A Domain is basically defined by a set of business objectives and constraints of an organization and it is composed by a set of interactive Domain Processes. A Domain Process accomplishes part or whole domain objectives under given constraints. Activated by Events, a Domain Process is functionally decomposed into Business Processes and Enterprise Activities. A Business Process represents an aggregation of Enterprise Activities. Both of them could be used in different Domain Processes. The Enterprise Activities define elementary tasks to be performed in the enterprise, which consume inputs to produce outputs and need allocation of time and resources for the full duration of their execution. Each Enterprise Activity involves one or more Functional Operations that are executed by the Functional Entities, with its own execution time requirements. Functional entities represent the production resources (humans, machines and applications). To guarantee the consistency among Enterprise Activities and Functional Entities, the Functional Entities provide a set of capabilities that fulfill the Enterprise Activity requirements.

The dynamics of the Domain Process and Business Process is defined by the Behavioral Rules. Behavioral Rules describe the interconnections among Enterprise Activities showing the following types: start and end of a process, single thread, and-split, and-join, or-split, or-

join and iteration. A complete description of a Domain Process in terms of its Enterprise Activities interconnected by Behavioral Rules corresponds to a representation of a workflow production process, and a WfMS can manage its process instances after the resource allocation. In fact, Domain Process, Business Process and Enterprise Activity are the concepts that represent the functionality and behavior of an enterprise model.

2.2. Process instantiation

A C-Wf Management System (C-WfMS) is responsible for the creation and control of occurrences of the defined Domain Processes and initiates each instance of any Domain Process when receives one (or more) Event. After the resource allocation and consequently the schedule of activities and resources, a C-WfMS will be capable to control the timing of the start and the end of each Enterprise Activity instance and the Functional Entities involved (production resources). Due to the need of defining the time granularity of Functional Entity allocation to an Enterprise Activity, we opted to consider that one Functional Entity remains allocated to an Enterprise Activity during all the execution time of the latter.

3. UML tools and workflow

In [9], the authors propose the application of **use cases** to identify the functional requirements of the system. Through use cases it is possible to build up a business-model for a company. A business use-case model describes the business processes of a company in terms of business use cases and business actors. Like the use-case model for a software system, the business use-case model presents a system from the usage perspective and outlines how it provides value to its users.

Actually, the software development process presented by the authors is defined as a use-case driven process, where the developers create a series of design and implementation models that realize the use cases.

According the software development process defined in [9], the analysis classes are essentially identified from the use cases. It is justified because the use case realization is described in terms of analysis classes and their objects. The realization of the different flows or scenarios of the use case is depicting through analysis objects interaction.

The enterprise software always work in the context of higher-level business processes, where there are actors interacting with the system, which defines the use case model of the software. These business processes are types of workflow because they represent the flow of work and objects through the business [4]. Thus, we consider that the workflow model for a company could be defined using the business use case model as a reference, allowing the consistent integration between the WfMS with the enterprise software.

3.1. Activity diagram

According [4] [9], Activity Diagrams could be used to model a use case. Typically, for a Use-Case description it is necessary to identify the actors involved and to describe the flow of events (basic and alternative paths). Based on a use case, it is possible to identify the objects required to accomplish the functional requirements involved, as well as the answers of the system during its execution.

In Activity Diagrams we identify the activities that must be realized to execute a use case and the relationships among them. Besides that, it is possible to identify the objects involved in which activity and define how their role, state and attributes values are changed. At the Table 1 it is presented the main concepts of Activity Diagrams, extracted from [4] and [11].

Table 1 – Activity diagram concepts

Concept	Details
Activity	Ongoing non-atomic execution within a state machine. Activities ultimately result in some *action*.
Action	An executable atomic computation that results in a change in state of the system or the return of a value.
Action State	A state that represents the execution of an atomic action, typically the invocation of an operation.
Activity State	It is a composite, whose flow of control is made up of other activity states and action states. Activity states are not atomic, meaning that they may be interrupted. Activity states can be further decomposed, their activity being represented by other Activity Diagram.
Transition	Represents the flow of control between two activities. Transition shows the path from one action or activity state to the next action or activity state.
Object Flow	Represent an object involved in a flow of control associated with an activity diagram.
Object State	A condition or situation during the life of the object during which is satisfies some condition, performs some activity, or waits for some event.
Swimlane	A partition for organizing responsibilities for activities. Swinlane do not have a fixed meaning, but they often correspond to organizational units in a business-model.

However, [5] claims that the application of Activity Diagrams as a UML tool remains limited due to the insufficient understanding. In fact, other authors claim the same: [14] [12]. Probably, this is the reason why there is no many works using Activity Diagrams for workflow modeling.

4. The workflow activity diagram

The main contribution of the C-Wf reference model consists of the integration of both enterprise and workflow modeling concepts. In order to make feasible the use of Activity Diagrams jointly with the C-Wf in enterprise modeling, and increase its application in business use-case model, we propose an extension of this diagram called Workflow Activity Diagram (WAD). In this extension, we associate the Activity Diagram elements with the C-Wf classes using stereotypes and we define some new properties for this diagram. Stereotypes are UML extension mechanism that allows the creation of new kind of building blocks that are derived from the existent ones but that are specific for a particular problem [4].

Each element defined at the WAD is represented by a stereotype in order to establish the relationship with a C-Wf class. Table 2 shows the main elements of Activity Diagrams and the related classes at the C-Wf.

Table 2 – Relationship between activity diagram elements and C-Wf model classes

Element	C-Wf Model Class	Comments
Swimlane	Domain	Represents the organizational unit where the business process or enterprise activities that compose a domain process are executed.
Activity	Business Process Enterprise Activity	Business Process and Enterprise Activities define the decomposition levels of a WAD.
Transition	Transition	Defined from the Domain Process Behavioral Rules.
Object Flow	Event Functional Entity	The Functional Entity class is specialized in three classes: Human, Application and Machine.

The main aspects addressed by WADs are:

- to describe the workflow model as a set of use cases using the C-Wf concepts;
- to apply the objects from C-Wf as stereotypes in order to describe a domain process and its business process and enterprise activities;
- to improve the Activity Diagram representation applying the principles of workflow modeling;
- to take advantage of the fact that UML is an OMG standard and its use is growing quickly.

In our approach, each use case identified at the business-model is related with a Domain Process, which is described by a WAD. When present into a WAD, each Business Process represents a potentially reusable sub-process that can also be related to a use case at the business-model. It means that each Business Process, identified at one Domain Process, must be detailed by another WAD in order to identify its particular workflow. To guarantee the consistency among WADs, all object flows connected to a Business Process must be present into the corresponding WAD that details such Business Process.

An Enterprise Activity is an aggregation of Functional Operations, which are executed by one or more Functional Entities. As a consequence, we associate each Functional Operation directly with one set of Functional Entities qualified for its execution. The resource allocation process will allocate the specific Functional Entity that will execute the Functional Operation for an Enterprise Activity instance. In order to define the time granularity of the Enterprise Activities, we considered that its Functional Operations components are executed simultaneously. It means that the Functional Entities required for the execution of each Functional Operation must be allocated during all the execution time required for the Enterprise Activity occurrence.

4.1. The WAD and C-Wf concepts

The objects involved at a WAD are represented through the usual notation defined at UML, i.e., the stereotype name showed as text strings surrounded by guillements (<< >>) placed in the object symbol. Thus, the stereotyped classes defined for the WAD are:

<<Event>>

Event instances are used as a trigger to Domain Process. Thus, an event instance is related with the first Business Process or Enterprise Activity of the WAD. An event is an external input generated by an actor to a use case, starting the WAD.

<<Human>>

Instances of this class represent the human Functional Entity involved with the Functional Operations execution. For this reason, Human instances are related with Enterprise Activities, although it is possible to associate it with Business Process due to its transitivity property (Business Process are aggregations of Enterprise Activities).

<<Application>>

Represents software application identified by UML Packages and the respective object related with the workflow activity. A Package is a general-purpose mechanism for organizing elements into groups. Graphically, a package is rendered as a tabbed folder. The objects that belong to a software application are

encapsulated into a package and its relationships with the objects from other packages define the integration links among these kinds of Functional Entities. Consequently, objects from different Applications could be related with a same workflow activity.

Considering the WfMC terminology [13], we opted to represent into a WAD only the objects that can be classified as Workflow Relevant Data (the WfMC other types are: Application Data and Workflow Control Data).

The Application objects correspond to the Entity and Control analysis classes defined at [9]. An Entity class is used to model information that is long-lived and often persistent. Entity classes often show a logical data structure and contribute to the understanding of what information the system is dependent. Control classes represent coordination, sequencing, transaction, and controls of other objects and are often used to encapsulate control related to a specific use case. Control classes are also used to represent complex derivations and calculations. The representation of an object with the related Application is presented at Figure 2.

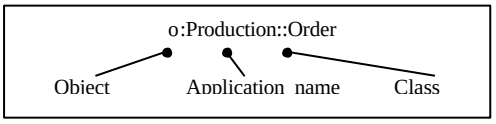


Figure 2 – Object path name

<<Machine>>

Represents all kinds of physical Functional Entity used to execute a Functional Operation. Thus, a Machine instance could be related just with Business Process and Enterprise Activities.

The state of a Human, Application and Machine instances are defined according to the Functional Operation in execution. Event instances don't require state definition because they are just used as a trigger to Domain Processes and Business Processes.

4.2. Workflow activity diagram representation

The proposition of the WAD is to improve the capacity of the UML Activity Diagrams to be used for workflow modeling. The main aspects addressed are:

- representation of the different activity levels (business process and enterprise activities) required in a domain process specification;
- association of the resources (functional entities) required to execute an activity (business process and enterprise activities);
- identification of the software application (represented by packages) related with the activities that composes the domain process workflow.

Figure 3 shows a WAD for a Domain Process identified by the use case *Place Production Order* for a hypothetical manufacturing enterprise. The workflow activities require to be related with the objects from the Production System and the Inventory System. At the sequence the elements that compose the WAD are examined.

Business process and enterprise activity

The Business Processes and Enterprise Activities define the two kinds of activities at the WAD. We adopted for an Enterprise Activity the same representation of an Activity in a UML Activity Diagram (see for example, the Enterprise Activity *Opening Customer Order* in Figure 3). For a Business Process, we define a stereotype representation as showed at the Business Process *Schedule Customer Order* in Figure 3.

In some cases, like the *Identify Required Materials* Enterprise Activity (see Figure 3), it is necessary to express that the same activity will be executed one or more times depending on the conditions involved. In UML, it is defined as a *dynamic concurrency*, meaning that multiple copies of the activity may occur concurrently. A multiplicity string shows it in the upper-right part of the activity symbol [11].

When we have a *dynamic concurrency*, it is important to note that the multiple copies of the activity are performed during the same time interval. Interleaving or simultaneous executing two or more threads can achieve this concurrency. If the situation requires simultaneous execution, it is necessary to allocate one instance of each <<Human>> or <<Machine>> Functional Entity necessary for each thread. On the other hand, for the <<Application>> Functional Entity each thread could use different instances as input and must generate different instances as output. In order to represent the different possibilities of *dynamic concurrency* we extend the UML representation of the multiplicity string by using *i for interleaving and *s for simultaneously execution of threads.

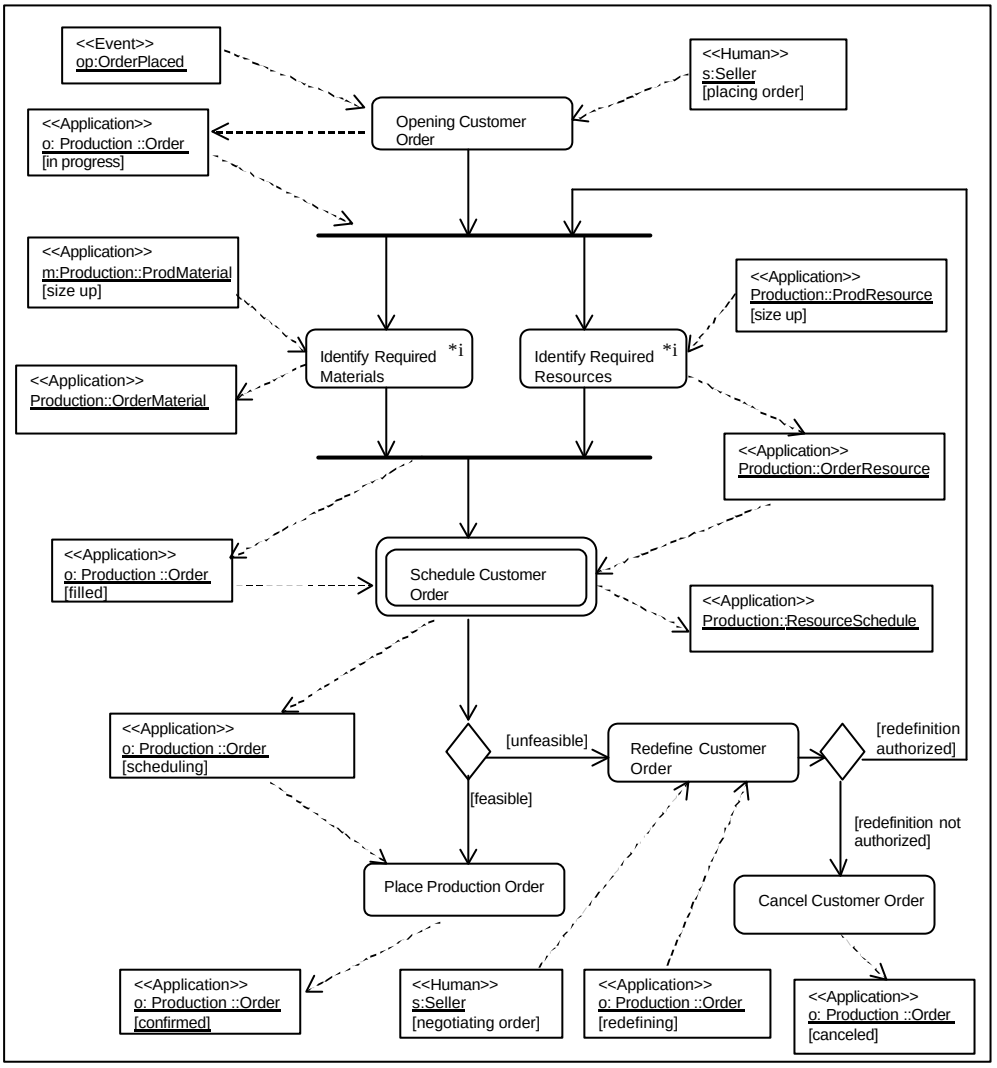


Figure 3 – Workflow activity diagram example

Object flow

As we said before, for the execution of the Functional Operation (that composes an Enterprise Activity) it is necessary to have one or more Functional Entity. In this way, each enterprise activity has at least one Functional Entity object related. The <<Human>> and <<Machine>> objects are applied to execute the activity at the real world, as for example the *Seller* object at the *Opening Customer Order* Enterprise Activity (Figure 3). The <<Application>> objects represent the objects of the information system that are produced or destroyed by an Enterprise Activity (for example, the *Opening Customer Order* Enterprise Activity creates an *order* object) or applied to fulfill a Functional Operation (for example, the *orderResource* object at the *Identify Required Resources* Enterprise Activity).

In order to represent the object flow we propose the notation showed at Figure 4 for the object Production. The Functional Entity stereotype defines the object meaning at the Workflow Diagram Activity. The Object State has the same UML notation and it is optional. The Object State is related to the Functional Operation required by the Enterprise Activity since it represents the state of the object during its execution (for example, the object *prodMaterial* remains in the state *size up* during the execution of the Functional Operation that defines the amount required of the material to fulfill the

customer order). Actually, since the <<Application>> objects belong to a software application, the Functional Operations are implemented by methods of these objects at the information system.

The objects may be the output of one activity and the input of many other activities. The <<Human>> and <<Machine>> always are input objects for the activities since they will be used to perform a Functional Operation. The <<Application>> objects could be input or output in an activity. They will be output when the activity creates, destroys or just modifies the value of some attributes without requiring the execution of any operation by itself. When the <<Application>> object is required to execute some Functional Operation, then it is considered input for the activity.

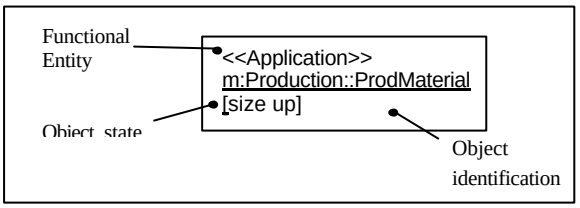


Figure 4 – Object representation

The <<Event>> object always must be input for the first activity of the WAD. Sometimes, could be interesting to identify the object <<Event>> generated by an activity of the workflow model. In this case, an object <<Event>> represents an output of the domain process, which will be an input object <<Event>> for another domain process establishing the link between them.

At Figure 5, it is showed the packages and its respective objects for the WAD presented at Figure 3. In this example, all the objects that are presented at the object flow are showed into its respective package. In the Functional Entity Machine package there is not objects because this kind of Functional Entity is not applied at the use case *Place Production Order*. In fact, the Process Activity package is dependent on the Functional Entity packages through the Functional Operation class. It means that each instance of the Functional Operation class can be linked with one or more Functional Entity instances.

Transitions

The transitions define the flow of control among the activities. The transition from an activity to another occurs when the current activity is completed and the condition is true. As in UML, we do not define a formal format for transition condition expression.

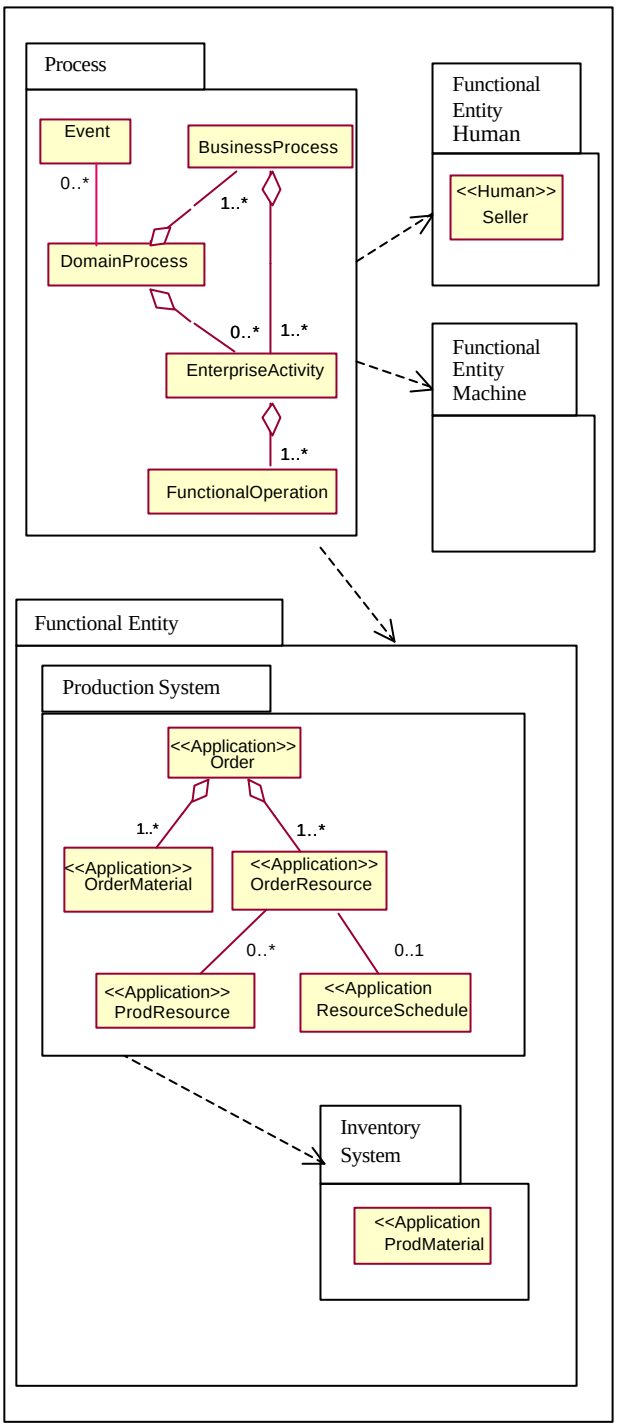


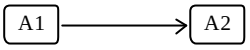
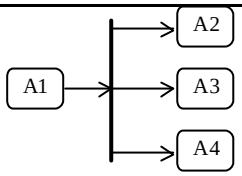
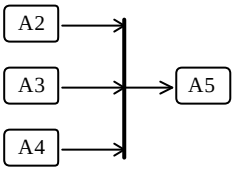
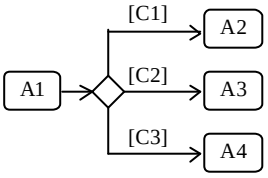
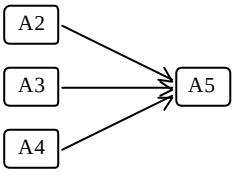
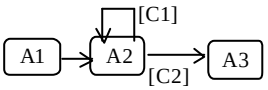
Figure 5 – Packages and objects

Besides the sequential transition, it is possible to represent another kind of paths to model flow of control as presented at Table 3. This notation is an extension of

UML Activity Diagrams applying concepts defined in [13].

When there is *dynamic concurrency* in a WAD, even though we may have interleaving or simultaneous execution of threads, we considered that only at the end of all threads the output condition will be evaluated and such transition may occurs.

Table 3 - Transition path representation

Routing possibilities	Diagram sample (considering the natural execution flow from left to right)
Single thread: an activity can be executed after the completion of another.	
And-split: a single thread of control splits into two or more threads that are executed in parallel within the workflow, allowing multiple activities to be executed simultaneously.	
And-join: a point in the workflow where two or more parallel executing activities converge into a single common thread of control. It is a synchronization point in the workflow.	
Or-split: a point within the workflow where a single thread of control makes a decision upon which branch to take when encountered with multiple workflow branches. The decision is specified by transition conditions (e.g. C1, C2 and C3).	
Or-join: a point within the workflow where two or more alternative activity workflow branches re-converge to a single common activity as the next step within the workflow.	
Iteration: a workflow activity cycle involving the repetitive execution of one (or more) workflow activity(s) until a condition is met. C1 and C2 are transition conditions.	

Swimlanes

The swimlane element of UML Activity Diagrams defines an interesting aspect for workflow modeling since it propitiates to represent the Domain where the Enterprise Activities occur. In fact, although a domain process must be associated with a domain, some of the enterprise activities required to fulfill it are executed in another Domains.

Using swimlanes it is possible to visualize the interactions among the Domains. Besides that, it propitiates the identification of the Functional Entities required to participate of the Domain Process, which are under responsibility of another Domain.

At Figure 6 it is presented a partial WAD presented at Figure 3 with swimlane representation. Due to the lack of space, we omitted the object flows

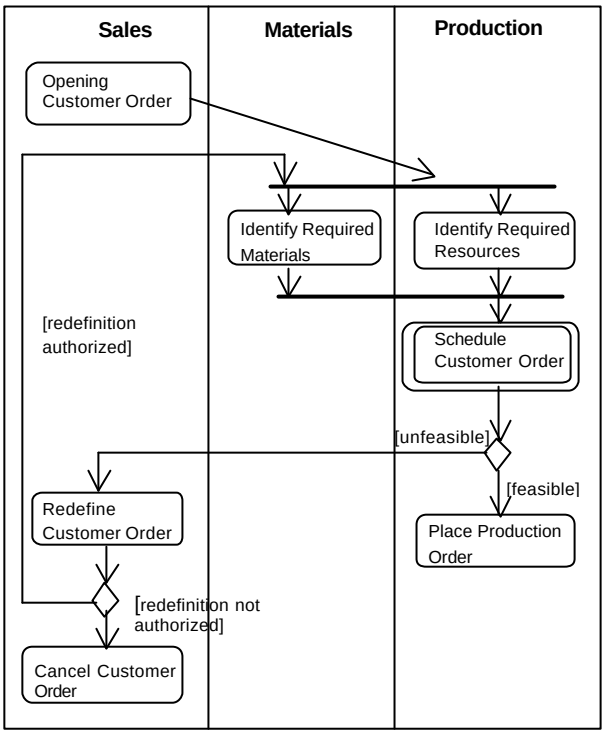


Figure 6 – Workflow diagram activity with swimlanes

5. Applying the workflow activity diagram

The WAD is a tool for workflow modeling that applies stereotypes derived from the G-Wf model. It propitiates to represent a Domain Process in terms of its Business Process and Enterprise Activities components with the identification of the Functional Entities required for their execution. However, the main potential of this approach is (i) the integration with the information system model, and (ii) the identification of the Functional Entities associated with the Functional Operations as a input for a resource allocation process system.

As explained before, [9] propose the application of use cases to identify the functional requirements of the system and build up the business-model. The process of the information system definition is based on its use cases, where the main idea is to describe how a specific use-case is realized and performed in terms of analysis classes and their interacting analysis objects. A use-case realization has a textual flow-of-events description, class diagrams that depict its participant analysis classes, and interaction diagrams that depict the realization of a particular scenario of the use-case in terms of analysis object interactions.

In our approach, each use case corresponds to a Domain Processes or Business Processes. In order to specify the use case realization, we propose to describe the flow of work through the WAD, identifying for each activity the analysis object involved in its execution. The analysis objects are defined as the Functional Entities that are required to execute the Functional Operations of the Enterprise Activities. Through this approach it is possible to integrate the business use-case model with the workflow model, since both of them use the same object classes to realize a use case that is described by its WAD.

The WAD is a tool that could be used to depict the workflow model of all process (Domain Process and Business Process) of an enterprise, as well as to describe the business use-case model, having the C-Wf as a reference model. The description of the flow of work defined by the activities for a process is the input for a WfMS and the object flow related with the activities defines the resources required for its execution, which must be scheduled by a resource allocation method.

We consider that a resource allocation method must be able to attend in a dynamic way all the production events that affect the production system, like new production orders, machine breakdown, raw material fault, production fail, workers fault, etc. For this purpose, the method must use all of the information about the status of the production system. The production system status is basically defined by the value of the attributes of each enterprise activity instance. In its turn, the attribute values of the enterprise activity instance are derived from the attribute values of each functional operation that is executed in order to accomplish the enterprise activity. These attributes concerns with the temporal values defined for each functional operation instance, e.g. deadline, estimated execution time, time-out, as well the status of its execution, e.g. time remaining, execution situation and so on.

All of these attributes are defined at the C-Wf model including the aspects related with the functional entity instances through its particular work list that defines its commitments and its situation in each work time period (basically available to order or not). Using the WAD, it is possible to identify the relationships between the business processes and enterprise activities that compose the domain process. Considering that the Enterprise Activities need functional entities to be executed, each activity must be planned considering its estimated execution time and the availability of the necessary functional entities. The whole planning must be done in such a way as to respect the production event deadline.

6. Related work

There are few works concerning integration between UML and workflow modeling, considering the characteristics defined by the WfMC and OMG. We consider that WfMC proposed [6] [13] a wider-ranging reference model and, at present, any modeling approach must go along with such reference model. In this sense, we have found the following works in the literature.

The approach of G.Wirtz, M.Weske and H. Giese [14] [15] extends UML to allow workflow modeling, integrating standard object oriented structure modeling using UML diagrams with Petri-Net techniques for specifying behavior, by the use of Object Coordination Nets (OCoNs). According the authors, this integration provides an adequate support for modeling all aspects of workflows. Basically, OCoNs works with two types of places, event pools (for tokens) and resource pools (for resources). And presents three kinds of nets to describe the behavior: protocol nets (PN), service nets (SN) and resource allocation nets (RAN). PN specifies externally visible behavior of services provided. SN describes the detailed workflow when performing a single service, and RAN describes the resource management. G.Wirtz, M.Weske and H. Giese conclude that the formalism permits the designers to deal with complex workflow systems in a scaleable manner. Comparing with our approach (WAD), we have looked for a solution only using UML concepts, basically employing stereotypes to model the workflow processes. Also, our approach clearly models the splits and joins of threads and represents synchronous and asynchronous dynamic concurrency that can be present into a workflow model. To achieve this purpose, we have proposed a more precise understanding of UML Activity Diagrams and its relationships with the other UML description tools. Although UML+OCoNs seems to be richness formalism than WAD, we believe WAD can be more familiar to UML skilled designers.

P. Hruby [7] [8] presents a method that uses UML for specification of WfMS. Basically, it is defined four stereotypes corresponding to Business Object, Business Process, Workflow and Team Role. Besides that, the

modeling of a workflow application is done as follows. UML static structure diagram is used to represent a team structure. UML sequence diagram represents instances of business processes and interactions between business processes and actors. UML use-case diagrams represent static relationships between business processes. UML collaboration diagrams also represent interactions and relationships between business process and actors. UML activity diagrams can represent allowable ordering of business processes. The differences between the P. Hruby method and our approach are: (1) our approach adopts the WfMC reference model that defines a larger set of workflow constructs, and (2) WAD describes the behavior of each Domain Process. In our opinion, it is difficult to map the WfMC constructs to UML without the clearly definition of the semantic interrelations among UML tools, an UML problem stated by [14] [12] too.

L. Baresi et al. [1] presents a brief description of the WIDE workflow development methodology. WIDE have three main steps: (1) analysis, (2) WF design and (3) mapping to target WF systems. For the analysis step, WIDE employs UML tools: *use-cases*, *scenarios* and *sequence diagrams*. Only in special situations where there is a significant level of complexity, WIDE uses *activity diagrams*. For the Wf design, it is used the WIDE Wf model. Similar to the approach of G.Wirtz et al., the description of the process is done by a specific formalism and not by using some of UML tools. Even though the WIDE methodology seems to deal with all aspects relevant to build a workflow system, the use of its own formal model to describe the dynamic aspects of each business processes may be unfamiliar to UML skilled designers.

7. Conclusion

This work has presented an extension of UML Activity Diagram called WAD. This diagram is useful to depict the workflow model describing the activities – business process and enterprise activities – involved in a Domain Process. Moreover, for each activity it is identified the resources – Functional Entity – required for its execution. These resources are identified as analysis classes from the Class Diagram of the enterprise business-model built up at the analysis model.

The WAD basically represents the elements defined at UML Activity Diagram using stereotypes in order to represent the main C-Wf model concepts [3]. Consequently, each element defined at the WAD is described by a stereotype establishing its relationship with a C-Wf class. The WAD depict the workflow model identifying its activities and resources required for its execution defining its relationships and sequentially. Also, WAD represents occasional dynamic concurrency in a workflow, which can be synchronous or asynchronous execution threads.

The contributions of our approach are: (1) the description of workflow model as a set of use cases using the C-Wf concepts; (2) the use of C-Wf objects as stereotypes in order to describe a domain process and its business process and enterprise activities; (3) the improve of the Activity Diagram representation by applying the principles of workflow modeling; (4) taking advantage of the fact that UML is an OMG standard and its use is growing quickly.

Our next steps are the build of an execution environment composed by a Java-based Resource Allocation Process System, based on the framework defined in [2], and a Workflow Enactment Service that supports resource allocation management. Due to a partnership established between Hewlett-Packard Computers – Brazil and PUCRS, we are intended to use HP-Changengine.

8. References

[1] Baresi, L. et al. WIDE Workflow Development Methodology. In: Intl. Conf. On work Activities Coordination and Collaboration (San Francisco-CA, USA: Feb. 22-25 1999). **Proceedings…** San Francisco: ACM Press,1999. p.19-28

[2] Bastos, R.M. **Resource Allocation Planning using Multi-Agent Systems**, Porto Alegre – Brazil: PPGC-UFRGS, 1998. (PhD Thesis, in Portuguese)

[3] Bastos, R.M. & Ruiz,D.D.A. Towards an Approach to Model Business Processes using Workflow Modeling Techniques in Production Systems. In: HICSS´34 (Maui – Hawaii: Jan.3-6 2001) **Proceedings…** Los Vaqueros – CA: IEEE Press, 2001. (CD-ROM/Abstracts Proceedings)

[4] Booch, G. et al. **The unified modeling language user guide**. Reading, MA : Addison-Wesley, c1999. 482 p.

[5] Fowler, M. **UML distilled : applying the standard object modeling language**. Reading, MA : Addison-Wesley, c1997. 183 p.

[6] Hollingsworth, D. **The Workflow Reference Model.** Hampshire – UK: WfMC, Jan. 1995.

[7] Hruby, P. Specification of Workflow Management Systems with UML. In: OOPSLA-98 Object-Oriented Workflow Management Systems Workshop (Vancouver – Canada: Oct. 18-22 1998). **Proceedings…** Vancouver – Canada: ACM Press 1998.

[8] Hruby, P. Structuring Specification of Business Systems with UML. In: OOPSLA-98 Business Object Workshop (Vancouver – Canada: Oct. 18-22 1998). **Proceedings…** Vancouver – Canada: ACM Press 1998.

[9] Jacobson, I. et al. **The unified software development process**. Reading, MA: Addison-Wesley, c1998. 463 p.

[10] Object Management Group. **Unified Modeling Language (UML) 1.3 specification.** Available at: http://www.omg.org/cgi-bin/doc?formal/00-03-01.pdf.gz.

(access date: May 2001)
[11] Rumbaugh, J. et al. **The unified modeling language reference manual**. Reading, MA : Addison-Wesley, c1999.
[12] Schmidt, M.T. Building Workflow Business Objects. In: OOPSLA'98 Business Object Workshop. (Vancouver, Canada: Oct. 18-22, 1998). **Proceedings…** Heidelberg: Springer, 1998.
[13] Workflow Management Coalition. **Terminology and Glossary.** Hampshire – UK: WfMC, Feb. 1999.
[14] Wirtz, G.;Weske, M.; Giese, H. Extending UML with workflow Modeling Capabilities". In: 7th CoopIS (Eilat – Israel: Sept. 6-8 2000) **Lecture Notes in Computer Science** v.1901, Heidelberg: Springer, 2000. P. 30-41.
[15] Wirtz, G.; Giese, H. Using UML and Object-Coordination-Nets for Workflow Specification. **Proceedings**… IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC'2000) Nashville-TN USA: Oct. 8-11 2000.
[16] Zelm, M.; Vernadat, F. B.; Kosanke, K. "The CIMOSA business modelling process", **Computers in Industry**, Amsterdam: Elsevier, Oct. 1995, pp.123-142.