Building workflows definitions based on business cases

Jorge Cardoso

Department of Mathematics and Engineering
University of Madeira
9050-390 Funchal, Portugal
jcardoso@uma.pt

Abstract. The maturity of infrastructures that support e-services allows organizations to incorporate Web services as part of their business processes. One prominent solution to manage, coordinate, and orchestrate Web services is the use of workflow technology. While workflow management systems architectures, language specifications, and workflow analysis techniques have been extensively studied there is a lack of tools and methods to assist process development. The purpose of our study is to present a framework to assist process analysts and designers in their task, allowing the creation of processes (Web processes and workflows) with a higher quality. The framework structures a comprehensive set of steps that drives the analysis and design of processes based on requirements gathered from communication with managers and experts.

1. Introduction

The Web, the development of E-commerce, and new architectural concepts such as Web services have created the basis for the emergence of a new networked economy [1]. With the maturity of infrastructures that support e-commerce, it will be possible for organizations to incorporate Web services as part of their business processes. A wide spectrum of modern workflow system architectures has been developed to support various types of business processes [2].

Research has targeted three main areas: workflow architectures, specification languages, and process analysis. These areas of research are of recognized importance for the construction of sophisticated and robust workflow systems. Nevertheless, one important area has been overlooked, the research of the lifecycle of process application development.

In fact, studies on the lifecycle of process development have been reduced and are almost inexistent. In 1996, Sheth et al. [3] established that workflow and process modeling was one of the outstanding research issues which should be investigated.

The lifecycle of workflow applications development is comparable to the lifecycle of software development [4]. The use of adequate methodologies to assist the lifecycle of processes development is a key determinant to the success of any workflow project and requires the availability of specific tools – different from the ones used in software development – to model each phase of the cycle.

Our work has started with the use of expressive graphical process modeling languages (such as STRIM [5]) to model workflows [6]. In this paper, we argue that better methodological support for stepwise creation of Web processes and workflows that can ensure the fulfillment of business processes' strategic goals is necessary.

Our work targets the development of a framework to assist process analysts and designers to model business processes and design workflows. The framework is to be used during the analysis and design phase. It can be viewed as a methodology which structures a comprehensive set of steps that drives the design of workflows based on requirements gathered from communication with staff, managers, and domain experts.

This paper is structured in the following way. Section 2 presents the requirements of our framework. In section 3, we present and describe our framework. Finally, section 4 presents our conclusions.

2. Framework Goals

Practitioners, consultants, as well as academics, have differing views about business process and workflow development. Some organizations view workflow development as an ad-hoc activity to archive the automation of a few manual procedures, others view it as the improvement or redesign of isolated business processes, and only a minority view it as a comprehensive process re-organization, and use methodologies, lifecycles, and modeling tools to decompose organization's ongoing activities into a well defined set of workflows.

Workflow modeling lifecycle is composed of various phases, including analysis, design, implementation, testing, and maintenance. The number of phases and the phases themselves are not structured in a rigid manner. Therefore, several methodologies can be used for workflow development, comparable to the water fall model, spiral model, and rapid prototyping model.

In our study, we are particularly interested in two phases: analysis and design. Each phase includes a set of different perspectives that needs to be considered when developing a framework for workflow analysis and design.

Our goal is to supply a framework to assist workflow developers in their task, independently of the methodology used for workflow development. Our framework is a basic conceptional structure composed of steps, procedures, and algorithms that determine how process analysis and design is to be approached.

3. The Framework

The intention of this section is to give an overall description of our framework to construct workflows based on the knowledge gathered from interviews, group brainstorming sessions, and meetings (in this paper we will use the term 'interview' to designate these three methods of gathering knowledge).

The interviews are essentially carried out between process analysts and people who have the expertise and knowledge of the processes' business logic. The latter group might, typically include people such as administrative staff, department managers, mid-range managers, and even CEOs.

The input of the framework presented in this paper is a set of task names, and the output is a workflow. The workflows include tasks or Web services, transitions, control flow variables, and control flow conditions. The framework relies heavily on interviews to supply the knowledge which cannot be inferred automatically.

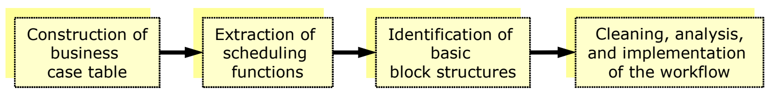

Fig. 1. The four steps of the framework

The framework has four major steps which are discussed individually in the following sections. These steps are the construction of business case table, extraction of scheduling functions, identification of basic block structures, and the cleaning, analysis, and implementation of the workflow. The phases are carried out sequentially as illustrated in Fig. 1.

3.1 Business Case Table Construction

The basic property of a process is that it is case-based [7]. This means that every task is executed for a specific case. Use cases have long been advocated for business process design as well as software design [8]. To capture all the cases represented in a process, we introduce the concept of business case table. The table has the main advantage of being a simple, yet powerful, tool to capture and describe business cases.

In the first phase, by means of interviews, we build a business case table. Each business case corresponds to an entry in the table and establishes the task scheduled at runtime based on business variables assertion. Business variables are variables that influence the routing or control flow in a process. For example, in a banking Web process application, the business variable Loan Amount determines the acceptance or rejection of a loan request. If the variable has a value greater than $500.000, then the loan is rejected and the task '*reject*' is executed, otherwise the task *'accept'* is executed.

A business case table is based on a two dimensional table. The schema of the table is the following. The columns are divided into two classes. The first class regroups a set of business variables, while the second class includes the tasks that are part of a process. Each entry of the table relates business variables and tasks with information indicating if a task is to be scheduled at runtime or not.

The first cells of each row, corresponding to the columns of the first class, contain values that can be assigned to business variables. The data cells corresponding to the columns of the second class contain information indicating if a particular task is to be scheduled at runtime or not. The idea is to establish if a given task is to be scheduled based on the assertion of a set of business variables. Formally, we are interested in evaluating for each $task_t$ the following function, where bv_i is a business variable:

$$scheduled(task_t, bv_1, bv_2, \ldots, bv_n) \in \{\checkmark, \times\} \quad (1)$$

A data cell corresponding to the columns of the second class may contain the scheduled symbol (✓) or the not-scheduled symbol (×).

Understanding the business case table schema is relatively easy, whereas its construction is far more challenging and complex. The methodology (described in [9]) to construct and fill the table with business cases is an iterative and interactive process carried out through interviews. Its construction requires the involvement of designers and managers. For each row and for each column of the second class, it should be inquired the symbol of function (1). Example of a business case table is shown in Table 1.

Table 1. Example of a business case table

Variable names							Task names					
Traveler	Location	Funding Source (Payment)	Person Filling the Form	Traveler/User is a CWA Mgr. for that trip?	Traveler/User is a M&CT Prog. Mgr. for that trip?	Is CWA and M&CT same person?	Fill Form (Traveler,OA or User)	Check Form (OA)	Confirmation (Traveler)	Sign (CWA Mgr.)	Inform (Program Manager)	Info (1st L Mg
1st Line Manager	Foreign	Boeing	Traveler	yes	yes	yes	✓	✓	×	×	×	[illegible]
						yes	✓	✓	×	×	×	[illegible]
					no	no	✓	✓	×	×	✓	[illegible]
						no	✓	✓	×	×	✓	[illegible]
				no	yes	no	✓	✓	×	✓	×	[illegible]
						no	✓	✓	×	✓	×	[illegible]
					no	yes	✓	✓	×	✓	×	[illegible]
						no	✓	✓	×	✓	✓	[illegible]
				yes	yes	yes	✓	✓	✓	×	×	[illegible]
						yes	✓	✓	✓	×	×	[illegible]
					no	no	✓	✓	✓	×	✓	[illegible]

Symbol Conflicts. One important step when constructing a business case table is resolving symbol conflicts. Symbol conflicts indicate that the scheduling of a set of tasks depends on one or more business variables. This can be verified when the two available symbols have been assigned to the same data cell in the business case table. To resolve a symbol conflict, the process analyst – with the help of interviewees – should identify at least one business variable that controls the scheduling of a conflicting task. When such a variable is identified the following steps are taken:

1. A column is added to the left side of the business case table and rows are added to the table.

2. The column is labeled with the name of the business variable identified.
3. Each row of the table is duplicated *n-1* times, where *n* is the domain set cardinality of the newly introduced business variable.
4. The data cells corresponding to the new business variable column are set to the values of its domain.

Once the table's schema is updated to reflect the introduction of a new business variable, the data cells must also be updated with appropriate scheduling symbols. As previously, the process analyst should carry out (additional) interviews to determine which tasks are scheduled at runtime based on the business variables present in the table.

Quality of Service. One important requirement of business processes is the management of Quality of Service (QoS). During the construction of a business case table, the business analyst and domain expert set QoS estimates for each task. The estimates characterize the quality of service that a task will exhibit at runtime. Quality of service can be characterized according to various dimensions. In our framework, we have used a QoS model [10] composed of the following dimensions: time, cost, and reliability. The information will be used in a latter phase to compute the QoS of the overall business process.

3.2 Extracting Scheduling Functions from the Business Case Table

In the second phase, we extract a set of scheduling functions from the business case table. For each task, a scheduling function that rule the scheduling of tasks is extracted (see equation 1). A scheduling function is a Boolean function for which the parameters are business variables from the business case table. Each function models the scheduling of a task at runtime, i.e. for a given set of business variables and their assertion, the function indicates if a task is scheduled at runtime or not.

To extract a set of scheduling functions, we first need to map a business case table to a truth table. The mapping can be achieved in the following way.

- For each business variable determine the minimum number of bits *mnb* necessary to represent the variable. Represent each bit with a different binary variable (for example, 'a', 'b', 'c', …).
- Create a mapping between each business variable value and a binary number, starting with '0'. Each business variable value has *mnb* bits and can be represented with a sequence of binary variables, for example, 'ab' or '/ab' (the symbol / indicates negation).
- Map the symbols ✓ and ✗ to the Boolean domain {0, 1}. The symbol ✗ is mapped to '0' and the symbol ✓ is mapped to '1'.
- Create a new table using the two mappings described previously.

Once the mapping is done, we can extract scheduling functions from the truth table. he extracted functions are logic disjunctions of conjunctions of business variables. Two methods can be used to generate the functions: Karnaugh maps [11] and the Quine-McCluskey [12] method.

Table 3 shows a scheduling function table which was constructed based on the data present in a business case table. The table is composed of three business variables ‘a’, ‘b’, and ‘c’. We have selected simple letters to represent business variables to simplify the handling of the truth table.

Table 2. Scheduling table constructed from a business case table

Task	Scheduling Function
Check Form	1
Sign	/a/b
User Reservation	/a/bc
Send Tickets	/a/bc
Reject	/a/b/c
Notify Manager	/a/b
Book Flight	/ab
Book Hotel	/ab
Not Authorized	a/b
Notify	1

The Quine-McCluskey method is particularly useful when extracting scheduling functions with a large number of business variables. Additionally, computer programs have been developed employing this algorithm. The use of this technique increases the degree of automation of our methodology. Remember, that this was one of our initial goals.

3.3 Identify Basic Block Structures

Business process management systems are process-centric, focusing on the management of flow logic. Most workflow languages are able to model sequential, parallel, and conditional routing which are modeled with standard structures such as *and-split*, *and-join*, *or-split*, and *or-join* [13]. Tasks associated with sequential and parallel building blocks are executed in a deterministic fashion, while conditional blocks are examples of non-deterministic routing. Conditional blocks indicate that the scheduling of a task depends on the evaluation of a Boolean condition.

This third phase consists of using the scheduling functions from the previous phase and identifying the sequential, parallel, and conditional building blocks that will make up the process in development. This phase is composed of two major steps:

- Identify sequential and parallel building block associated with a process and
- Organize these basic blocks using conditional building blocks.

Identifying Sequential and Parallel Structures. The objective of the first step is to identify sequential and parallel structures, and define a partial order for the tasks associated with these structures. To complete this step, the following activities are performed:

a) Create a set S of sets s_i, where each set contains all the tasks that have the same scheduling function,
b) Label each set with its scheduling function,
c) For each set, establish existing sequential and parallel building blocks, set a partial order for the tasks

In the first activity, we produce a set S of scheduling sets s_i, where each set s_i contains all the tasks that have the same scheduling function. The idea is to create sets of tasks with the following property: if a task of set s_i is scheduled at runtime, then all of the tasks in s_i are also scheduled. The second activity associates each set with a scheduling function label. Finally, the last activity establishes the sequential and parallel building blocks and defines a partial order for each set s_i. Each set s_i can be organized using a sequential and/or a parallel basic building block structure. Fig. 3 shows an example of the diagrammatic representation of the sets created with their scheduling functions.

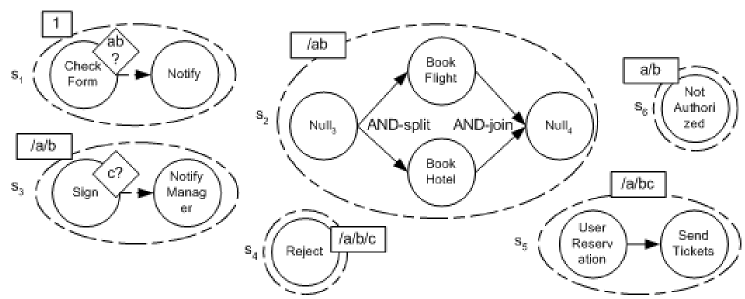

Fig. 2. Parallel and sequential block structures and partial orders for the sets s_i

Conditional structures cannot occur for the sets s_i since non-determinism has already been captured with the scheduling functions.

The establishment of sequential and parallel building blocks and partial orders may require the use of *null* tasks (also known as *dummy* tasks). A null task does not have a realization. Null tasks can be employed to modify a process to obtain structural property (e.g., well-handled and sound) or to make possible the modeling of specific business process procedures.

Identifying Conditional Structures. At this point, we have already identified the sequential and parallel building blocks. The next step is to construct a task scheduling graph based on the scheduling sets s_i. The aim of the graph is to identify the conditional building blocks of a process and determine how they control and organize the scheduling sets previously recognized (i.e. sequential and parallel building blocks). A set of assumptions and rules are used to structure scheduling sets into a process graph. The algorithm, assumptions, and rules used to identify conditional structures and construct the workflow are described in [9]. An example of the resulting process graph after applying the algorithm is shown in Fig 4.

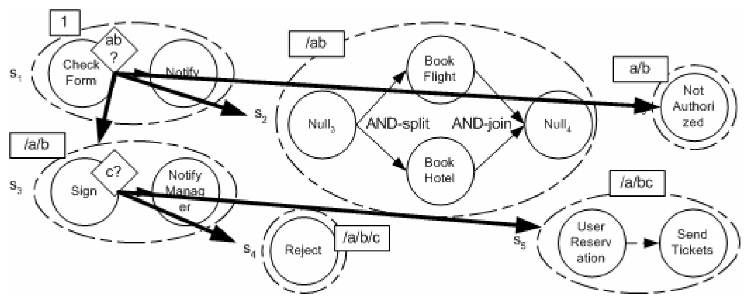

Fig. 3. Task scheduling graph

Nevertheless, several workflow elements are missing. It is apparent in our example that the workflow does not include any joins matching the *or-splits* and that the workflow has several ending points.

Both problems can be solved by matching *or-splits* with *or-joins*. Aalst [14] has pointed out the importance of balancing *or/and-splits* and *or/and-joins* to obtain what is called a 'good' workflow. For example, two conditional flows created via an *or-split*, should not be synchronized by an *and-join*, but an *or-join* should be used instead. Matching *or/and-splits* may require the use of *null* tasks.

Setting Probabilities for Transitions. In order to enable the analysis of workflow QoS, it is necessary to initialize task QoS metrics (the step was completed during the business case construction) and initialize stochastic information which indicates the probability of transitions being fired at runtime.

The process analyst – with the help of interviewees – needs to associate conditional transitions with a probability between 0 and 100, i.e., each transition that connects two sets s_i of the task scheduling graph needs to be associated with a probability. The sum of the probabilities of the outgoing transitions of a set s_i needs to be 1. These values are only estimates and can later be recomputed and updated according to the workflow execution.

3.4 Cleaning, Analyzing, and Implementing the Workflow

In the last phase, we cleanup of any dummy (null) tasks and, if necessary, the workflow may be slightly restructured or modified for reasons of clarity.

Since QoS estimates for tasks and for transitions have already been determined, we can now use several techniques to analyze workflow QoS. Mathematical methods, such as the Stochastic Workflow Reduction (SWR) algorithm [15], and Simulation [16] can be used to compute overall QoS metrics for a workflow. Alternatively, the workflow can be converted and analyzed using Petri nets and Petri nets analysis tools [7].

Once the cleaning and analysis are completed, the process design is ready to be implemented. The method proposed in [17] can be used to this end. Their method, targeting more technical aspects, includes the selection of the target workflow system and the mapping of graphical diagrams describing a business process at a high level into a process specification.

4. Conclusions

Although major research has been carried out to enhance workflow systems, the work on workflow application development lifecycles and methodologies is practically inexistent. The development of adequate frameworks is of importance to guarantee that workflow are constructed according to initial specifications.

Unfortunately, it is recognized that despite the diffusion of workflow systems, methodologies and frameworks to support the development of workflow applications are still missing. In this paper, we describe a framework to assist process analysts during their interviews with administrative staff, managers, and employees in general to design workflows.

The core of the framework presented has been employed successfully to design a small size process. We believe that the framework is also appropriate to design larger size workflows and that it represents a good step towards the modeling of business processes.

References

1. Sheth, A.P., Aalst, W.v.d., and Arpinar, I.B. (1999): Processes Driving the Networked Economy. *IEEE Concurrency*, **7**(3): p. 18-31.
2. Cardoso, J. and Sheth, A. (2003): Semantic e-Workflow Composition*. Journal of Intelligent Information Systems (JIIS)*. **21**(3): p. 191-225.
3. Sheth, A., Georgakopoulos, D., Joosten, S., Rusinkiewicz, M., Scacchi, W., Wileden, J., and Wolf, A. (1996): *Report from the NSF Workshop on Workflow and Process Automation in Information Systems*. Deptartment of Computer Science, University of Georgia: Athens, GA.

4. Sommerville, I. (2000): *Software Engineering*: Addison-Wesley Pub Co, 2000. 693.
5. Ould, M.A. (1995): *Business Processes: Modelling and analysis for re-engineering and improvement*. Chichester, England, 1995.: John Wiley & Sons, 1995.
6. Cardoso, J. and Teixeira, J.C. (1998): Workflow Management Systems: A Prototype for the University of Coimbra. in *5th International Conference on Concurrent Engineering*. Tokyo, Japan. p. 83-87
7. Aalst, W.M.P.v.d. (1998): The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, **8**(1): p. 21-66.
8. Jacobson, I. (1995): *The Object Advantage: Business Process Reengineering With Object Technology*: Addison-Wesley, 1995.
9. Cardoso, J. (2003): *Poseidon: A framework to assist Web process design based on business cases*. University of Madeira: Funchal. p. 21.
10. Cardoso, J., Sheth, A., and Miller, J. (2002): Workflow Quality of Service. in *International Conference on Enterprise Integration and Modeling Technology and International Enterprise Modeling Conference (ICEIMT/IEMC'02)*. Valencia, Spain, Kluwer Publishers
11. Karnaugh, M. (1953): The Map Method for Synthesis of Combinational Logic Circuits. *Transaction IEEE*, **72**: p. 593-599.
12. McCluskey, E.J. (1956): Algebraic minimization and the design of two-terminal contact networks. *Bell System Technical Journal*, **35**: p. 1417--1444.
13. Aalst, W.M.P.v.d., Barros, A.P., Hofstede, A.H.M.t., and Kiepuszeski, B. (2000): Advanced Workflow Patterns. in *Seventh IFCIS International Conference on Cooperative Information Systems*. p. 18-29
14. Aalst, W.M.P.v.d., Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques, in *Business Process Management: Models, Techniques, and Empirical Studies*. 2000, ed. A. Oberweis, Springer-Verlag: Berlin. p. 161-183.
15. Cardoso, J. (2002): *Stochastic Workflow Reduction Algorithm*, LSDIS Lab, Department of Computer Science, University of Georgia. http://lsdis.cs.uga.edu/proj/meteor/QoS/SWR_Algorithm.htm
16. Miller, J.A., Cardoso, J.S., and Silver, G. (2002): Using Simulation to Facilitate Effective Workflow Adaptation. in *Proceedings of the 35th Annual Simulation Symposium (ANSS'02)*. San Diego, California. p. 177-181
17. Casati, F., Fugini, M., Mirbel, I., and Pernici, B. (2002): WIRES: a Methodology for Designing Workflow Applications. *Requirements Engineering Journal*, **7**(2): p. 73-106.