Addressing Dynamism in E-negotiations by Workflow Management Systems

Sarita Bassil[1] Morad Benyoucef[1] Rudolf K. Keller[2] Peter Kropf[1]
[1]*Département IRO, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Québec, H3C 3J7, Canada, {bassil, benyouce, kropf}@iro.umontreal.ca*
[2]*Zühlke Engineering AG, Wiesenstrasse 10a, CH-8952 Schlieren, Switzerland, ruk@zuehlke.com*

Abstract

Workflows (Wfs) are a major enabling technology for e-commerce. In our research, a Combined Negotiation (CN) is modeled and enacted using Wf technology. The modeling task captures the sequencing of the individual negotiations as well as the dependencies between them, and the enacting task runs the model. A CN support system (CONSENSUS) is used by the user to perform both tasks. Supporting dynamic modifications to the model during run-time should increase the benefits of our approach. In this paper, we highlight the need for such support by identifying the dynamic aspects that can occur while negotiating the different items of a package (i.e., the CN object). To address these aspects, we experimented using ADEPT, a Wf Management System supporting dynamism. This leads us to discuss the Wf Reference Model of the Wf Management Coalition, and suggest a "dynamic" extension to the current functional areas and architecture.

1. Introduction

Sourcing is defined by a set of steps related to the procurement of products/suppliers [13]: identification of requirements and suppliers, evaluation of market dynamics, negotiation, and configuration.

Negotiation is of utmost importance in the context of sourcing; furthermore, it is required for effective support of B2B interactions. Many negotiation types are practiced in B2B markets, some of which are described in [13]. Among them, the multivariable negotiation is based on multiple parameters beyond just price, but including quality, delivery, warranty, and financial terms. The multi-stage negotiation consists of an RFI, an RFP, and an RFQ (Request For Information, Proposal, Quotation). In this type of negotiation, the buyer creates online requests for bids, the suppliers bid online, and then the buyer analyzes the bids [2]. Interactions between buyer and supplier are referred to as a multi-threaded negotiation. Furthermore, the finalization of purchases is not made without due consideration of availability/cost of services required to deliver these products. This latter negotiation is known as a synchronous supply chain negotiation.

Sourcing is an interesting framework that provides opportunities for procurement savings. Taking into account the number of companies that have been highly successful in achieving benefits through B2B sourcing [2], we chose to consider a sourcing application as our running example throughout the paper.

The "importing package" example described below stems from the area of transportation. In fact, importing goods is a complex procedure in which a buying company is involved in a number of activities/services such as the purchase, the shipment (the term "transportation" is sometimes used in this paper), the insurance, and the forwarding of goods. These services are obviously interrelated and hence, we can imagine a combined negotiation (CN) model as described in [4]. Furthermore, many constraints exist as well, which can also be encompassed by the CN model. Here are some of the constraints that are likely to be involved: the maximum price the buying company is willing to pay for the goods, the quantity needed, the terms of payment, the delivery date, the packaging of the goods. Regarding this latter aspect, we can imagine for instance the importation of cement in bulk or in bags. With regard to the shipping service, which may include inter-modal transportation, a number of scenarios are possible. The supplier can cover the freight shipment and insurance from warehouse of origin to warehouse of destination. Another alternative is to let the buyer cover all charges. In this latter scenario, a constraint might be for instance to find a truck with a suitable arrival (resp. departure) time to port of shipment (resp. from port of destination), taking into account the vessel loading (resp. unloading) time. The buying company could have preferences for specific shipping companies, and may also specify the maximum amount to be spent for each shipment phase, as well as the total amount for the whole shipping. As for the insurance,

The completion of this research was made possible thanks to funding provided by the NSERC (CRD-224950-99), Bell Canada's support through its Bell University Laboratories R&D program, and support by the CIRANO.

the buying company could also have some restrictions regarding the insurance companies, the kind of insurance, the price to pay, etc. Dependencies may exist between items. Indeed, a special kind of insurance could be preferred while a specific packaging of goods is considered.

Suppose that the complementary (i.e., cannot have one without the other) goods and services discussed above are negotiable (keeping in mind that the B2B sourcing techniques RFP and RFQ presented above, are negotiation types), and that the buying company chooses to engage in different negotiations for the goods and services, trying to make the best deal with respect to its interests. The need for a support system to solve a CN has already been motivated within several papers [4, 5]. Unfortunately, this solution is heavily based on a static Workflow Management System (WfMS). This latter can be defined as a software that manages a workflow (Wf) efficiently by tracking and controlling its execution. It supports the definition, the execution and the monitoring of a Wf. The static aspect of such systems comes from the fact that the system does not provide functionalities allowing the user to change the Wf instance during execution (e.g., by inserting a new task, by deleting a task already present).

However, the key to make CN support more effective is to ensure that the buying company is working with WfMSs that are flexible enough during run-time, to accommodate the various internal (coming from the buying company) or external (coming from the supplier and/or the service companies) contingencies and/or obstacles that can appear.

Indeed, while negotiating, the supplier, the forwarding agent or even the shipping company can make a new offer that might interest the buying company, requiring this latter to review one or multiple negotiation tasks already planned within the Wf model. If a strike reduces the supplier's activities, the buying company may choose for instance to give up the CN (i.e., cancel it altogether). Finally, the buying company may change its mind concerning a specific attribute (e.g., the maximum price to pay for a specific item). It should be possible to re-set the predefined attributes related to each negotiation task; of course this should be done before launching the task.

The example above illustrates the important role of dynamism in e-negotiations, and especially in CNs. In this paper, we present the need for and nature of dynamism as required by e-negotiations. Furthermore, we show to which extent state-of-the-art WfMS technology is able to support such dynamism, and we outline how systems should be extended to cope with this need.

Section 2 of this paper addresses process modeling in general and discusses the modeling of our "importing package" using a static WfMS. In Section 3, we discuss dynamism, first by identifying a number of dynamic scenarios in our CN example; then by demonstrating that ADEPT [12] – a WfMS supporting dynamism – is fit to cope to some extent, with dynamism in CN field; and finally by reviewing the Wf Reference Model [17], proposing a certain support for dynamism. Section 4 briefly discusses related work, and Section 5 concludes the paper.

2. Modeling CNs

2.1. Process modeling

The use of CONSENSUS to support the user in conducting a CN has already been discussed in [4]. The system includes a WfMS to model the sequencing of the negotiations (represented as tasks) as well as the dependencies between them. It also allows to specify the different attributes related to each negotiation. The modeling phase (i.e., build-time phase) within CONSENSUS ensures to conduct the CN according to a well-thought plan. However, the enactment phase as it is implemented now, allows running the CN, and hence the Wf instance, in a *static* manner, only. The limitations related to this approach will be discussed later in this paper.

Let us consider and discuss in this section the conceptual modeling in general. In fact, a model is an abstract representation of reality that excludes much of the world's infinite detail. Curtis *et al.* [7] specify that a model reduces the complexity of understanding or interacting with a phenomenon by eliminating the detail that does not influence its relevant behavior. However, many essential forms of information must be kept to adequately describe a model. A number of goals for (process) models are discussed in [7], and range from comprehensibility to enactability taking into account the automation of execution.

To the extent that automation is involved, process modeling becomes a vital issue in redesigning work and allocating responsibilities between humans and computers. A CN is indeed a complex process, and modeling gives a visual representation, which is easily understandable by humans, and identifies and formalizes all the necessary items of the CN. This may be helpful in a prospective evolution or modification of the current negotiation items, their sequencing and the dependencies between them. Modeling the CN also incites to reason about its variables and attributes. Further details about modeling, and CN modeling issues can be found in [5].

2.2. Modeling our CN example using WLPI

WLPI (WebLogic Process Integrator) is a WfMS from BEA systems [3]. The Studio unit of WLPI is a client application that we used to model our running example. The CN model is created using a graphical tool integrated within the Studio. This model is stored in the WLPI database, and the Studio monitors its execution. A Worklist unit is another client application used to create and start instances of the model. WLPI is considered a static WfMS since it does not provide any dynamism during execution.

Figure 1 shows our running example created using WLPI Studio. Tasks are the core process activities of a Wf, and they evolve through various states (created, activated, executed, and marked done) as the Wf progresses.

In our example, the buying company has to take a decision regarding the number of negotiations that should be launched for the purchase of the goods. These tasks could be initiated at the same time, but only one deal should be struck. The next step will be to start negotiations for the shipment services. We choose to begin by negotiating the sea shipment, and then the two surface shipments (from warehouse of origin to port of shipment, and from port of destination to warehouse of destination). The reason why the sea shipment is negotiated first is that surface transportation is usually more flexible and available. It will hence be easier to schedule the truck arrival (resp. departure) time to port of shipment (resp. from port of destination) with respect to the vessel loading (resp. unloading) time (than to do it in the opposite way). The insurance and the forwarding negotiations are planned in sequence as the last two items of the model. (For the sequencing, see Figure 1.)

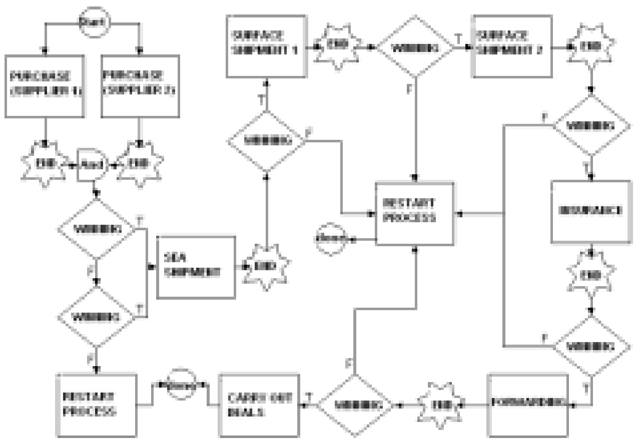

Figure 1. "Importing package" Wf model in WLPI

When we fail to make a deal on an item, after concluding deals on other complementary items, we talk of "exposure". To avoid exposure, we would have to restart the whole process ("Restart Process" task in Figure 1) by renegotiating some (or all) of the attributes of the deals already made. For instance, if we fail to find suitable transportation for a given date (fixed in a previous deal), then we could go back and re-discuss the delivery date with the supplier. In the worst case, this procedure could lead us to breaking our commitments.

A dynamic Wf can offer a better solution in such a case. Indeed, instead of restarting the whole process, a new negotiation could be launched for transportation with another shipping company, by inserting this new task into the Wf instance while it continues to run.

The "Restart Process" task is one of the ways that we can use to remedy the lack of flexibility during run-time. In the next section, we present some other scenarios highlighting dynamism, and we try to address one of them using a dynamic WfMS (ADEPT).

3. Addressing dynamism

3.1. Dynamic aspects in our CN example

Although it is widely recognized that WfMSs should provide flexibility, most of today's systems unfortunately have problems dealing with changes. However, new offers, contingencies, and obstacles that can appear during negotiation, may require modifications of the Wf instance.

Indeed, taking into account our example, an obvious dynamic change could derive immediately after negotiating the purchase of the goods. The supplier can offer for instance, to cover the freight shipment and insurance from warehouse of origin to warehouse of destination. The buying company could be interested in this offer, and hence decides not to engage in any of the following steps of the CN (i.e., transportation, insurance, forwarding). It should be possible for the buying company to remove all these steps from the instance of the Wf during run-time.

Obviously, a similar offer could also come from the forwarding agent. In this case, the buying company might find it interesting to engage in the negotiation with the forwarding agent in parallel with the transportation, thus the possibility to move (or to delete and to re-insert) the forwarding task right after the purchase of goods tasks is wished for. In case the negotiation with the forwarding agent succeeds (covering the freight shipment and insurance), a next step would be to delete all the negotiation tasks related to transportation and insurance.

Among other possibilities, the two dynamic scenarios described above could occur in a real-world importing process. Hence, it would be advantageous for a CN to allow on-the-fly changes (i.e., make changes while an instance of the Wf model is running).

3.2. Dynamic modifications using ADEPT

ADEPT (Application Development based on Encapsulated pre-modeled Process Templates) is a WfMS from the University of Ulm [12]. It offers support for some dynamic changes, giving the possibility, at run-time, to deviate from the pre-modeled task sequence. The ADEPT Wf-Editor is a build-time client application for modeling tasks. As with WLPI, the Wf model is stored in a database. The ADEPT Client monitors the execution of a model, allowing the user to intervene by inserting or deleting a task to the instance already created and launched. The task to insert should exist in one of the instances already created (even the ones related to a different Wf model). It is not permitted to define/model a new task during run-time. Moreover, a certain number of constraints must be satisfied before proceeding to the modification steps, and no modification is allowed while a specific task is running.

We used ADEPT to model and run the "importing package", in order to address the issue of dynamism in CNs. Two main criteria were applied to retain this system among the few promising prototypes (see Section 4) that have recently emerged to deal with flexibility. Indeed, the first criterion is related to the interest granted to this system within the literature [8, 15], whereas the second criterion concerns its availability.

Figure 2 shows our running example within the ADEPT Client (Monitoring unit), based on the second scenario described in Section 3.1. Boxes in Figure 2(a) represent tasks that correspond to the different negotiations of the "importing package" as shown in Figure 1. A "start" node (S), an "end" node (E), and a "carry out deals" task (node C) are added. Two "empty" nodes are used for the and-split and the and-join of the "purchase (supplier 1 *and* 2)" tasks (nodes S1 and S2). Inserting a task to the current instance requires synchronization with tasks that must be completed before and after the inserted one. In our example, the "forwarding" task (node F) should be activated after the two "purchase (supplier 1 *and* 2)" tasks, and obviously before the "carry out deals" task. The edge from node S1 (resp. S2) to node F, and the one from node F to node C in Figure 2(b) show the synchronization. Figure 2(c) depicts the case where the negotiation with the forwarding agent succeeds; all the remaining negotiations related to transportation (nodes T1, T2, and T3) and insurance (node I) are deleted (even the already activated task "sea shipment" – node T1), going straight to the "carry out deals" task. Note that the two tasks "forwarding" and "sea shipment" are activated in parallel; however, the "forwarding" task is completed first.

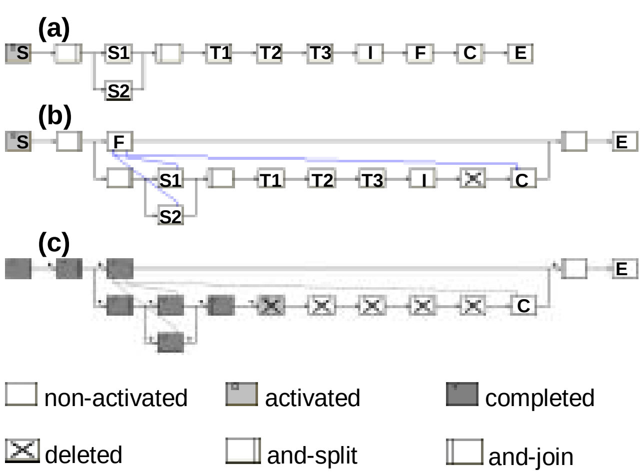

Figure 2. "Importing package" during run-time in ADEPT. Instance state (a) after creation, (b) after moving (i.e., deleting and inserting) task F, (c) after deleting tasks: T1, T2, T3, and I

From our experience with ADEPT, we realized that defining decisions makes our model less flexible to deal with dynamic modifications during run-time, since it is not easy to adjust a moved task (we talk of moving a task when we delete a specific task, and insert it in a different place, e.g., "forwarding") with its corresponding decision branch. Consequently, we chose to model our CN example without decision branches letting the user decide manually whether to go for the next negotiation in the sequence, to delete specific task(s), or even to insert new one(s); obviously this should take into account the negotiation results (e.g., deal or not, fixed delivery date) of the previously completed task. Furthermore, allowing automatic launching of tasks (as a result of a decision branch) reduces the opportunities for user intervention, since no modification is allowed while a certain task is running. The previous argumentations suggest that in order to offer a more flexible model (i.e., model that supports more dynamic modifications), we need to define less automatic tasks, avoiding for instance decision branches.

In the case of dependent attributes between tasks (e.g., an item needs the result of a predecessor item as input), ADEPT does not allow to delete the producing task. This is perfectly coherent. However, since it is not allowed to modify attributes – mainly to delete the pre-affected attributes to the consuming task – it makes, once again, our model less flexible regarding deletion. Although we did not experiment this with our running example, we are pretty sure that dependent attributes may easily appear in CNs. We think that a possible solution could be to permit dynamic modification of attributes.

3.3. Workflow Reference Model and dynamism

The Wf Management Coalition (WfMC) [14] has developed an overall model for Wf systems. This model, called the Wf Reference Model, identifies the components of WfMSs, thus enabling individual specifications to be developed within its context. At the highest level, *all* WfMSs may be characterized as providing support in three functional areas [17]: (1) the build-time functions, concerned with defining, and modeling the Wf process and its activities; (2) the run-time control functions, concerned with managing the Wf processes in an operational environment and sequencing the activities to be handled as part of each process; and (3) the run-time interactions with human users and IT applications for processing the various activity steps. The WfMC specifies that *some* WfMSs may allow dynamic alterations to process definitions from the run-time operational environment [17]. Since the run-time operational environment is involved within the second and third functional areas, a WfMS supporting dynamic alterations could be seen, from our point of view, as a system that extends these two functional areas by a set of run-time "process modification functions" that allow users to modify instances of the original model with minimum effort.

At a lower level, the architecture of the Wf Reference Model identifies five interfaces [17]: Wf definition tools, Wf client applications, invoked applications, Wf enactment services, administration and monitoring tools. These interfaces are related to the Wf enactment service of the WfMS, and they are supported by a set of operations (i.e., API calls). These operations are gathered within a number of groups (e.g., session establishment, Wf definition operations) distributed within the five interfaces stated before.

A Wf client application is defined as the one supporting interactions with user interface desktop functions. Consequently, a possibility to allow dynamism in a WfMS is to add to this interface a set of operations for the creation, and the deletion of a particular object within a Wf definition, and even the setting, and the deletion of a particular object attribute. These operations are already covered by the current API of the WfMC (WAPI), and they are gathered within the "Wf definition object operations" group [17]. Hence, enabling dynamism could be seen as the addition of this group to the "Wf client applications" interface.

4. Related work

CN is a novel negotiation type [4], and CONSENSUS was probably the first Wf-based system to support it. Dynamism has widely been recognized as an important feature of WfMSs in general, but in our context, the inability to deal with it limits the benefits of the CONSENSUS approach. Currently, many researchers are working on problems related to dynamic Wfs [1, 8, 12, 15], however few commercial systems provide support for this kind of Wfs [10, 11]. The first work in the literature that motivated the issues of dynamic Wf change was [9]. Recently, Wf prototype efforts have emerged to address flexibility in Wf systems. These prototypes include ADEPT [12] (discussed in Section 3.2) and Milano [1]. Ellis and Keddara introduced rather a Modeling Language to support Dynamic Evolution within Wf Systems (ML-DEWS) [8]. As a final note concerning related work, we would like to highlight the fact that dynamism is an important aspect in numerous domains other than CN, such as software engineering [6].

5. Conclusion and future work

In this paper, we mainly showed the need for dynamism in CNs. We began by motivating the use of Wf technology for the modeling part of a CN. We then pointed to the lack of flexibility in static WfMS, and tried to address dynamism by way of a WfMS that supports dynamic modifications. Finally, we reviewed the Wf Reference Model recognizing the lack of support for dynamism in the current architecture, and proposing an extension for dynamic support within WfMSs.

As future work, firstly, we are looking at integrating ADEPT into the current prototype of CONSENSUS. This will lead to consider user intervention in the "Workflow Monitoring and Control Tool" of the current CONSENSUS architecture [4]. As a second direction, a wish list of dynamic modifications required for CNs should be produced. We need to be aware of the shortcomings and limitations of current dynamic WfMSs in respect to the modeling and running of CNs. These limitations should give us a valuable input for future versions of current dynamic systems.

6. References

[1] Agostini, A., and De Michelis, G., Improving Flexibility of Workflow Management Systems. In [16], 218-234.

[2] B2B Sourcing a Quick Hit, 2000. *Sourcing On an Online Exchange*. On-line at <http://www.ebizchronicle.com /backgrounders/november/b2bsourcing.htm>.

[3] BEA Systems WLPI. On-line at <http://www.beasys.com /products/weblogic/integration/index.shtml>.

[4] Benyoucef, M., Alj, H., Vézeau, M., and Keller, R.K., Combined Negotiations in E-Commerce: Concepts, and Architecture. *Electronic Commerce Research Journal*, 1(3):277-299, July 2001. Special Issue in Theory and Application of Electronic Market Design. Baltzer Science Publishers.

[5] Benyoucef, M., Bassil, S., and Keller, R.K., Workflow Modeling of Combined Negotiations in E-Commerce. In *Proceedings of the Fourth International Conference on Electronic Commerce Research (ICECR-4)*, 348-359, Dallas, Texas, November 2001.

[6] Cugola, G., Tolerating Deviations in Process Support Systems via Flexible Enactment of Process Models. *IEEE Transactions on Software Engineering*, 24(11), 1998.

[7] Curtis, B., Kellner, M.I., and Oliver, J., Process Modeling. *Communications of the ACM*, 35(9):75-90, September 1992.

[8] Ellis, C.A., and Keddara, K., A Workflow Change Is a Workflow. In [16], 201-217.

[9] Ellis, C.A., Keddara, K., and Rozenberg, G., Dynamic Change within Workflow Systems. In *Proceedings of the Conference on Organizational Computing Systems*. ACM Press, 10-21, New York, 1995.

[10] FileNet. Ensemble User Guide. *FileNet Corp.*, Costa Mesa, California, 1998.

[11] InConcert. InConcert Process Designer Guide. InConcert, Inc., Cambridge, Massachusetts, 1997.

[12] Reichert, M., and Dadam, P., ADEPTflex: Supporting Dynamic Changes of Workflow without Losing Control. *Journal of Intelligent Information Systems*, 10(2): 93-129, 1998.

[13] Sourcing: The key to competitive advantage. *Powerpoint Presentation from Aberdeen Group*, 2000. On-line at <http://www.aberdeen.com/>.

[14] The Workflow Management Coalition. On-line at <http://www.wfmc.org/>.

[15] van der Aalst, W., and van Hee, K., Workflow Management: Models, Methods, and Systems. *The MIT Press*, 2002. ISBN 0-262-01189-1.

[16] van der Aalst, W., Desel, J., and Oberweis, A. (Ed.), Business Process Management: Models, Techniques, and Empirical Studies. *LNCS 1806 Spring-Verlag*, 2000. ISBN 3-540-67454-3.

[17] Workflow Management Coalition, The Workflow Reference Model. *WFMC-TC-1003*, January 1995, 1.1. On-line at <http://www.wfmc.org/standards/docs/tc003v11.pdf>.