

CROSS-ORGANIZATIONAL WORKFLOWS: A CLASSIFICATION OF DESIGN DECISIONS *

Pascal van Eck,¹ Rieko Yamamoto,² Jaap Gordijn,³ Roel Wieringa¹

¹*Department of Computer Science, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands.*

vaneck,roelw@cs.utwente.nl

²*Fujitsu Laboratories Ltd., IT Core Laboratory
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki
Kanagawa, 211-8588, Japan.*

r.yamamoto@jp.fujitsu.com

³*Department of Computer Science, Vrije Universiteit Amsterdam
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands.*

gordijn@cs.vu.nl

Abstract Web service technology enables organizations to open up their business processes and engage in tightly coupled business networks to jointly offer goods and services. This paper systematically investigates all decisions that have to be made in the design of such networks and the processes carried out by its participants. Three areas of different kinds of design decisions are identified: the value modeling area, which addresses economic viability of the network, the collaboration modeling area, which addresses how business partners interact to produce the goods or services identified in the value modeling area, and the workflow modeling area, which addresses the design of internal processes needed for the interactions identified in the collaboration modeling area. We show, by reporting on a real-world case study, that there are significant differences between these areas: design decisions are unique for each area, IT support for collaboration processes is orthogonal to IT support for workflows, and the role of web choreography standards such as BPEL4WS differs for both of them.

Keywords: cross-organizational workflows, process design, web services

1. Introduction

Recently, a number of standards have been proposed for machine-readable specification of inter-organizational coordination processes, such as ebXML BPSS, BPEL, and WSCI (Alonso et al., 2004). Although these formalisms differ in many respects, they are all based on the same idea: they provide (and only provide) XML-based syntactic constructs to specify valid sequences of web service invocations of some business partners. Currently, not much is known about guidelines for actually designing cross-organizational processes. In this paper, we report on our research on these design guidelines. Our guidelines are compatible with any of the known web service choreography standards, because they should be used before the designer considers using one of these standards. Briefly, these decisions concern the commercial viability of the business activities being coordinated, the coordination mechanisms required to realize these commercial activities, and the workflows for each participating organization to support this coordination.

In the next section, we introduce the running example used throughout the paper. It is based on a real-life industry project (Yamamoto et al., 2004). In section 3 we describe our design approach and in the following sections, we describe the decisions made in each of the three views: commercial viability, coordination process, workflow processes.

2. Case study: A portal for music fans

Fans of a particular artist are interested in information about the artist, merchandise, and song scores, but also want to chat about their favorite artist. Our task is to build a *portal* that organizes information and services related to the artist. In March 2003, a Japanese project developed two such portals by integrating various e-services delivering content, merchandise and more, from different providers. The services provided included information about the singers' schedules, the sale of song scores, the sale of goods (T-shirts, pens, CDs, photographic prints, etc.), the management of fan clubs, the mailing of fan club news and birthday congratulations, and accounting for members' fees. As we will see, the portal is actually a constellation of enterprises, which collaborate to create a portal for a specific artist, with interesting services and content for fans.

3. Design decisions

Before committing considerable resources to creating a collaboration with another organization, an organization usually seeks answers to the following two questions: is the collaboration commercially viable (e.g., generating a positive cash flow), and is the collaboration feasible (e.g., does it not overextend

organizational and IT support change capabilities). We approach these questions by dividing them in three areas of concern: value modeling (economic viability), coordination modeling (feasibility/impact with respect to relations of organizations with each other), and workflow modeling (feasibility/impact with respect to internal structure and processes).

In the *value modeling* area, we address the enterprises and final customers that participate in the collaboration. As such, the value model presents who is offering *what of economic value* to whom and expects *what* in return. The latter refers to the notion of economic reciprocity; an important notion in commercial trade. In addition, the value model shows whether valuable objects are offered as a bundle (potentially by different suppliers) or not. *Bundling* (Choi et al., 1997) is an important notion in business to increase total sales and is in e-business settings of specific interest because information integration enable multi-supplier bundles. Finally, a value model shows the assignment of value activities (activities that yield profit) to performing actors. In the recent past, we have seen in the context of e-business many shifts of such activities from the one enterprise to another enterprise. All these design decisions are quantified in terms on revenues and expenses and are summarized in *profitability sheets*, representing the net cash flow for each actor involved. A discounted net cash flow analysis known from investment theory (see e.g. Horngren and Foster, 1987) can then be used to decide whether to invest in the collaboration or not.

We distinguish between on the one hand, coordination process by which business coordinate their behavior in a collaboration (*coordination modeling* area), and on the other hand, internal processes of each of the businesses participating in a collaboration (*workflow modeling* area). A *coordination process* consists only of interactions between two or more parties in the collaboration. These interactions involve externally visible behavior of each of the coordinating businesses. The set of all interactions of one business is called its *abstract business process*. In general, there will be one or more *internal business processes* that jointly realize the abstract process of a business. Most of these internal processes will be confidential, as it contains confidential business rules and uses confidential data. The distinction between coordination processes, abstract business processes and internal business processes is also made in the BPMN standard (White, 2004). A similar distinction is also made by van der Aalst and Weske, 2001.

Traditionally, when a business manager negotiates a collaboration agreement with other businesses, no technical knowledge is needed. However, when digital services are traded, the value network will be implemented as a computer network and almost all activities will be performed by software. In this case, the value model is a requirements specification for a network of communicating software systems, and the business manager needs the assistance of the

software requirements engineer to point out opportunities for allocating tasks to business actors, and for guarding the implementability of the value model. If a software requirements engineer would not participate in the negotiation about the value model, then the manager would unwittingly make software engineering decisions. Moreover, unrealizable parts of the value model would get changed later on in the implementation process, and those decisions would be made by software engineers, not by managers. Consequences of these changes for the value model would go unnoticed, and in effect a software engineer would then change the value model without giving a commercial motivation for this. By having a software requirements engineer participate in value modeling, we ensure that commercial decisions are made in the value modeling task only, and technical decisions are made in the software engineering task only.

4. Value modeling

Value modeling design decisions

We represent the outcome of value modeling decisions using the e^3 -value notation described in Gordijn and Akkermans, 2003; Gordijn and Akkermans, 2001. The purpose of the e^3 -value method is to represent enterprises (actors) who exchange things of *economic value* with each other, in an economically reciprocal way. Reciprocity expresses economic rationality: An enterprise offers something of value but requires something of value in return. e^3 -value contains tools to assess profitability for each actor. In previous work (van Eck et al., 2004), we identified a series of design decisions to be taken on the business value level:

- Which *consumer needs* do exist? A customer need is a state of felt deprivation of some basic satisfaction (Kotler, 1988).
- How are these consumer needs *satisfied* by *value objects*? Value objects are things that can be produced or consumed by enterprises and end-customers, and are by definition of *economic value*.
- Who is *offering/requesting* value objects to/from the environment? Each enterprise or end-customer produces/consumes value objects to create profit or to increase economic utility.
- What are the *reciprocal* value object exchanged between enterprise/end-customers? If an enterprise/end-customer requests a value object from its environment, what is offered in return for that.
- What *bundles* of value objects exist? Many e-business practices (Choi et al., 1997) are characterized by good or service (un)bundling. Bundling

refers to offering a set of value object as one, because of higher profits, or because of a better need satisfaction.

- What *partnerships* do exist? This closely refers to bundling as objects in a bundle are often offered by more than one enterprise.

Similar to a so-called Computation-Independent Model (CIM) in OMG's Model Driven Architecture (MDA), a value model completely abstracts from the internal structure of any software artifact in the domain. However, a value model seems to be broader than a CIM: in our opinion, a CIM focuses on a specific software artifact while a value model focuses on a constellation of networked businesses that may employ tens if not hundreds of software artifacts.

Case study value model

These design decisions can be represented by an e^3 -value value model (see Fig. 1). e^3 -value represents *actors* (enterprises and final customers) that exchange *value objects* (goods and services) with each other through *value interfaces*. These interfaces consist of *ports* offering or requesting value objects. Final customers have a customer *need*. To satisfy such a need, a set of value exchanges need to be executed by all enterprises collaborating in satisfying that need. We represent these exchanges by a *dependency path*. A dependency path consists of the need, the interfaces exchanging objects contributing to need satisfaction, and internal actor dependencies between interfaces. If an actor has a *need*, he will exchange objects of value through one of his interfaces to satisfy the need. Additionally, exchanges via an interface may cause exchanges via another interface of such an actor (e.g. to buy raw materials to produce to object requested). So, the focus is on *what* actors exchange of *economic value*, and not *how* they do so from a business process perspective.

The value model shows two portals for two artists, each of which offers services to fans of that artist. Each artist portal integrates a number of services offered by so-called function providers. Since each artist portal promotes music of a particular artist, it is paid for by the record companies of the artist. In order to use the Akinori Nakagawa portal, one has to be a member. In contrast, the Kitaro portal offers also some goods to non-members in addition to providing member services.

A member of the Akinori Nakagawa portal can do two things: (1) use member services (e.g. viewing content of a specific artist), and (2) buy merchandise related to the artist. There are two kinds of membership services: (1) member management services, which are about joining the portal, canceling membership, changing personal information and more, and (2) content services, which consist of a database of freely available content such as photographs taken by and from the artist, pieces of music, etc. These services are themselves outsourced. The portal only integrates them and adds the branding of the spe-

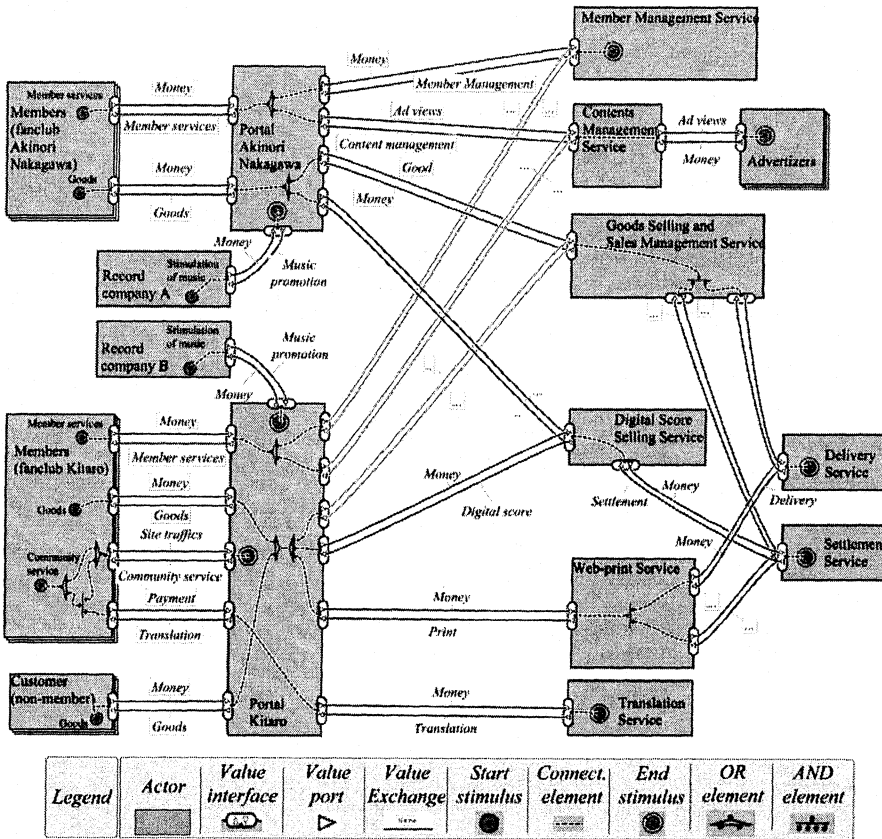


Figure 1. Value Model of the case study.

cific artist. The portal pays for member management services and obtains this money from members. The content provider obtains money from the portal and is financed by selling ads surrounding the content. For the fan, content is free.

A member can also buy goods (merchandise of an artist). For doing so, a service is needed that handles the sales of goods. We distinguish two types; merchandise goods and digital scores. The sale of both types of goods is outsourced to external parties by the portal. Selling goods requires payment (settlement) services (e.g. a credit card service). Consequently the goods selling and scores selling services require a payment service. Additionally, 'goods selling' requires delivery of the physical goods. As can be seen, the goods selling service uses a delivery service for doing so.

The Kitaro portal looks like the Akinori Nakagawa portal but has some additional services. First there is a web-print service. Members can use this service

to print various materials such as photographs. Also, the Kitaro portal has a community service (basically a discussion list). This service comes in two flavors. First, there is a discussion service for Japanese users in the Japanese language. The second service adds a paid translation service. For translation, an additional function provider is used.

The e^3 -value model illustrates that even seemingly simple services can involve a network of businesses that operate according to complex business models. The profitability of these business models for all cooperating businesses can be estimated if we assign *valuation functions* (basically a pricing formula) to all the ports exchanging money, and if we estimate the number of customer needs per timeframe, we can generate out of the model profitability sheets on a per actor basis. These sheets can be used to assess the potential for profit generation for each actor involved. It is a requirement that all actors can make a profit; otherwise the *model* is considered not sustainable. Due to lack of space, we do not present the sheets for the case at hand; the interested reader is referred to Gordijn and Akkermans, 2003 for the construction of such sheets.

5. Coordination modeling

Coordination modeling design decisions

A dependency path in an e^3 -value model shows which value exchanges are triggered by the occurrence of a consumer need, but it does not show how these exchanges are coordinated. Coordination is the interaction between a number of actors needed to produce a result. In the design of a coordination processes, the following choices have to be made. These choices are all relative to the design decisions made in the value model.

- Which information is exchanged between business partners, and in which order?
- What are the trust relations between the actors?
- Are additional actors needed to resolve trust issues (e.g., trusted third parties?)
- Who is responsible for the coordination activities at each business partner?

In this paper, we ignore trust relations (Wieringa and Gordijn, 2005) and focus on the coordination required to implement the value model under the assumption that all partners trust each other.

Exchange	Description
Members - Portal Kitaro	<p>Low trust. The coordination process must ensure that for each transaction, payment is guaranteed:</p> <ul style="list-style-type: none"> ■ Goods transfer from portal provider to members: goods are delivered via the delivery service on behalf of the portal provider after payment is confirmed by the settlement service. ■ Money transfer from members to portal provider: members have to use the settlement service for each order.
Portal Kitaro - Web-print service	<p>High trust. There is a long-term custom contract between the portal provider and the web-print service provider.</p> <ul style="list-style-type: none"> ■ Print ordering by portal provider: portal provider is expected to order prints one by one, for each client order. ■ Money transfer from portal provider to web-print service provider: The web-print service provider sends invoices at his discretion, which is paid by the portal provider's financial department.
Web-print service - settlement service	<p>High trust. Business is conducted according to the standard business procedure defined by the settlement service provider.</p> <ul style="list-style-type: none"> ■ Delivery of settlement service: the settlement service is provided to the members on behalf of the web-print service. ■ Money transfer from web-print service provider to settlement service provider: The settlement service sends a monthly subscription invoice.

Table 1. Value exchanges in the dependency path from Members via the Kitaro Portal and Web-print service to the settlement service. Exchanges refer to actors in Fig. 1.

Case study coordination processes

We take as an example the dependency path for goods that starts at the members of the Kitaro fanclub, goes through the Kitaro portal and the Web-print service and ends in the delivery and settlement service. The value exchanges that make up this dependency path are listed in Table 1.

A coordination process shows us how to deliver goods or services to one specific customer. However, some of the value exchanges to be implemented by the coordination process do not have to be carried out for each specific customer-driven instance of the process. This is the case for the entire exchange between the web-print service provider and the settlement service provider (both directions) and for the money transfer from the portal provider to

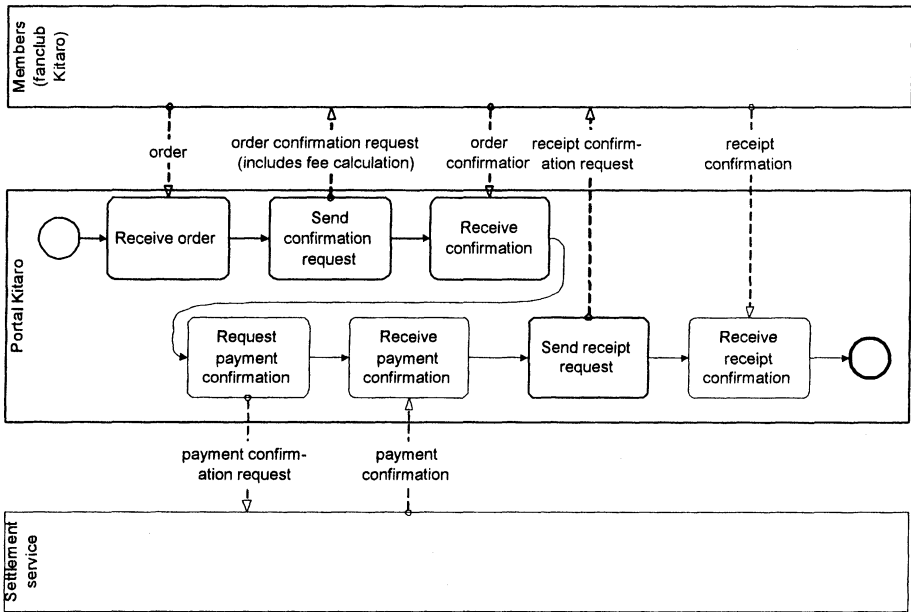


Figure 2. Coordination process of fanclub member and portal. See Fig. 3 for legend.

the web-print provider: they are aggregated in time-triggered processes as indicated in Table 1.

We assume that (secondary) business processes are already in place for sending invoices (settlement service provider and web-print service) and receiving and paying them (web-print service provider and portal provider) mentioned in Table 1. Therefore, they do not have to be taken into account in the design of the coordination processes for realizing the dependency path that we focus on.

Fig. 2 shows the coordination process (in BPMN notation, see White, 2004) that is being used in the case study between a member of the Kitaro fanclub and the Kitaro portal. The fanclub member starts by submitting an order. The portal receives this order and answers with a request for an order confirmation. The fanclub member confirms his order and receives a request to confirm receipt of the goods. The actual receipt of the goods is not in the figure, as this figure only contains information exchange, not the exchange of physical goods. After receipt of the goods, the fanclub member sends a confirmation. Because of space restrictions, we have chosen to only depict information exchange for a successful execution of the process.

The portal in turn coordinates its activities with the web-print service. This coordination process is depicted in Fig. 3.

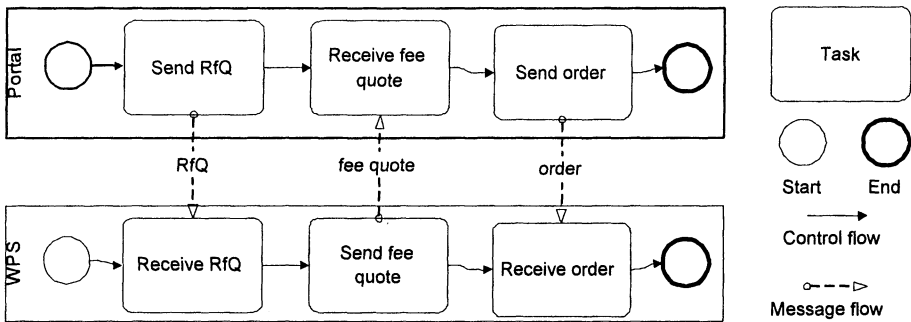


Figure 3. Coordination process of portal and web-print service (WPS).

IT support for coordination processes

The collaboration processes described above consist only of information exchange activities of business partners. IT support for coordination processes therefore only involves IT support for these information exchange activities. The following design choices have to be made:

- What technology to use (e.g., HTML forms, web services)?
- Synchronous or asynchronous information exchange?
- What is the format of the message data exchanged?

In our case study, information exchange with the fanclub members is based on HTML forms, as fanclub members (who are human end consumers) want to interact with the portal directly. The portal software designers have complete freedom in designing these forms.

Information exchange with the settlement service is most probably governed by the standard procedures published by the settlement service provider. In this case study, all relevant member data for payment is already collected in the first information exchange in the coordination process. This data is forwarded to the settlement service. Another possibility is that members are redirected to a website of the settlement service where they have to enter payment data.

We assume that information exchange between the portal provider and the web-print service provider is within the design charter of our case study and that there is an interest in using web service technology for this. In its basic form, web service technology is based on an asymmetric paradigm: a web service client calls a passive web service. The process depicted in Fig. 3, however, is symmetric. Therefore, choices have to be made as to who is the client and who is the server. There are two options: (1) The web-print service provider is always the server and provides a web service with two operations: one to receive a quote and one to submit an order. (2) A web service with one operation

is defined for each message flow arrow in Fig. 3; the web service is offered by the partner at the receiving end of the message flow arrow and called by the partner at the sending end of the message flow arrow. The number of operations in each option is influenced by choices with respect to whether operations are called synchronously or asynchronously: each asynchronously defined operation results in one additional operation, namely the callback operation. For this case study, we proceed with the first option, which results in the following web services (defined using WSDL; details left out):

```

<wsdl:definitions>
  <message name="orderDataMsg">
    <part name="..." type="..."/>
    ...
  </message>
  <message name="feeQuoteMsg">
    ...
  </message>
  <portType name="portalPT">
    <operation name="submitRfQ">
      <input message="orderDataMsg"/>
      <output message="feeQuoteMsg"/>
    </operation>
    <operation name="submitOrder">
      <input message="orderDataMsg"/>
    </operation>
  </portType>
</wsdl:definitions>

```

We use a BPEL4WS abstract business process to specify the choreography associated with these operations as follows:

```

<process name="portal2WPScoordination" abstractprocess="yes">
  <partnerlinks>
    <partnerlink name="portal2WPS" partnerLinkType="..."
      myRole="WPS" partnerRole="portal"/>
  </partnerlinks>
  <sequence>
    <receive partnerLink="portal2WPS" portType="portalPT" operation="submitRfQ"/>
    <reply partnerLink="portal2WPS" portType="portalPT" operation="submitRfQ"/>
    <receive partnerLink="portal2WPS" portType="portalPT" operation="submitOrder"/>
  </sequence>
</process>

```

As this is an abstract business process, the internal workflow executed to determine what to reply (second item in the sequence) to the received message (first item in the sequence) is left unspecified.

The above WSDL and BPEL fragments are sufficient to represent IT support for the coordination process depicted in Fig. 3. These fragments can be published by the web-print service provider and downloaded by the portal provider to find out how to conduct business. It is also possible to specify the process that the portal provider executes:

```

<process name="portal2WPScoordination" abstractprocess="yes">
  <partnerlinks>
    <partnerlink name="portal2WPS" partnerLinkType="..."
      myRole="portal" partnerRole="WPS"/>
  </partnerlinks>
  <sequence>
    <invoke partnerLink="portal2WPS" portType="WPSport" operation="submitRfQ"/>
    <invoke partnerLink="portal2WPS" portType="WPSport" operation="submitOrder"/>
  </sequence>
</process>

```

This fragment would be put in a separate BPEL4WS document; the relation with the BPEL4WS process description of the web-print service providers is only implied by the shared WSDL definition. This is a consequence of a fundamental design decision in BPEL4WS: BPEL4WS can only represent executable business processes (i.e., internal workflows, which will be treated in the next section) and abstract business processes, but not coordination processes (see Section 3 for the distinction between these three kinds of processes). The business process specification part of ebXML, as well as WSCI, do allow specification of coordination processes.

As we can learn from this example, a complex business collaboration as described by the dependency path that we have focused on in our case results in a number of relatively small web service choreography specifications such as the BPEL4WS fragments presented above. At least BPEL4WS does not allow a full specification of the entire coordination process associated with the dependency path (but other choreography standards may allow this). No notation that we are aware of would allow coordination actions that do not result in message exchange, such as physical delivery of goods.

6. Workflow modeling

Workflow design decisions

The coordination processes presented in the previous section only present message exchange between business partners. Workflow modeling focuses on the internal processes of each partner and determines how each partner brings about the exchanges specified in the coordination processes. Guidelines for design choices in workflow design can be found in organization theory (Daft, 1998), process design (Ould, 1995), and operations management (Slack et al., 1998). We do not summarize all of this but point out one of them:

- Are orders satisfied by taking the items ordered from stock or by manufacturing them at the moment the order arrives?

This question concerns the so-called customer order decoupling point (CODP), which is the place in the workflow beyond which the workflow is no longer for a specific client order.

Case study workflow processes

In our case study, deciding where the CODP lies comes down to deciding who keeps printed material on stock. There are three possibilities: (1) The portal orders printed material in batches of say 1000 copies and keeps them on stock (CODP is at the portal). (2) The web-print service prints the material in batches and keeps them on stock. The portal orders printed material per copy (one for each client request) and does not keep them on stock (CODP is at

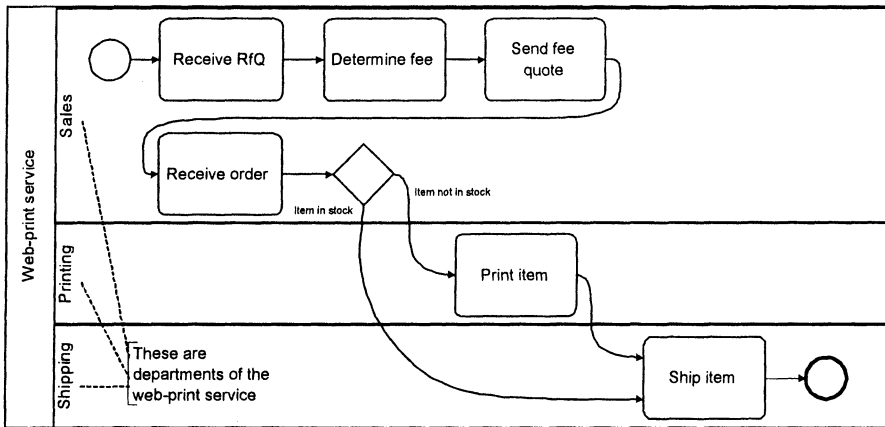


Figure 4. Web-print service internal process.

the web-print service). (3) Neither the portal nor the web-print service keeps printed material on stock. Instead, copies are printed in batches of size 1, with one batch per client order (no CODP).

The best alternative in this domain is usually determined by the printing technology used. E.g., it is prohibitively costly to run very small full-color offset printing jobs. For larger jobs, printing price per copy is low. With color laser printing, however, price per copy is constant, even for batches of size 1. The price per copy is, however, higher than for offset printing.

For commercial reasons, we cannot disclose the choice made in this case study, but we assume that the printer has chosen option (2). This gives us the workflow shown in Fig. 4. The workflow for the portal is trivial: it just relays messages between the fanclub member and the web-print service. This relaying is already fully captured in the middle part of Fig. 2, and is also presented in Yamamoto et al., 2004.

IT support for workflow processes

A workflow describes a process that is carried out by actors who work in one or more departments of one business. Although nowadays most workflow steps will be at least supported by information systems if not completely executed by it, one still has to consider exactly what steps need support from information technology and what this support entails. So, a workflow should not be confused with a specification of the behavior of one or more information systems. The main IT support design decisions are:

- Which workflow steps need support from information systems?
- What functions do these information systems need to offer?

- Distribution decisions, e.g. central IT facilities or facilities per location

These questions are fairly standard information systems design questions for which many design methods are available.

There are two different roles for a web service choreography standard such as BPEL4WS. First, there is an implementation relation between collaboration processes (which may be described in BPEL4WS) and internal workflows. So, a BPEL4WS description can be considered a partial specification of the workflows that have to be designed. Second, in a number of cases, internal workflows have to be formally specified and supported by workflow management systems. In these cases, the workflows can be described as BPEL4WS executable processes and executed by the BPEL4WS execution engines that are currently emerging in the market.

7. Conclusion

In this paper, we have systematically identified all design decisions that need to be made when designing multi-party collaborations. This revealed a clear distinction between value modeling, coordination modeling and modeling internal workflows. IT support is different for each of the latter two, as is the role of web service technology and choreography standards. For each type of modeling, we have shown examples (using our case study) of modeling techniques such as *e³-value* and BPMN. These modeling techniques are relatively lightweight while still providing enough insight to support design decisions. Moreover, these techniques are simple enough to be understood by non-technical stakeholders, which is important as design decisions made by software engineers influence design decisions that have to be made by business stakeholders and the other way around.

Many papers on web service technology that are currently being published focus on automatic service discovery and composition. We have not used any of this. One could argue that our case study is too conservative and should have used e.g. web service discovery to dynamically find a suitable settlement service and web service composition to dynamically create the collaborations instead of statically at design time. However, it is currently much too early to employ these technologies in real-world case studies. More importantly, we think that it is essential to first fully understand the static case before moving on to dynamic composition of business collaborations. As an aside, in a case like this which involves marketing mass-produced goods, a need remains for static, long-running relations for reasons of logistic and manufacturing efficiency.

Further work includes a validation of our approach on more case studies, and an analysis of the correctness relationship between a coordination process and a value model (when does a coordination model correctly implements a

value model?) and between an internal and an abstract business process (when does an internal business process correctly implement an abstract process?)

References

- Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). *Web Services: Concepts, Architectures and Applications*. Springer-Verlag.
- Choi, S.-Y., Stahl, D. O., and Whinston, A. B. (1997). *The Economics of Doing Business in the Electronic Marketplace*. MACMillan Technical Publishing, Indianapolis, IN.
- Daft, R. L. (1998). *Organization Theory and Design*. Thomson Publishing, sixth edition.
- Gordijn, J. and Akkermans, J. (2003). Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134.
- Gordijn, J. and Akkermans, J. M. (2001). Designing and evaluating e-Business models. *IEEE Intelligent Systems - Intelligent e-Business*, 16(4):11–17.
- Horngren, C. T. and Foster, G. (1987). *Cost Accounting: A Managerial Emphasis, sixth edition*. Prentice-Hall, Englewood Cliffs, NJ.
- Kotler, P. (1988). *Marketing Management: Analysis, Planning, Implementation and Control, 6th edition*. Prentice Hall, Englewood Cliffs, NJ.
- Ould, M. A. (1995). *Business Processes—Modelling and Analysis for Re-engineering and Improvement*. Wiley.
- Slack, N., Chambers, S., Harland, C., Harrison, A., and Johnston, R. (1998). *Operations Management*. Pitman Publishing, second edition. ISBN: 0-273-62688-4.
- van der Aalst, W. M. P. and Weske, M. (2001). The P2P approach to interorganizational workflows. In Dittrich, K. R., Geppert, A., and Norrie, M. C., editors, *Advanced Information Systems Engineering: 13th International Conference, CAiSE 2001*, volume 2068 of *Lecture Notes in Computer Science*, pages 140–156.
- van Eck, P., Gordijn, J., and Wieringa, R. (2004). Value-based design of collaboration processes for e-commerce. In Yuan, S.-T. and Liu, J., editors, *Proceedings 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, EEE'04*, pages 349–358. IEEE Press.
- White, S. (2004). Business process modeling notation (BPMN). <http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf>, visited 20041123.
- Wieringa, R. J. and Gordijn, J. (2005). Value-oriented design of service coordination processes: Correctness and trust. In *Accepted for Proceedings of the ACM Symposium on Applied Computing, Organizational Engineering Track*. <http://www.cs.vu.nl/~gordijn/sac05-submitted.pdf>.
- Yamamoto, R., Ohashi, K., Yamamoto, K., Inomata, J., and Matsuda, T. (2004). Lessons learned from requirements analysis for implementing integrated services. In *Proceedings of the International Workshop on Service-Oriented Requirements Engineering – SORE 2004*.