INTERACTION-ORIENTED COLLABORATIONS

Giorgio Bruno
Dip. Automatica e Informatica, Politecnico di Torino, Torino, Italy

Keywords: Collaborative business processes, binary collaborations, choreographies, abstract orchestration models.

Abstract: This paper addresses binary collaborations and choreographies, based on web services technology. The nature of the problem leads to two complementary approaches: one focuses on activities, and the other on interactions. This paper follows the interaction-oriented approach and proposes a modeling notation, called Interaction-Oriented Nets (IONs), which allows binary collaborations, choreographies and abstract orchestration models (i.e. abstract business processes made up of communication activities) to be homogeneously represented.

1 INTRODUCTION

In the domain of collaborative business processes, collaboration is the term denoting the situation in which two or more participants (i.e. their business processes) exchange messages so as to achieve a common goal. Before carrying out the actual collaboration, the participants have to agree on an a priori model, called collaboration model or choreography, specifying how the interactions have to take place. The models defining the interactions between two participants are called binary collaboration models.

Collaboration models are addressed from two perspectives: one is focusing on the observable activities of the participants, and the other on the interactions. These approaches are not conflicting: in fact, a one-way interaction subsumes two activities, a sending activity in one participant, and a receiving activity in the other.

The activity-oriented collaboration models are also called inter-organizational (or cross-organizational) workflows. They are global models defining the observable activities of the participants as well as their ordering constraints.

The major issues addressed in this area of research are the modeling language for the global model, and the mapping from the global model to the local (i.e. pertaining to a given participant) business processes, also called local workflows or abstract processes.

This approach is appealing as the global model is similar to a business process, and for this reason the modeling language is the same. As to the mapping, several techniques have been proposed. The Public-To-Private technique (van der Aalst and Weske, 2001) follows a top-down approach consisting of three steps: the participants agree on a global Petri net, then the public model is partitioned into public parts, one per participant, and finally each participant refines its public part into a private workflow. The private workflows are guaranteed to conform to the global model, as the private refinement is based on a specific notion of inheritance. The technique (Zhao, Liu and Yang, 2005) based on relative workflows follows a bottom-up approach, since each participant can expose different public parts, called relative workflows, to the other participants. A mixed approach is the one based on workflow views (Orlowska and Schulz, 2004). Workflow views are the public parts of a cross-organizational model, called a coalition workflow: private workflows interact with workflow views and workflow views interact with each other, either directly or through a mediator.

The approach focusing on interactions is more abstract. In fact, interactions, which are of two types, i.e. one-way interactions and two-way ones, are abstract entities, as in reality they are carried out by sending activities and receiving ones. Interaction-oriented binary collaboration models were first proposed in the e-business domain by the RosettaNet consortium (Damodaran, 2004), which adopted the UMM modeling notation (UMM, 2003) and the ebXML BPSS textual description (BPSS, 2001). The content of application messages in a variety of

Bruno G. (2007).
INTERACTION-ORIENTED COLLABORATIONS.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 202-207
DOI: 10.5220/0002376502020207

business cases, quality of service requirements and transactional features are the key issues of this initiative. A recent proposal is the Web Services Choreography Description Language (WS-CDL, 2005): it is an XML-based language addressing peer-to-peer collaborations.

This paper follows the interaction-oriented approach and proposes a modeling notation, called Interaction-Oriented Nets (IONs), which addresses binary collaborations and choreographies homogeneously. On the contrary, UMM does not deal specifically with binary collaborations, and BPSS handles binary collaborations only. The same notation can also be used to represent abstract orchestration models (AOMs), an AOM being a local (i.e. pertaining to a given participant) business process restricted to communication activities and control-flow ones.

This paper is organized as follows. Section 2 presents Interaction-Oriented Nets and shows how binary collaborations can be represented. Section 3 illustrates light choreographies, which are meant to complement binary collaborations by capturing the global constraints. Section 4 addresses abstract orchestration models and section 5 presents the conclusion and the future work.

2 BINARY COLLABORATIONS

Two examples of binary collaborations are shown in Fig. 1. They are based on a special kind of Petri nets, called Interaction-Oriented Nets (IONs), which can be informally described, as follows.

There are two types of transitions in an ION, i.e. ordinary transitions and interaction-oriented ones. The latter represent either one-way interactions, such as order, or two-way interactions, such as rfq/quote (rfq stands for request for quote). In two-way interactions, a slash (/) separates the request message from the response one. As in RosettaNet, application messages, such as rfq, quote and order, are acknowledged by means of signal messages; usually a signal message acknowledges that an application message has been received and has been syntactically validated. Signal messages do not need to be shown in collaboration models. The types of the messages are defined in an XML schema file associated with the collaboration model.

A binary collaboration takes place between two participants, denoted by two conventional roles, i.e. requester and provider. An interaction also takes place between two participants, denoted by two conventional roles, i.e. initiator and responder. The collaboration requester coincides with the initiator of the first interaction. If an interaction is initiated by the collaboration provider, the request message is underlined (this case does not appear in Fig. 1).

There are similarities with the work done by the Language/Action community: in fact, the notions of business act, action pair and business transaction (Lind and Goldkuhl, 2003) are similar to the notions of one-way interaction, two-way interaction and binary collaboration, respectively, although the former appear to be more abstract than the latter.

Binary collaborations are meant to be used in abstract orchestration models and in choreographies; then, more specific roles (e.g. buyer and seller), instead of the conventional ones, will be adopted, as will be shown in the next sections.

An ION has a source place (initially marked) and a sink place: moreover, it is case based, as it describes the life cycle of a single collaboration. To make the model more compact, places are absorbed in links, unless they have two or more incoming links or two or more outgoing links.

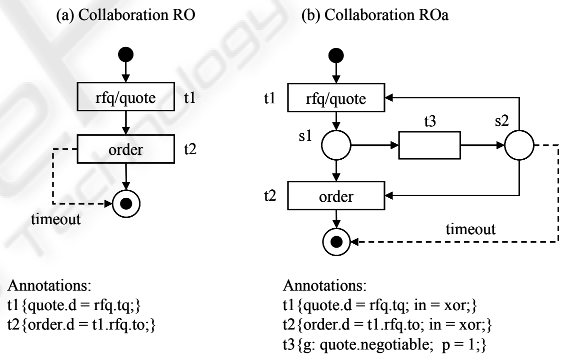

Figure 1: Binary collaborations RO (a) and ROa (b).

In collaboration RO, shown in Fig. 1a, the requester sends a request for quote, rfq, to the provider, and then waits for a quote. If the quote is accepted, the requester will send an order to the provider.

As in UMM, interactions have attributes, the most important of which are the deadlines ("d") of the messages involved. If a message is not sent/received before its deadline, the corresponding interaction will fail, unless a timeout path is provided. If an interaction fails, the whole collaboration gets blocked; the parties, however, can agree on a protocol for unblocking collaborations.

The scripts used in the annotations, i.e. "quote.d = rfq.tq" and "order.d = t1.rfq.to", mean that the deadlines of messages quote and order will be set to the values read from attributes tq and to of the rfq.

Past messages can be used in the scripts as global variables.

Timeout links introduce timeout paths, i.e. those paths to be followed when interactions fail. In Fig. 1, timeout links, i.e. the dashed links, lead to the sink state, and hence conclude the collaboration. In fact, the order is optional, and its absence is not a reason for making the collaboration fail.

Collaboration ROa, shown in Fig. 1b, presents a more complex protocol, in which a quote is assumed to include a flag indicating whether it is negotiable or not. After receiving a quote, the requester can send a purchase order or, if the quote is negotiable, it can send a revised request for quote. If the quote is negotiable, two alternative paths are possible, one consisting of interaction order and the other starting with interaction rfq/quote.

The choice of a path can be either data-driven or event-driven. In the first case, the choice must be based on public information, visible to both parties: such public information is given by the contents of past messages, i.e. the messages exchanged by the parties before the choice is made. In the second case, the choice depends on the arrival of future messages. When collaboration ROa is in state s1, a data-driven choice takes place, depending on the contents of the last quote. Transitions can have guards ("g") and priorities ("p"), whose default value is 0. If the quote is negotiable, transition t3 fires, otherwise transition order is enabled.

State s2 determines an event-driven choice. An event-driven choice (also called deferred choice) occurs when a place is followed by two or more interactions in the same direction. While the sender is free to select which message to send, the receiver is assumed to be able to receive whichever message will be sent, therefore the term "deferred choice" expresses the viewpoint of the receiver.

In order to fire, a transition may require all its input places to be marked (i.e. non-empty) or just one; in the first case its parameter "in" is set to "and" (the default value), and in the second case it is set to "xor". Transitions rfq/quote and order have both a "xor" input behavior, as they have two input places, which are never jointly marked.

3 LIGHT CHOREOGRAPHIES

Choreography usually denotes an a priori global model meant to capture all the interactions taking place for a given purpose among a number of participants. As such it is a much debated notion. It is often associated with the idea of a leading organization having the authority of imposing the required behavior on the participating organizations.

Three points of weakness have been pointed out (Zhao, Liu and Yang, 2005): a leading organization may not exist, a participant may be willing to select its own partners, and participants are exposed to unnecessary information.

This paper proposes a weaker notion of choreography, called light choreography, which is meant to complement binary collaborations.

Binary collaborations express necessary precedence constraints on the interactions taking place between pairs of participants; however additional constraints might be needed. Light choreographies are meant to make such additional constraints explicit and public to all the parties involved, so that each party can work out an appropriate abstract orchestration model. A case study requiring a light choreography is as follows.

The case study is a simplified supply chain involving a buyer, a distributor and a supplier, which are denoted by their initials, b, d and s.

The buyer sends a purchase order (bo = buyer order) for certain goods to a distributor which can fulfil the order in two ways: a) with one delivery (dd = distributor delivery) coming from an internal warehouse; b) with two deliveries, a distributor delivery (dd) and an external delivery (sd = supplier delivery) coming from an external supplier.

After receiving the buyer order, in case b, the distributor selects a supplier with a reverse auction similar to collaboration RO shown in Fig. 1a, and then informs the buyer of the supplier selected with a notification message (dn = distributor notification). Message dn includes attribute deliveryN, whose value can be 1 or 2; in case the value of deliveryN is 2, message dn also includes a reference to the supplier involved. Then the buyer sends some delivery information (bi = buyer information) to the supplier, if it is the case.

After the deliveries have been performed, i.e. messages dd and sd (if it is the case) have been received by the buyer, the buyer makes a payment in favor of the distributor and sends it a payment notification (bp = buyer payment).

After delivering the goods to the buyer, the supplier sends a payment request (sr = supplier request) to the distributor; after making the payment in favor of the supplier, the distributor sends a payment notification (dp = distributor payment) to the supplier.

For the sake of simplicity, deadlines and exceptions are ignored.

A business case with two deliveries is informally presented in the sequence diagram shown in Fig. 2a.

Three binary collaborations are implied, i.e. BD, DS and BS, as shown in Fig. 2b; they are identified by the initials of the participants in capital letters, the first letter denoting the requester. The interactions, such as dd and sr, that are initiated by the collaboration provider are shown underlined.

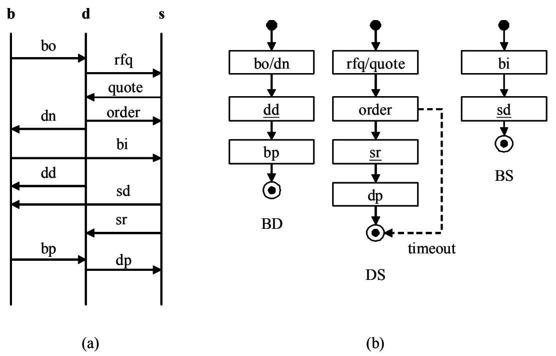

Figure 2: Sequence diagram (a) and binary collaborations (b) for the case study.

This case study presents the routing pattern called request with referral (Barros, Dumas and ter Hofstede, 2005) as the buyer sends message bi to the supplier that is indicated in message dn received from the distributor.

The (light) choreography for the case study is shown in Fig. 3; it is based on the binary collaborations shown in Fig. 2b. A choreography model is an ION with two additional annotations, one defining the participants in terms of their roles, and the other declaring the binary collaborations needed.

This choreography involves three participants denoted by their initials; if a message is meant to contain a reference to a participant, this message is shown next to that participant, as in the case of message bd.dn, which is shown, within parentheses, next to participant s.

The collaborations section lists the binary collaborations needed along with the participants involved and gives them suitable identifiers; as an example, bd identifies collaboration BD taking place between the buyer and the distributor. The interactions in the choreography are preceded by the appropriate collaboration identifiers, e.g. bd.dd and bd.bo/dn.

The choreography shown in Fig. 3 features two forks and one join. Fork f1 is needed because the two deliveries may take place in any order. Fork f2 enables both join j1 and the request for payment (sr) from the supplier. Join j1 is needed because the payment in favor of the distributor is made after the deliveries.

Place s1 determines a data-driven choice: if attribute deliveryN of message bd.dn is equal to 1, transition t10 fires and moves the token form s1 to s2. In this case the buyer waits for the distributor delivery only.

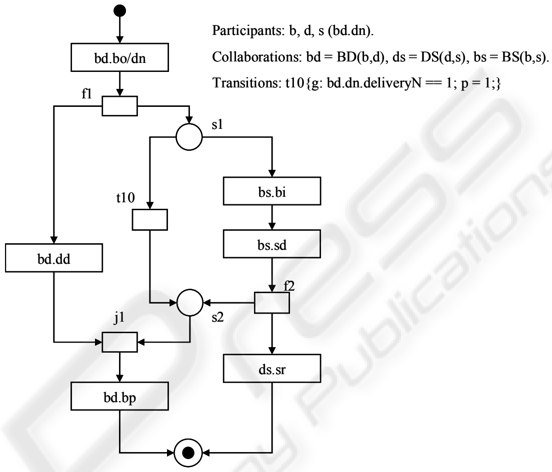

Figure 3: Light choreography for the case study.

The precedence between dd and bp in binary collaboration BD is necessary but not sufficient; in fact, the choreography shows that, when two deliveries are needed, bp has to be preceded by sd, as well. In this sense BD is a weak binary collaboration, while RO and ROa shown in Fig. 1 are strong binary collaborations.

The solution given to the case study is based on a number of considerations.

Firstly, choreographies do not replace binary collaborations. The main reason is to expose only the interactions needed for global coordination, while those related to the details of binary protocols are not revealed. In fact, the choreography shown in Fig. 3 does not include interactions ds.rfq/quote and ds.order, because they are of no interest for the buyer. Moreover, interaction dp is not included in the choreography, as it follows interaction sr on the basis of binary collaboration DS. Binary collaborations are needed as they drive the implementation; a previous paper (Bruno and La Rosa, 2006) has shown how to automatically generate WSDL documents and BPEL processes from binary collaboration models.

Secondly, choreographies are not global models; they do not capture all the interactions taking place for a given purpose, but only those necessary for global coordination. A critical issue is the presence of several participants playing the same role: as a matter of fact, several suppliers are involved in the reverse auction conducted by the distributor. In such situations, multiple interactions such as those illustrated in the next section are likely to appear. While a global model is meant to represent all of them, in a choreography only one representative per role is needed.

4 ABSTRACT ORCHESTRATION MODELS

While collaboration models establish how the parties have to interact so as to achieve a common goal, it is up to each party to orchestrate (i.e. to combine) the collaborations it is involved in. An abstract orchestration model (AOM) fits that purpose. It provides the so-called behavioral interface (Barros, Dumas and Oaks, 2006) of a given participant, from which an actual orchestration process can be developed.

In this paper AOMs are meant to organize the interactions pertaining to a given participant in a proper control structure: hence they are still interaction-oriented nets (IONs) enriched with specific features, in particular multiple interactions and nested interactions. They are abstract models, for a number of details, as will be illustrated in this section, are left unspecified.

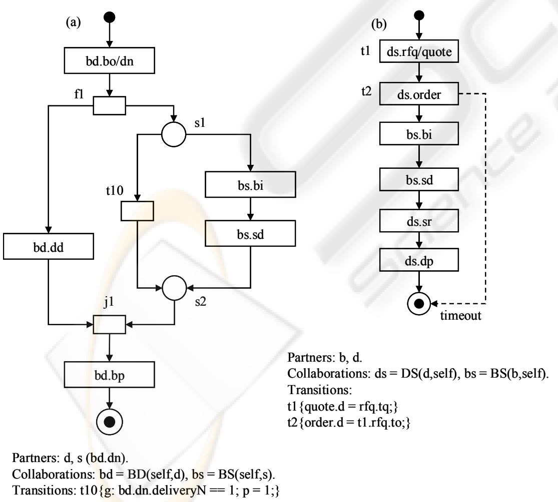

Figure 4: Abstract orchestration models of the buyer (a) and of the supplier (b).

In Fig. 4 the buyer AOM and the supplier one are presented, while the distributor AOM is shown in Fig. 5.

Specific annotations list the binary collaborations needed along with the other participants.

The buyer interacts with a distributor and a supplier; as the supplier is introduced by the distributor to the buyer with message dn, this message is shown in the partners section, next to the supplier.

AOMs are based on binary collaborations; as binary collaborations do not specify the specific roles involved, it is the AOM task to declare which binary collaborations it needs along with the role it plays (self) and the role of the other participant. The buyer is involved in two binary collaborations, bd and bs, playing the requester role in both of them.

The distributor is involved in several collaborations ds, each with a different supplier, as it is supposed to conduct a reverse auction with them. Notation “ds* = DS(self,s)*” defines ds* to be a collection of similar collaborations (ds indicates one of them).

The thick link from bd.do/dn to d1 in Fig. 5 is the nesting operator used in AOMs: its source is a two-way interaction and its destination is a component AOM. In fact, the distributor, after receiving a purchase order from the buyer and before replying with a dn message, is meant to operate as indicated in the nested AOM.

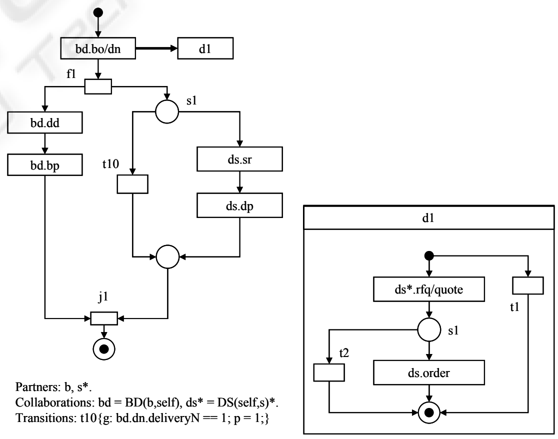

Figure 5: Abstract orchestration model of the distributor.

Transitions t1 and t2 in d1 are abstract, for there are no annotations associated with them. In fact, t1 is meant to fire if the distributor decides not to involve any supplier, and t2 is meant to fire if it decides not to accept any of the quotes received. In both cases, only the distributor delivery will take place.

Interaction ds*.rfq/quote is called a multiple interaction and indicates a multiplicity of interactions rfq/quote each taking place between the distributor and a different supplier. The exact way (e.g. in parallel or in sequence), in which such interactions will be performed, is an implementation detail and the AOM is not concerned with such aspects. After those interactions have been completed, the buyer is meant to select the best quote and, if there is any, it will send an order to the corresponding supplier.

While it is possible (Barros, Dumas and Oaks, 2006) to automatically obtain an orchestration model for each participant from a choreography model, this paper, nevertheless, considers binary collaborations to be essential, for two reasons.

Firstly, they enforce the protocol at the lower level. In fact, binary collaborations can give rise to run-time entities which maintain the state of the actual collaborations. This is particularly useful when multiple collaborations are involved. In fact, the orchestration process of the distributor could mistakenly send the order to a supplier that did not provide any quote. Therefore the run-time checks performed by a run-time collaboration entity can prevent a process from sending or receiving a message in wrong order (or not complying with timing constraints).

Secondly, run-time collaboration entities can implement the proper interaction protocol (based on timeouts and retrials) thus relieving the orchestration processes of this burden.

5 CONCLUSION AND FUTURE WORK

This paper has presented a modeling notation, called Interaction-Oriented Nets (IONs), which addresses binary collaborations, light choreographies and abstract orchestration models (AOMs) homogeneously.

Current work proceeds in several directions. While binary collaborations are well understood, in choreographies and in AOMs there are still several issues to be settled.

Choreographies are subject to well-formedness rules, which are related to their particular use. As an example, it does not make sense to say that an interaction between x and y precedes an interaction between w and z (with x, y, w and z indicating different participants), as there is no way to enforce that precedence, without the participants being coordinated by a central entity.

Choreographies and binary collaborations have joint operational purposes. Each participant can obtain an AOM from them, as shown in Fig. 4 and in Fig. 5, and then it can complete it with internal activities. AOMs have to be validated against the choreographies and the binary collaborations they are based on. Moreover, a first-cut AOM can be automatically obtained from a choreography model, and then manually enriched. As an example, the buyer AOM shown in Fig. 4 can be easily obtained from the choreography presented in Fig. 3 by means of suitable reduction rules.

REFERENCES

Barros, A., Dumas, M., ter Hofstede, A.H.M., 2005. Service interaction patterns. In *Lecture Notes in Computer Science*, 3649, 302-318. Springer. Berlin.

Barros, A., Dumas, M., Oaks, P., 2006. Standards for web service choreography and orchestration: status and perspectives. In *Lecture Notes in Computer Science*, 3812, 61-74.

Bruno, G., La Rosa, M., 2006. From collaboration models to BPEL processes through service models. In *Lecture Notes in Computer Science*, 3812, Springer. Berlin.

Damodaran, S., 2004. B2B Integration over the Internet with XML – RosettaNet successes and challenges. Retrieved February 20, 2007, from http://doi.acm.org/10.1145/1013367.1013398.

BPSS, 2001. Business Process Specification Schema, Version 1.01. Retrieved February 20, 2007, from http://www.ebxml.org/specs/ebBPSS.pdf.

Lind, M., Goldkuhl, G., 2003. The constituents of business interaction - generic layered patterns. *Data & Knowledge Engineering*, 47, 327–348.

Orlowska, M.E., Schulz, K.A., 2004. Facilitating cross-organisational workflows with a workflow view approach. *Data & Knowledge Engineering*, 51, 109-147.

UMM, 2003. UN/CEFACT Modeling Methodology (UMM) User Guide. Retrieved Retrieved February 20, 2007, from http://www.unece.org/cefact/umm/UMM_userguide_V20030922.pdf.

van der Aalst, W.M.P., Weske, M., 2001. The P2P approach to Interorganizational Workflows. In *Lecture Notes in Computer Science*, 2068, 140-156. Springer. Berlin.

WS-CDL, 2005. Web Services Choreography Description Language, Version 1.0. Retrieved February 20, 2007, from http://www.w3.org/TR/ws-cdl-10/.

Zhao X., Liu C., Yang Y., 2005. An organisational perspective on collaborative business processes. In *Lecture Notes in Computer Science*, 3649, 17-31. Springer. Berlin.