

Process Choreography Modelling

Alistair Barros

Abstract A dedicated B2B process perspective has been the subject of significant development in the BPM field over recent years, seen notably through the BPMN 2.0 specification, which includes process choreography as one of its largest extensions. In the wider context of information systems analysis and design methodologies, the B2B perspective represents a crucial context through which requirements and iterative design of solutions are developed, as seen through numerous methods and techniques developed in the field across the last 40 years. To date, an understanding of how contemporary choreography proposals, typified by BPMN 2.0 process choreography, measure up to classical IS methodology capabilities, remains unaddressed. Consequently, a methodological understanding of how choreography could be used, from high-level analysis to detailed design, remains open. In this chapter, we address this gap. Based on an example taken from the supply chain management domain, we identify three important requirements for process choreography that make this wider perspective amenable to methodological systems analysis and design: *functional scoping* of different areas concerning a domain which can then be modelled and related to each other in isolation; *stepwise refinement* of choreography models, reminiscent of classical analysis techniques; and the introduction of *conversation semantics* expressing the intent of logically related message exchanges of choreographies. Accordingly, we propose extensions to choreography modelling and an improved analysis of requirements, such as breakdowns in negotiations that take place between collaborating partners, using concepts directly supported by, and illustrative extensions of BPMN 2.0 process choreography.

A. Barros (✉)

Queensland University of Technology, Y Block, QUT Gardens Point Campus, Brisbane, QLD 4000, Australia

e-mail: alistair.barros@qut.edu.au

1 Introduction

Choreography, as originally coined through Web services standardization efforts, is a particular aspect of business processes which relates to the way business partners coordinate their activities in a value-chain. The focus is not on full orchestrations of processes operating within these partners, but rather on the collaboration that takes place *between* partners. Collaboration in value-chains entails messages (document) exchanges in an orderly fashion: e.g. first a retailer sends a purchase order request to a supplier; next the supplier either confirms or rejects intention to investigate the order; then supplier proceeds to investigate stock for line-items and seeks outside suppliers if necessary; accordingly the supplier sends a confirmation or rejection back; during this period the retailer can send requests to vary the order, etc.

The need for modelling choreographies, over and above conventional business process modeling, has become increasingly important as businesses shift their operations into wider value-chains featuring many collaborating partners and dynamic outsourcing and insourcing of services (vom Brocke 2007). Such a setting can involve not tens, but hundreds, of message exchanges. Interactions between partners can go beyond simple request-response interactions into more complex multi-cast, contingent requests, competing receives, streaming and dynamic routing among different patterns (Barros et al. 2005). Moreover, message exchanges cluster around distinct scenarios, otherwise known as *conversations*, such as: creation of sales orders; assignment of carriers of shipments involving different sales orders; managing the “red tape” of crossing customs and quarantine; processing payment and investigating exceptions. Conversations, as such, entail a set of message exchanges that are correlated in different ways, e.g. (Barros et al. 2005) provides a list of patterns for correlating message exchanges into conversations (e.g. key-based, function-based).

By abstracting away from internal processing details of processes, choreography models bring message exchanges and their logical grouping as conversations into view. This allows partners to plan their business processes for inter-operation without introducing conflicts. An example of a conflict could arise if a retailer was allowed to send a variation on a purchase order immediately after sending the initial request – because a supplier may not be able to efficiently confirm availability of stock. Once conversational sequences in choreography models are agreed upon, they can be mapped to each partner’s orchestration models (Decker and Weske 2007).

In terms of developments in business process modeling, choreography languages, as introduced in recent years, are largely suitable at the detailed design, and, often, implementation focused, phase. This is because the details of message exchange and message correlation are seen as considerations of interoperability, which is relevant once implementation choices have been made (e.g. using Web services and orchestration through WS-BPEL).

The concern of collaborations, however, is also of interest during higher levels of process analysis where interactions between partners establish the *context* upon which requirements are analyzed. Typical lines of enquiry involve determining the

functional scope of the business domain being analyzed and the landscape of partners, their underlying business processes and the triggers that activate their execution, the business objectives advanced and the operational impediments that stand in the way, etc. This is the subject of the early stages of IS analysis and design in which informal, diagrammatic techniques are typically used to understand collaborations between partners, e.g. Structured Analysis and Design (Yourdon 1988).

The difference between classical techniques of analysis and contemporary techniques for choreography modeling – both of which concern process collaboration – is that former is informal, omitting detailed considerations of message exchange, and supporting business analysts to establish the broader organizational context through iterative and typically intensive “whiteboard” analysis.

This chapter provides insights into the way choreography modeling can be extended for the purposes of both high-level process analysis and detailed design. To this end, it first provides an insight into current state-of-the-art for choreography modeling, illustrating how message exchanges and conversations can be modeled through Let’s Dance and its standards “successor” – in choreography proposals of Business Process Modelling Notation (BPMN) 2.0. With this insight in place, it then discusses requirements for choreography languages that are pertinent for high-level process analysis. Three requirements for extending choreography modelling are proposed, namely: the way choreography models are scoped, detailed and inter-related for large domains; the way they are refined in a stepwise manner from the highest context level to the detailed implementation-specific level; and the way intent of message exchanges qualify message exchanges in order to improve analysis of models from a semantic point of view. To illustrate how these requirements can be met, specific extensions are illustrated using the Semantic Object Modelling (SOM) framework. The result is show how choreography modeling can support complex domains with many participants and processes, how it can be used across high-level analysis and detailed design tasks, and how improved semantic analysis of models is possible, e.g. breakdown in the negotiations intended by message exchanges can be automatically detected.

2 Choreography Modelling Developments Through Process Languages

A straightforward way of modeling choreographies is by connecting process models at points where messages are exchanged. In BPMN this is done through the collaboration diagrams, as illustrated in Fig. 1. For a detailed insight into BPMN, the reader is referred to Aagesen and Krogstie (2014), Kemsley (2014), and White et al. (2008).

Figure 1 shows a collaboration diagram where BPMN pools are expanded to reveal orchestration details per participant (for *Shipper*, *Retailer* etc.). Message

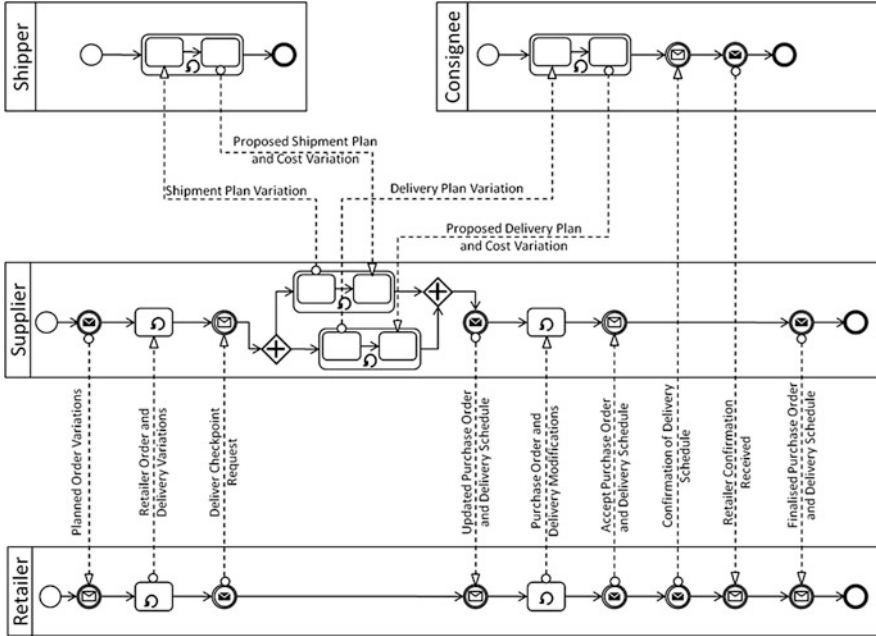


Fig. 1 Interconnecting process models

flows (dashed arrows) connect the elements in the different pools related to different participants and thus indicate message exchanges. For example, a *Planned Order Variations* message is sent by the *Supplier* to the *Retailer*; the corresponding send and receive have been modelled using regular BPMN messaging events. BPMN also lends itself to supporting a number of messages of the same type being sent. For example, a number of *Retailer Order and Delivery Variations* messages can be sent from the *Retailer* to the *Supplier*, indicated by respective multiple instances constructs (for brevity, the actual elements for sending/receiving inside the multiple instances construct have been omitted).

Taken as a whole, the scenario modeled in Fig. 1 entails shipment planning for the next supply replenishment variations: the *Supplier* confirms all previously accepted variations for delivery with the *Retailer*; the *Retailer* sends back a number of further possible variations; the *Supplier* requests to the *Shipper* and *Consignee* possible changes in delivery; accordingly, the *Retailer* interacts with the *Supplier* and *Consignee* for final confirmations.

It should be noted that in practice, inter-process connections would be made against process models which serve as interfaces, since these allow hiding of actual internal processes and provide flexibility for internal processes to change without “breaking” interconnections. A major problem with model interconnections for complex choreographies is that they are vulnerable to errors – interconnections may not be sequenced correctly, since the logic of message exchanges is considered from each partner at a time. This in turn leads to deadlocks. For example, consider

the role of *Retailer* in Fig. 1 and assume that here, by error, the order of *Confirmation Delivery Schedule* and *Retailer Confirmation received* (far right) were swapped. This would result in a deadlock since both, Retailer and Consignee would wait for the other to send a message. Deadlocks in general, however, are not that obvious and might be difficult to spot.

Accordingly, the need to model choreographies, independent of the perspective of individual partners – the so-called *global* perspective – was inspired through Web services standardization efforts. WS-CDL (Kavantzias et al. 2005), which has succeeded previous efforts, models messages exchanges as first-class constructs. WS-CDL is implementation-specific and, as it turned out, difficult to map into popular process execution languages like WS-BPEL. This has inspired efforts for developing *implementation independent* (conceptual) modeling languages, notably Let’s Dance (cf. Zaha et al. 2006). Figure 2 reformulates the above example of Fig. 1 to show how the message construct in Let’s Dance could be adapted to describe choreographies, as a precursor to BPMN choreography developments.

As shown in Fig. 2, a choreography activity represents the message exchange as an activity-like construct. The sender and receiver, directionality of message exchange, and the message type are expressed. Multiple instances, looping and sub-process from regular BPMN are adapted for choreography activities to model concurrent iterations and decomposition of message exchanges in choreography activities.

As can be seen, the logic of a conversation is relatively simple to follow. Process routing constructs are leveraged to model the sequencing of message exchanges – without any dependency on processes of the participants. Of course, the

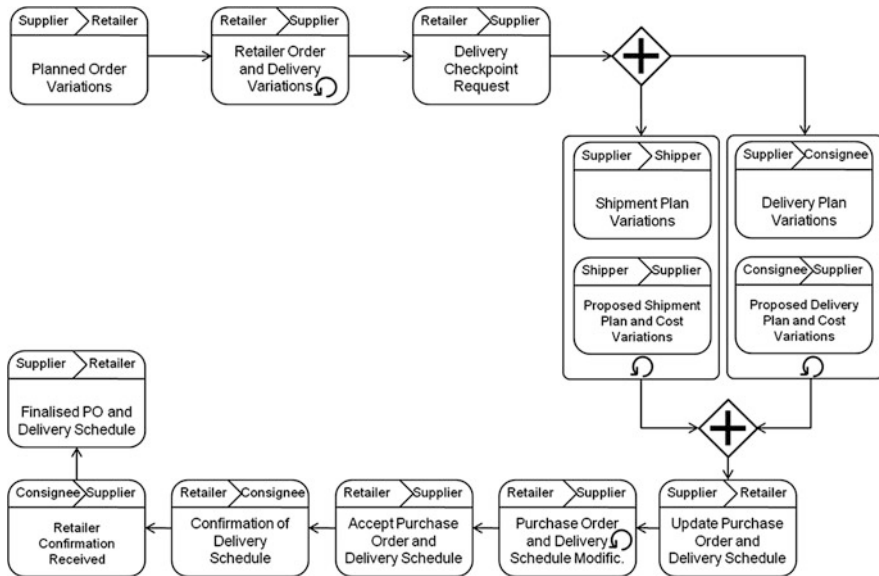


Fig. 2 Modelling of message exchanges as flow elements

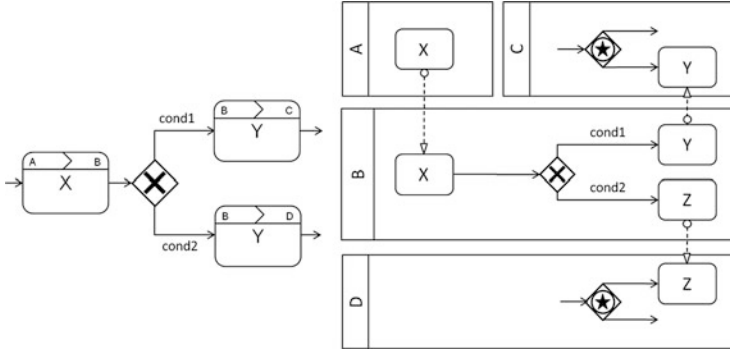


Fig. 3 Mapping an OR-split in a choreography model into partner process models

choreography model needs to be mapped to participant processes. A major problem in this regard is the *local enforceability* of the required sequencing. That is to say, the sequencing in the global choreography model should be reflected in the sequencing of message exchanges related within individual partner processes. An example of an unenforceable sequence would be if an exchange took place between a *Retailer* and a *Supplier* which was followed by an exchange between *Shipper* and *Consignee*. How does *Shipper* know when *Supplier* received the message from *Retailer*?

Figure 3 provides an insight into how a choreography model containing an exclusive OR-split would be mapped into local models.

The choreography fragment on the left hand side in Fig. 3 specifies that there is an exclusive decision after message exchange X between actor roles A and B. The alternatives are sending message Y from B to C or message Z from B to D. This decision is reflected in the process model by an exclusive gateway in pool B, followed by two sending activities Y and Z. Pools D and C feature the corresponding receiving activities preceded by an event-based gateway, which not only waits for the potential interaction to happen, but also for other events - indicating that interaction Y or Z may not happen. Such events could be further interactions or even a timer event to prevent the process from waiting indefinitely.

Such developments of choreography languages, notably those of Let’s Dance, were considered in the development of process choreography in BPMN 2.0. In the BPMN 1.1, a *collaboration diagram* type was available to model interactions across participant processes captured within pools (optionally partitioned in lanes of pools). The interactions between processes were captured using message flows (as depicted in Fig. 1). As such, collaboration diagrams entail inter-connected processes and can lead to inconsistencies and deadlocks, as described above. This led to the new proposal of a BPMN *choreography diagram* with choreography activities proposed as the way of explicitly modelling message interactions through choreography diagrams without recourse to “wiring” up process models, i.e. drawn from an *orchestration diagram* type. A problem with introducing additional behavioural logic in the choreography models is that it increases the complexity

of the model, making it practically useful only for *individual* conversations to be modelled. Hence, BPMN 2.0 also supports groupings of interactions through *conversation diagrams*, which provide the highest level of process models used to understand the B2B participant landscape. With the proposal of new model types, comes the need for a well-defined alignment across these. Consider the following examples which illustrate this.

Figure 4 shows an example of a BPMN 2.0 conversation diagram, providing support for a participant, “birdseye” perspective and groups of interactions or conversations between these, as originally motivated in (Barros et al. 2007a, b). Participant roles (e.g. *Retailer*, *Supplier*) are captured (through box symbols) and connected to conversation symbols (e.g. *Delivery negotiations*). As can be seen, the conversation diagram types allows modellers to understand the broad interaction dependencies of participants, without needing to understand the details inside processes of participants, as captured through the collaboration diagrams, or the details of interaction sequences, as captured through the choreography diagrams. This opens up the possibility of allowing modellers to understand how conversations are “chained” together in support of the systems requirements as a whole. In Fig. 4, we can see an informal indication of this through the conversation dependencies. Barros et al. (2005) define patterns describing conversation dependencies. An example is conversation hierarchy, where conversations are broken up into further conversations. This is indicated with the “collapsed symbol” on *Delivery negotiations*, which says that it consists of interactions which are made up of other interactions. Another example is conversation splitting, where one conversation between participants splits up into one or more conversations with other sets of

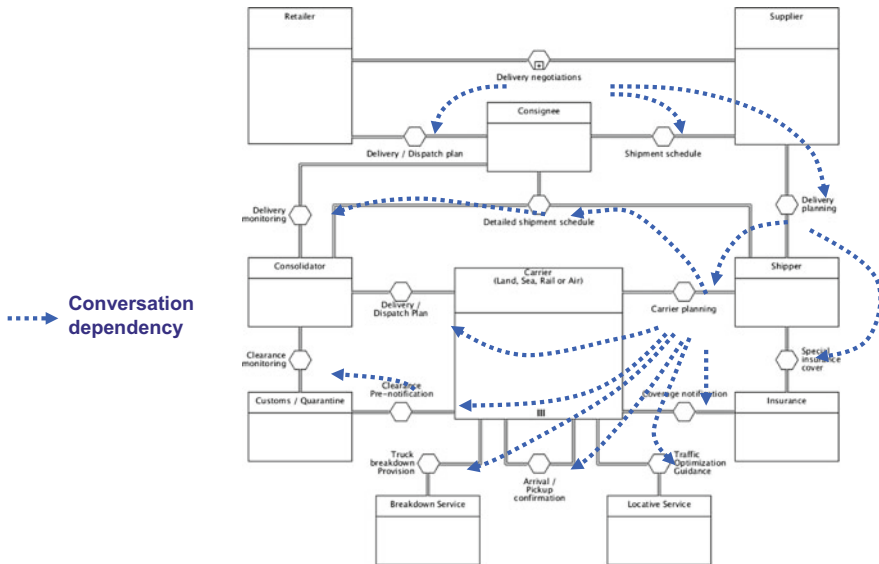


Fig. 4 Example of a BPMN 2.0 conversation diagram

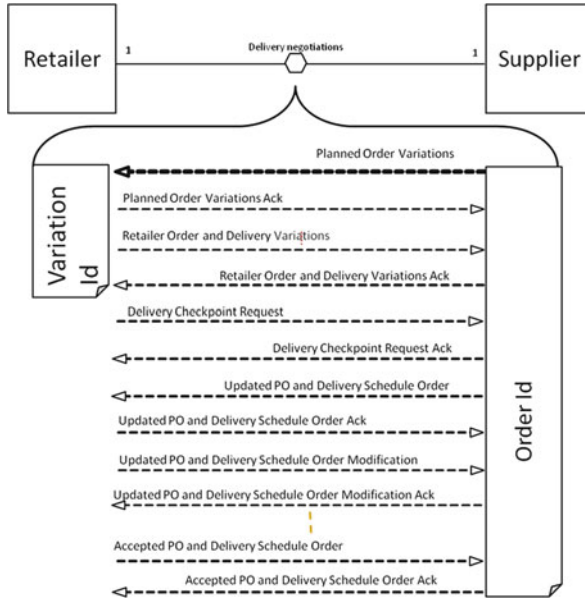


Fig. 5 Example of an expanded conversation

participants, e.g. *Delivery planning* splits up into *Carrier planning* and *Special insurance cover* interaction sets.

Figure 5 shows the interaction sets underpinning a conversation. These consist of reciprocal message flows between the participants. This is similar to collaboration diagrams of BPMN, except that the process details inside the participants are omitted and the correlation keys for the message flows are shown. The correlation key (e.g. *Order Id* and *Variation Id*) consists of the data elements used from inside the messages exchanged to associate the messages with processes. The participant processes use correlation keys to send and receive messages for proper communication to take place. In the example, message exchanges take place for an order only (correlated on *Order Id*) and some relate to variations of the order (*Order Id* and *Variation Id*).

In Fig. 6, we illustrate the BPMN choreography diagram depicting the behaviour of the message interactions for the *Delivery negotiations* conversations. Clearly, this diagram could be used to provide a more detailed behavioural elaboration of the essentially structural, conversation diagram and its conversation expansions. As stated above, choreography diagrams are typically best captured for individual scenarios corresponding to individual conversations.

Finally, in Fig. 7, we illustrate the collaboration diagrams and orchestration details inside these corresponding to the mapping from the choreography diagram of Fig. 6 (some of the message interactions have been omitted for brevity).

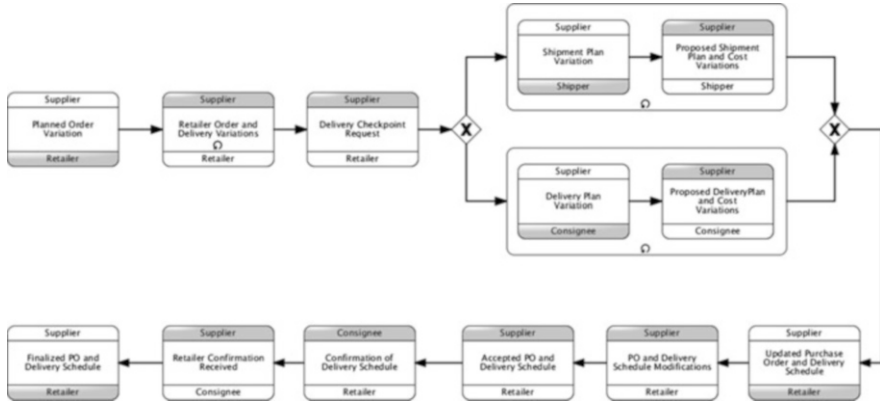


Fig. 6 Example of a BPMN 2.0 choreography diagram

3 Choreography Modelling at High-Level Process Analysis

To provide an impression of the complexity involved in B2B domains beyond the individual scenarios that are typically used to exemplify various choreography language proposals, consider the following:

Logistics, broadly understood, has the goal of fulfilling sales orders between buyers and suppliers, potentially spanning national boundaries. The process is triggered through a sales order and involves the management of shipments involving carriers and potentially different modalities (air, sea and land). Different parts of the order can be shipped from different suppliers, and shipments starting from different origins can be consolidated at different warehouses whose capability

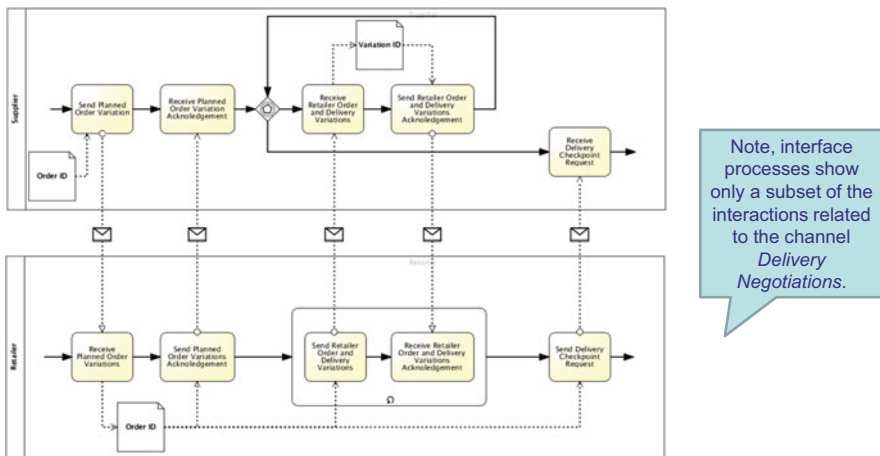


Fig. 7 Example of a BPMN 2.0 collaboration corresponding to a choreography diagram

(e.g. availability of freezing facilities) and capacity for different stock vary. Shipments that cross national boundaries need clearance from regulation authorities such as customs and quarantine. Payments for large or expensive shipments are made through letters of credit, whose monitoring and fulfillment need on-going interactions with banks or payment intermediaries. Each one of these requirements entails different parties in different processes, leading to different conversations with a variety of start conditions, exceptional conditions and object types.

Logistics concerns not only one-off sales orders but also sales contracts established over a certain period, e.g. a year, with replenishment quantities of line items subject to change over a rolling-wave (e.g. next 3 months). To sketch the organizational scenario:

The buyer (e.g. a supermarket) having determined supply requirements through market and relevant purchase patterns establishes a replenishment contract with each supplier (wholesalers of dairy, fruit and vegetable, meat etc.) over a period. Contracts identify periodic delivery at specific times. Variations on replenishment can occur after contracts are established, however within rolling wave periods (e.g. next 3 months), strict obligations are required for replenishment. Any deviations in time and materials which violate replenishment thresholds defined in the contract, lead to financial liability for the supplier. In addition, ad-hoc orders can be requested during the rolling wave.

Since value chains in practice feature tens to hundreds of stakeholders, the process of capturing a choreography needs to be incremental, iterative and detailed at the right level, to shed light on requirements in the first place, prior to detailed validation and implementation concerns. Some parties come to the fore through analysis of the operations of others. Other parties fade into the background as their operations are seen as ancillary. Only when the system landscape stabilises around common functions can detailed modeling of collaborations proceed.

To support the choreography modeling for the wider spectrum of analysis and design, the requirements, discussed in the following sections, are considered crucial.

3.1 Functional Scoping

For choreographies to be comprehensively modeled across a wide variety of requirements related to different business operations, models need to be carefully scoped and freed of unnecessary requirements. This would focus analysis on a related set of business requirements. In the logistics example, procurement of sales, establishment of a sales order/contract, assignment of carriers, and payments & exceptions are distinct and considerable business concerns, each entailing significant requirements for collaboration across different partners. Before the details of message exchanges can be properly discerned, a firm understanding of the following sorts of contextual issues needs to be established:

- What partners are involved and specifically, which of their functional areas are involved? What is the risk of their inclusion (or non-inclusion) given their current and future strategic directions?
- What are the broad business operations from the functional areas that are involved? In what ways do they need to be transformed (e.g. outsourcing decisions)? What problems for integration do they present (e.g. information, service or resource redundancies, bottlenecks and disconnections)?
- What scenarios are involved and do they cohere with the common functional areas? What would be the impact of broader restructuring of coordination?
- What are the different systems involved and, again, what problems of integration do they present (e.g. redundancies, bottlenecks, disconnections)?

Addressing these requires insights and consensus from different stakeholders with a variety of perspectives, be they: internal or external to an organization; strategic, tactical or operational; marketing, sales or delivery; regulatory or commercial; specific cases or concerned with overall analytics etc. In diverse value-chains, analysis of the many and different parts should therefore be focused through carefully scoped functional areas.

Different models for different functional considerations can arise by decomposing them from a common, ancestor choreography model. However, in diverse value-chains featuring related yet distinct areas – like product merchandizing, sales, transportation, payment & exception processing – starting from same process and refining models is unnatural. While these choreographies may relate to each other through shared interactions, it is not natural to think of such diverse processes as refinements of a common starting point. Indeed, this would lead to conceiving of an entire organization through a single high-level process.

Thus, we require dedicated mechanisms for supporting the scoping of choreography models. This would facilitate effective analysis of wide-spanning choreographies through common functional areas. Identifying common areas, indeed the basis for commonality, is not straightforward. Commonality could relate directly to existing organizational units, business activities or services. Under modern practice of enterprises however, processes should be expected to cut organizational boundaries, be utilized through different markets (e.g. a logistics company could support customers in health, manufacturing and high-tech) and delivery channels (e.g. franchises, subsidiaries and resellers of a company and its services).

3.2 Stepwise Refinement

In addition to the scoping of choreography models, refinement/decomposition is a well known mechanism used to manage the modelling of non-trivial processes.

Choreography languages such as WS-CDL and Let's Dance and use classical process decomposition through which an ordered set of interactions (e.g. purchase order validation) are contained in sub-models. Choreography sub-models, as such,

are used to simplify their parent models, leaving certain details to lower level models. Sub-models may also be reused in other models, allowing common functionality referenced in a variety of models.

However, a distinct feature of B2B value-chains is the number of different partners and the range of interactions that can take place for shared concerns. This can lead to cumbersome sub-models that are hard to comprehend outside the explanation of those who created them. To address this problem, extensions have been proposed for a *structural* aspect of choreography modeling, as we saw in Fig. 4, and also in Let's Dance's role-based choreography views. This allows a modeler to depict the presence of many conversations in a single choreography model diagram.

Role-based views have been introduced in Let's Dance and BPEL4Chor (Decker et al. 2008). A major limitation of these proposals, however, is that a *single* modeling level is used to abstract details of interactions. For choreographies with a large number of interactions, it limits the modeller's freedom to introduce as many levels of abstraction in order to describe a conversation with different levels of detail. Too many details of interactions are introduced at the same level, limiting the comprehensibility of individual conversations.

In contrast, classical analysis and design techniques such as Data Flow Diagrams and Structured Analysis Design Technique (Yourdon 1989) allow for *stepwise refinement* of models. Although quite general and lacking in a precise meaning, these techniques are typically applied in large-scale projects to capture interactions between functional entities (which include business processes). Once models are refined at detailed levels, a behavioural perspective is introduced to capture sequencing dependencies of actions being modeled. Being informal, these techniques require the modeler to form correspondence between structural and behavioural aspects.

Clearly, stepwise refinement of choreography models should be supported, incorporating a structural perspective depicting conversations and reciprocal message exchanges (the "Birdseye") and behavioural perspective providing message ordering details.

3.3 Conversation Semantics

Message exchanges in choreography models generally designate request-response patterns between collaborating partners. Message exchanges, as discussed above, are logically related to conversations which are intended to achieve a particular outcome (e.g. creation of a sales order or the preparation of a shipping contract). This is the case for even complex conversations in which, for example, request-responses can become nested at different levels and cascaded to other partners (e.g. assignment of external carriers) not involved in the highest request-response directly related to an outcome (e.g. fulfillment of a shipment contract).

Understanding when message exchanges have been sufficiently captured is a problem of requirements validation that is peculiar to choreographies. For well-established business operations, the insights developed through requirements analysis can lead to an adequate capture of message exchanges, and present practice can drive the validation of the different scenarios. If, on the other hand, a system is being extended or an altogether new system is being embarked upon, that assumption is far less likely to hold. Modelling of choreographies at the *conceptual* level is aimed at minimizing as far as possible inadequacies of supporting requirements which are determined at the more expensive phase of implementation. Since B2B value-chains encompass different partners, business processes and applications, the problem of insufficiently capturing requirements has a wide impact and therefore cost.

Current choreography techniques do not offer ways of guiding modelers towards sufficiently captured and validated models. Apart from soundness checks for livelocks, deadlocks and termination that has been the subject of a considerable research in workflow analysis techniques (van der Aalst 1997), choreography models remain susceptible to semantic discrepancies. This is, of course, true of business process modeling techniques in general. However, choreography language developments, having being steered mostly from the Web services community, have not engaged on techniques from conceptual modeling that have been specialised on collaboration.

In particular, action-oriented techniques (Agerfalk 2004; Dietz 2006) were proposed to explicitly model pragmatic aspects of human language in order to understand collaborations semantically – beyond the goal of achieving interoperability. Action-modelling techniques draw from Speech Act theory (Searle 1969) to explicate the *intent* of interactions between actors. The fundamental idea, determined from an understanding of how humans communicate, is that through a word or sentences, a *speech act* is performed. This is qualified by further components, most notably an illocutionary act which expresses an actor's intention (e.g. make an offer, request a quote, etc.); and a propositional act that refers to some propositional content and identifies what it is being talked about (e.g. an offer referring to a product, a sequence of tasks to be conducted in the future).

Speech acts formalise the social meaning of collaborations, e.g. initial requests, promises or obligations to act, and ensuing action. Consequently, they can be used to develop negotiation patterns so that message exchanges can be understood from the context of interactions that are taking place. A technique, DEMO (Dietz 2006), utilises Speech Acts to model interactions and provides some insight. Based on the illocutionary act (the intention of what is being said), DEMO identifies three phases within an interaction:

- The *offer* phase is made up of two speech acts, namely request, where an initiator requests something from an executor, and promise, where the executor promises to fulfill the request.
- In the *execution* phase, the executor executes what has been promised and thereupon states the fulfillment of the promise to the initiator in the result phase.

- In the *result* phase, the initiator then accepts the execution as being what has been requested and promised.

DEMO uses the illocutionary act to express how a speech act is to be taken. This is especially useful as the social context is implicitly or explicitly constituted by the intentional network of coordinating actors. When it comes to implementation, representational concepts are derived from this context. In that sense, context is determined by the potential actions, e.g. usage (make, accept, reject) of an offer.

Other approaches based on speech acts are Coordinator (Winograd 1987), SAMPO (Auramäki et al. 1988), Action Workflow (Medina-Mora et al. 1992; Denning and Medina-Mora 1995), MILANO (De Michelis and Grasso 1994), BAT (Goldkuhl 1995) and Action Diagrams (Agerfalk 2004).

A major critique of traditional action-oriented modelling approaches is their usage of interactional patterns which are too restrictive. For instance, consider Winograd's action for conversation patterns (Winograd 1987), Medina's workflow loop (Denning and Medina-Mora 1995) or DEMO's simple request, state, accept pattern (Dietz 2006). Here, a requirement for using individual speech-acts for compositions of conversational actions must strive for maximum flexibility. From an empirical point of view, this is quite obvious since anything (e.g. interruptions, re-questionings, sudden withdrawals etc.) can happen during conversations and thus it should be possible to refine actions towards arbitrary complex coordination between actors. A second critique is related to the refinement of conversational networks towards executable representations.

4 Illustrative Modeling Proposals

This section illustrates modeling proposals that address the following of the requirements for choreography modeling that have been identified in the previous section:

- Functional scoping
- Stepwise refinement
- Conversation semantics

4.1 *Functional Scoping*

The scoping of choreography models, as discussed in the previous section, is required to bring distinct areas of B2B value-chains into view, allowing detailed analysis to proceed from a wider perspective. To illustrate how model scoping applies to choreographies and some of the subtle issues of supporting what seems to be a rather simple requirement, consider Fig. 8. It depicts some of the different

Fig. 8 Choreography domains



functional areas of the Sales & Logistics case study, hereafter referred to as *choreography domains*.

Choreography domains (depicted as ellipses) provide the highest level of scoping for choreography models. As indicated in Fig. 6, more detailed sub-models of choreographies are associated with – indeed *contained* in – a given choreography domain model. For instance, Let’s Dance provides role-based, milestone-based and interaction-based sub-model types, and each of these would be contained in a domain model. Domains could also be associated with other organizational artefacts (e.g. organizational units, resources and policies) that are not explicitly used in choreography modelling but which are supported through, say an enterprise modelling framework that a choreography modelling tool “plugs” into.

As with the functional areas in a value-chain, domain models have dependencies with other domain models (seen by the adjacencies of ellipses). In the context of choreographies, this means that they share message exchanges. As examples, Collaborative Forecasting Product Replenishment (out of which an order is produced) connects with Logistics (governing shipment of goods) and with Collaborative Forecasting, Planning and Replenishment; Logistics connects with Payments and Exceptions. Dependencies between domains could be derived through the message exchanges of models that they contain, or the modeller may enforce dependencies at the domain level, thus constraining the scope of message exchanges in their contained models.

From Fig. 8, it can be seen that domains can be hierarchically structured: Logistics is decomposed into Carrier Appointment, Delivery and Claims & Returns. Large and complex domains may be decomposed at an arbitrary number of levels. Thus, a given domain can be decomposed into leaf and non-leaf domains. However only at leaf-levels do domains have models directly contained in them (non-leaf domains are purely used for abstracting domains).

Given that domain models are essentially containers and the concrete details of their choreography are captured in models that they contain, an issue for tooling is synchronizing a domain model. This is because different conversations modeled in different domains would be at different stages of development. Therefore, as different conversations are captured for domains, they need be synchronized and thus be made available for cross-domain interactions.

4.2 Stepwise Refinement and Conversation Semantics

As discussed in the previous section, stepwise refinement and conversation semantics play a part in the detailed analysis of choreography models. Current choreography languages inadequately support these, limiting their suitability for modeling large and complex B2B value-chains. To show how they can be supported and are closely related, the extension of Semantic Object Model (Ferstl and Sinz 2006) for choreographies, as proposed in (Hettel et al. 2008), is presented.

Modelling of choreographies entail both structural and behavioural views of message exchanges between roles, as shown in left and right hand sides respectively of Fig. 9.

In the structural view, there are no routing constructs for expressing the ordering of message exchanges. Instead, Speech Acts are used to qualify the intent of a message exchange. The Speech Acts fit a negotiation pattern underpinning SOM’s conversation semantics, as follows:

- Initialising (I) where both roles (actors in SOM) exchange information about the provided service
- Contracting (C) where both roles negotiate the terms of the service delivery/ consumption
- Enforcing (E) where the negotiated services are provided/consumed.

I, C and E identify the type of the illocutionary act (intention) of the Speech Act using a verb, e.g. order, request, confirm, and a noun identifying what is being talked about (propositional content), e.g. goods, delivery. In Fig. 7, a *Buyer* uses an I act to request a quote from *Supplier* for a specific product he is interested in purchasing and the I act from the *Supplier* signifies the corresponding response. While a single request and response feature in the I phase of this negotiation, further message exchanges could take place. With the C act, the *Buyer* places an order, and thus a relationship between the quote and order is implied. In the next step, *Buyer* and *Supplier* commit to provide and consume a service, as such, with respect to the negotiated terms. This service, namely the delivery of the ordered goods, is signified using the *E: Deliver Goods* transaction. In a negotiation pattern, the I and C may be optional depending on whether both roles already know each other and whether a basic agreement has been established between both.

The behavioural view in SOM provides details about the sequence of acts beyond the broader negotiation protocol established in the structural view. Unlike other choreography languages, behaviour is encapsulated within roles and not

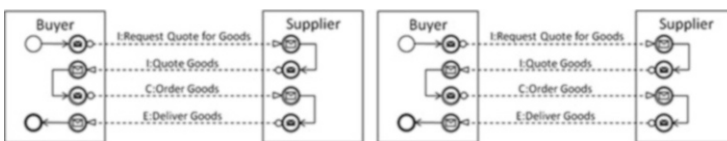


Fig. 9 Layer 1: Initial structural and behavioural view

across roles (e.g. choreography activities in the between pools as has been proposed for BPMN 2.0). This arguably provides more flexibility for the way roles act and respond to speech acts. For detailing the behaviour of partners, a BPMN-like notation was chosen with sending and receiving intermediate events linked by message flow edges. Sequence flow and gateways can be used to specify how one partner acts and reacts with respect to speech acts with others. When considered in isolation, none of the partners has a completely specified behaviour. It is only in connection with other partners that a complete behavioural description can be derived.

In support of stepwise refinement, reminiscent of classical analysis and design techniques like Data Flow Diagrams that have been prevalent in commercial projects for value-chain analysis, roles can be decomposed in order to reveal further roles. Figure 10 provides some details of a refinement of the SOM model shown in Fig. 9 (layer 1).

As depicted in Fig. 10, a number of decompositions have been applied. *Buyer* was decomposed into *Procurement* and *Consignee* interacting according to the feedback-control principle: the management role *Procurement* acts as a management role regulating (R) the operational role *Consignee* by sending an advice to receive goods, whereupon *Consignee* replies (F for feedback) by confirming the

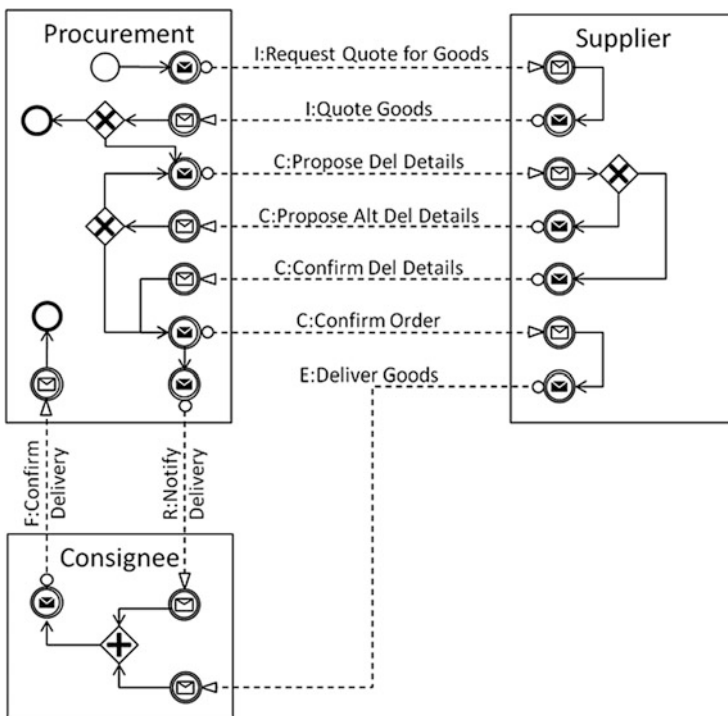


Fig. 10 Layer 2: Behavioural view showing the decomposition of buyer into procurement and consignee

receipt of the delivery. On the right hand side, *Supplier* has been first decomposed into *Sales* and *Logistics*. Furthermore, *Logistics* was decomposed into *Shipper*, *Carrier*, *Consolidator* and *Customs*.

The rule of role refinement requires that speech acts in the parent role be preserved. In Fig. 10, the acts between *Buyer* and *Supplier* have been preserved through *Procurement* and *Sales* as well as *Consolidator* and *Consignee*. Altogether new acts can be introduced between sub-roles of the same super-role, as seen with *Sales* and *Shipper*.

In addition, speech acts and corresponding tasks may be decomposed. As shown in Fig. 10, *C: Order Goods* was decomposed to reveal a detailed negotiation: *C: Propose Delivery Details*, where *Procurement* proposes details (such as date, quantity, quality and price); *C: Confirm Or Propose Alternative Details*, where *Supplier* confirms the details or proposes alternative details; and *C: Confirm Order*, where *Procurement* confirms the order with respect to the negotiated details. A further refinement sees *C: Confirm Or Propose Alternative Details* decomposed into the parallel sub-acts *C: Propose Alt Del Details* and *C: Confirm Del Details*. Here, *Supplier* has the choice between one of the aforementioned speech acts as reflected in XOR gateway. In turn, *Procurement* has a choice between either accepting the alternative details or proposing new details.

Taken together, the interplay of structural and behavioural views, and Speech Acts, provides improved manageability of the complexity and meaning of choreographies compared to that available in current choreography languages. The structural view provides simplified abstractions, holding the broad architecture of the choreography together. The behavioural view, with sequencing details of message exchanges (speech acts) localized in roles, can be developed in tandem with each level of the structural views or can be left to more detailed levels of modeling. Speech Acts on message exchanges provide the bridge between the two views.

4.3 Detecting Errors in Conversations

A major benefit of having conversational semantics, as described above, is the improved model checking that goes beyond detection of deadlocks, livelocks and the like. In particular, it is possible to detect semantic discrepancies in conversations. An insight into these and their detection is now described. The reader is referred to (Hettel et al. 2008) where a formalization of SOM and model checking is presented.

Key to error detection in conversations is the precise description of a conversation in SOM models. So far conversations have been intuited as a set of message exchanges, represented as speech acts between two roles. With Speech Acts, a conversation can be said to encompass all acts that are derived from an initial ICE or RF act between two roles. On a lower layer, a conversation may span several actors. By keeping track of all refinements that have been introduced for acts, different acts can be combined to one conversation. For instance, the Speech Act *E: Deliver Goods* between *Consolidator* and *Consignee* and the other acts between

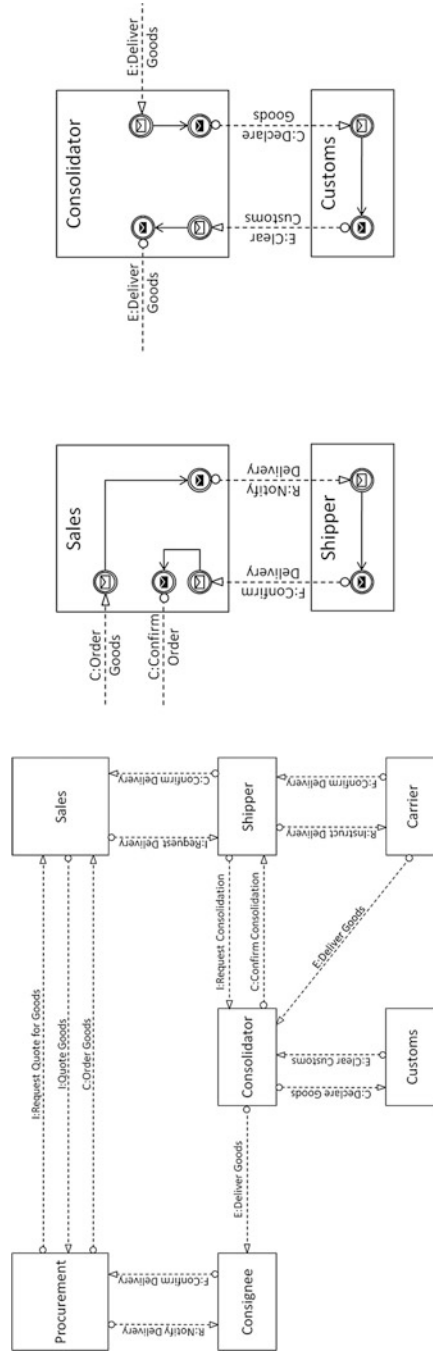


Fig. 11 *Left:* Relevant part of the behaviour involving Sales and Shipper as shown in Fig. 9. *Right:* Relevant part of the behaviour involving Consolidator and Customs as shown in Fig. 9

Procurement and *Sales* together form one conversation as they all originate from the same ICE (Fig. 11).

4.3.1 Negotiation Breakdown

Requirements for successful negotiations may be other subsequent negotiations necessary to arrange additional services needed to provide the overall service. As choreographies model the collaboration of loosely coupled and autonomous roles, participants may withdraw from negotiations at any time, causing it to fail. Such failures may cascade through the model and cause encompassing negotiations to fail as well – leading to a so-called *negotiation breakdown*. A possible negotiation breakdown may be caused by *Shipper*, as an unsuccessful negotiation between *Sales* and *Shipper* may impact on the negotiation between *Procurement* and *Sales* and may cause it to fail, too.

The negotiation breakdown analysis leverages SOM's typed Speech Acts to find subsequent negotiations between third parties that are encompassed in another negotiation. In order for a negotiation breakdown to occur, at least three actors, say X, Y and Z, must be involved, connected via two ICE conversations C1 and C2. Assume X initiates the negotiation with Y. To be able to provide the requested service to X, Y needs to arrange for additional services provided by Z, which has to be negotiated as well. Only when these additional services are secured, the negotiation with X can be closed successfully. A negotiation breakdown can occur when the last negotiation act in C2 leads to the last negotiation act in C1.

4.3.2 Provision Breakdown

Once, two actors have agreed upon consumption and delivery, the service has to be provided and consumed. However, it may happen that after committing to a service provision additional negotiations for supplementary services are required. If any of these negotiations fail, it may not be possible to provide the promised service, causing a *provision breakdown*. For instance, such a breakdown may be caused by *Consolidator* and *Customs* in the example depicted in.

For example, *Consolidator* talks to *Customs* after it received the goods from *Carrier*. If customs cannot be cleared for these goods, then the promised delivery cannot be made. This may pose a serious problem to other partners as they may be held liable to pay compensation for violating the contract. This scenario may be the result of erroneous modelling and therefore needs to be rectified by turning a possible provision breakdown into a possible negotiation breakdown. However, it may not always be possible to model the choreography differently to avoid such situations. Customs cannot be cleared upfront without having the actual delivery inspected. In this case the affected actors may consider a risk mitigation strategy to counter such scenarios.

For a provision breakdown to occur, two ICE conversations C1 and C2 are necessary. The two conversations need to be intertwined in such a way that after the negotiation part in C1 is done, more negotiation speech acts follow in C2. Moreover, the service provision in C2 must lead to the service provision in C1. In such a constellation, failing to acquire the service provision in C2 causes a provision breakdown in C1.

5 Conclusion

The notion of choreography has its origins in Web standardization efforts, out of which dedicated modelling proposals have emerged for implementation-specific languages and platforms. Choreographies address collaborations between partners in B2B domains, and focus on message exchanges in particular. Hence, languages and techniques supporting choreography modelling are of relevance across high level analysis, where cross-organizational contexts are necessary to guide requirements acquisition, to detailed design, where cross-partner interaction dependencies need to come into view for detailed specifications of individual and inter-operating processes.

In this chapter, we provided a background on choreography modelling and argued that the current capabilities are mostly suitable for detailed design. This creates a dichotomy for process specifications across modelling and design, despite situational differences in how modelling is applied. Based on insights from a logistics use case, we proposed three requirements for extending choreography modelling so that it could be equally suitable for high-level analysis. The requirement of scoping and stepwise refinement addresses the way models can be developed under the flux of requirements acquisition. In particular, we developed through SOM, a structural view of message exchanges between collaborating partners which simplify the context upon which the details of sequencing are introduced. For the requirement of conversational semantics, we introduced intent behind message exchanges through speech act theory. We discussed how analysis of conflicts in conversations, in the business sense, are possible, specifically breakdown in conversational negotiations and provisions.

Taken together, new insights are available for extending choreography modelling and the further challenges that lay ahead.

References

- Aagesen G, Krogtstie J (2014) BPMN 2.0 for modeling business processes. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1, 2nd edn. Springer, Heidelberg, pp 219–250
- Agerfalk PE (2004) Action-oriented conceptual modelling. *Eur J Inf Syst* 13(1):80–92

- Auramäki E, Lehtinen E, Lyytinen K (1988) A speech-act-based office modeling approach. *ACM Trans Inf Syst (TOIS)* 6(2):126–152
- Barros AP, Dumas M, ter Hofstede AHM (2005) Service interaction patterns: towards a reference framework for service-based business process interconnection. In: *Proceedings of the 3rd international conference on business process management 2005, BPM 2005*. Springer, Nancy
- Barros AP, Decker G, Dumas M (2007a) Multi-staged and multi-viewpoint service choreography modelling. *Workshop on software engineering methods for service-oriented architectures 2007, SEMSOA 2007*, Hannover
- Barros AP, Decker G, Dumas M, Weber F (2007b) Correlations patterns in service-oriented architectures, fundamental approaches to software engineering, FASE'07, Braga
- De Michelis G, Grasso M (1994) Situating conversations within the language/action perspective: the Milan conversation model. In: *Proceedings of the 1994 ACM conference on computer supported cooperative work*. ACM Press, New York, pp 89–100
- Decker G, Weske M (2007) Local enforceability in interaction petri nets. In: *5th international conference on Business Process Management (BPM), LNCS*, vol 4714. Springer, Brisbane, pp 305–319
- Decker G, Kopp O, Leymann F, Pfitzner K, Weske M (2008) Modeling service choreographies using BPMN and BPEL4Chor. *20th International Conference on Advanced Information Systems Engineering (CAiSE), LNCS*, vol 5074. Springer, Montpellier, pp 79–93
- Denning P, Medina-Mora R (1995) Completing the loops. *Interfaces* 25(3):42–57
- Dietz J (2006) The deep structure of business processes. *Commun of the ACM* 49(5):58–64
- Ferstl O, Sinz E (2006) *Foundations of information systems (In German)*, 5th edn. Oldenbourg, Germany
- Goldkuhl G (1995) Information as action and communication. In: Dahlblom B (ed) *The Infological equation, essays in the honour of B. Langefors*, Gothenburg studies in information systems. Gothenburg University, Gothenburg
- Hettel T, Flender C, Barros AP (2008) Scaling choreography for B2B value-chain analysis. *6th international conference on business process management 2008, BPM 2008, LNCS*, vol 5240. Springer, Milan, pp 294–309
- Kavantzaz N, Burdette D, Ritzinger G, Lafon Y (2005) Web services choreography description language version 1.0, W3C candidate recommendation, Technical Report, 2005, <http://www.w3c.org/TR/ws-cdl-10>
- Kemsley S (2014) Business process management and the social enterprise. In: vom Brocke J, Sonnenberg C (eds) *Handbook on business process management*, vol 1, 2nd edn. Springer, Heidelberg, pp 463–474
- Medina-Mora R, Winograd T, Flores R, Flores F (1992) The action workflow approach to workflow management technology. *1992 ACM conference on computer-supported cooperative work*. ACM, Toronto, pp 281–288
- Searle J (1969) *Speech acts*. Cambridge University Press, Cambridge, UK
- van der Aalst W (1997) Verification of workflow nets, vol 1248, LNCS. Springer, Berlin, pp 407–426
- vom Brocke J (2007) Service portfolio measurement. Evaluating financial performance of service-oriented business processes. *Int J Web Serv Res (IJWSR)* 4(2):1–32
- White SA, Miers D, Fischer L (2008) *BPMN modeling and reference guide*. Future Strategies, Lighthouse Pt
- Winograd T (1987) A language/action perspective on the design of cooperative work. *Hum-Comput Interact* 3(1):3–30
- Yourdon E (1988) *Modern structured analysis*. Yourdon Computing Series Press, Englewood Cliffs
- Yourdon E (1989) *Modern structured analysis*. Yourdon Computing Series Press, Upper Saddle River
- Zaha JM, Barros A, Dumas M, ter Hofstede A (2006) Let's dance: a language for service behaviour modelling. In: *Proceedings of 14th international conference on cooperative information systems 2006, CoopIs 2006*, Montpellier