Complex Systems Informatics and Modeling Quarterly
CSIMQ, Issue 3, July, 2015, Pages 63-83
Published online by RTU Press, https://csimq-journals.rtu.lv
http://dx.doi.org/10.7250/csimq.2015-3.04
ISSN: 2255-9922 online

Dynamic Socio-technical System Design based on Stakeholder Interaction

Albert Fleischmann[1], Werner Schmidt[2] and Christian Stary[3]

[1] Interaktiv.Expert, Pfaffenhofen, Germany
[2] Technische Hochschule Ingolstadt, Germany
[3] Johannes Kepler Universität, Linz, Austria

Albert.Fleischmann@Interaktiv.Expert, Werner.Schmidt@thi.de, Christian.Stary@jku.at

Abstract. In order to directly involve stakeholders in socio-technical system design, we argue for streamlining executable process specifications with business process modeling. Due to current agility requirements of organizations, socio-technical system development is considered one of the key activities of members of the organizations. Dynamic process adaptation enable handling the volatility of business operation and IT infrastructure. Subject-oriented process representations are key enablers to dynamic adaptation due to their capability for stakeholders to create directly executable models. In this way stakeholder can be involved in change management pro-actively. Subject-oriented models (i) represent all relevant features required for system control and decision making, and (ii) are executable on demand. This effectiveness enables organizational change in a creative and efficient way, while establishing innovative design and change management tools. Subject-oriented Business Process Management capabilities are reflected in this realm revealing benefits and potential for further research.

Keywords: Process modeling, subject-orientation, executable specification, dynamic control, decision making.

1 Introduction

The current situation in business (and society) can be characterized as volatile and highly dynamic, both, on the level of IT infrastructures, and business operation. For instance, banking support systems as backbones are increasingly attacked cf. [1] and it costs banks up to $100k per hour [2]. In addition, manipulations or interventions influence the flow of information cf. [3]. Finally, the organization of the financial sector depending mainly on investment and central bank decisions easily affects business operations [4], and thus requires changing business processes dynamically [5].

Chen et al. [6] have investigated the mediating role of business process agility and the moderating roles of environmental factors. Even though organization-wide IT capability may present the characteristics of rarity, appropriability, non-reproducibility, and non-substitutability, its impact on organizational performance is likely to be mediated by business process agility. Moreover, environmental complexity strengthens the effect of IT capability on business process agility (*ibid*.).

The quest for dynamic adaptation of processes challenges the way technology is developed and deployed. The more today's business applications, and thus business logic, can be modeled from a stakeholder perspective before putting it to operation the more likely adaptations can meet business requirements cf. [7]. It seems that regardless of the type of executing or underlying business application it is the process specification that matters, as Seethamraju et al. [8] have shown for Enterprise Resource Planning (ERP) systems.

In order to make use of stakeholder work models we have suggested enabling them to control directly the execution of their process representations in mutual context [9], [10]. Shifting modeling closer to execution seems to be promising, as quite recently Hajimirsadeghi et al. [11] proposed *Processbook*, a social-network-based management system for ad hoc processes carried out to achieve a personal goal. Stakeholders plan towards their goals through models based on to-do lists. They manage these lists in association with personal processes, including how this information can be shared with other users.

In some fields, such as web and database technology, stakeholders (users) have already gained access to execution details, either through diagrammatic languages representing the flow of control, or programming languages 'light', using structured markup notations cf. [12]. Once models become intelligible likewise for software developers and affected stakeholders while containing the relevant control information, modeling techniques incorporate programming (language) constructs cf. [13]. These constructs lay ground for the direct execution of models, and thus dynamic adaptation of processes by stakeholders.

Beyond that, innovation and novel designs require a close semantic distance between artefacts and stakeholders. Ben Shneiderman in his work on human needs and computing technology inspired by Leonardo da Vinci's (1452-1519) has emphasized integrative capabilities on science (scientific outlook), engineering (practical technologies), and arts (artistic skills). He considers human-centeredness to be the ultimate design goal in system development cf. [14], which reads in view of stakeholder-oriented process agility as follows:

- *Human needs are drivers of innovations* - stakeholders need to be provided with proper means of articulation and interactive experience, such as executable process notations, to (co-)develop socio-technical systems
- *Universal usability should be a design goal* - means of communication and information sharing need to be intelligible and usable for all stakeholders for process modeling and execution
- *Innovations should be based on creativity support tools* – a design support tool should encourage stakeholders to model work processes in a variety of ways, in order to represent the diversity of how tasks can be accomplished.

When applying these principles for socio-technical system design Leonardo's Laptop [14] becomes an Organization's Living Design Memory cf. [15]; [16]. Revisiting the capabilities of subject-oriented business process modeling [9], [10] aims not only adapting business processes to actual and current stakeholder needs, but also pro-actively creating organizational however individualized processes. The reflected technique and tool is intended to qualify stakeholders to autonomously (re-)organize the socio-technical systems they are part of, keeping an eye on their interactions rather than functional or role-specific behavior.

Developing organizations towards stakeholder needs requires understanding the current situation of socio-technical systems as a point of departure. Hence, upfront we address relevant context factors, such as globalization, and fundamental enablers of socio-technical system development in Section 2. Section 3 deals with open development issues in process- and stakeholder-driven design of socio-technical systems. The roadmap towards direct system control through stakeholders is detailed in Section 4. We provide fundamental cornerstones and constituents of process-based adaptive systems where stakeholders are enabled to reflect and design socio-technical systems they are part of. Subject-oriented design allows stakeholders taking responsibility in modeling and executing their model representations autonomously and individually, as long as they stick to mutually agreed interaction patterns, i.e., sending and processing messages or business objects. Section 5 concludes the paper wrapping up the findings and providing directions of future research.

2 Context Factors of Socio-technical System Development

In this section we review the understanding of socio-technical systems being part of socio-ecological systems (Figure 1) and their development in dynamically changing environments. Current trends that affect user and technical system behavior are: globalization, the use of cloud computing as infrastructure, and parallel computation improving application performance. Each of them is addressed in one of the following subsections.

2.1 Globalization

Globalization has developed from a catchphrase to a key concept for business planning and operation. It refers to 'the spread and connectedness of production, communication and technologies across the world' (see http://infed.org/mobi/globalization-theory-and-experience/). Besides the distribution of goods and services the speed of communication and (ex)change and the complexity and size of the networks involved have increased dramatically cf. [17]. Consequently, organizations operate in highly dynamic and volatile environments. In such socio-technical environments social relations are of equal importance as the exchange of information as they allow actors to act locally while collaborating globally cf. [18]. This finding is still valid, although an increase of economization of society has been assumed [19]. Consequently, interactions between relevant stakeholders need to be prominent when work models are created.

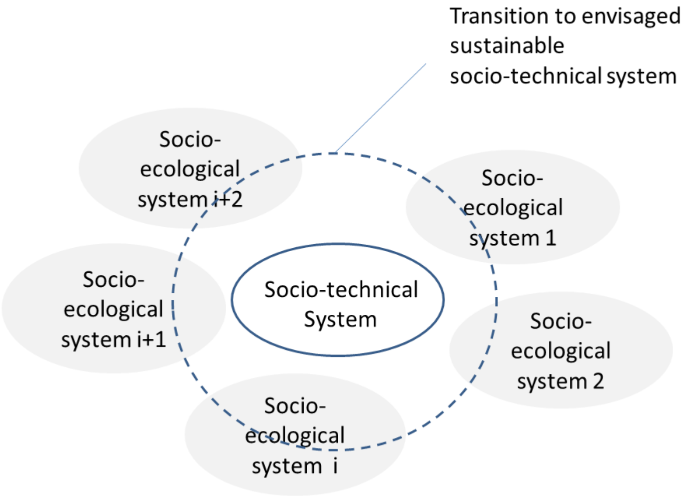

Transition to envisaged sustainable socio-technical system
Socio-ecological system i+2
Socio-ecological system 1
Socio-ecological system i+1
Socio-technical System
Socio-ecological system 2
Socio-ecological system i

Figure 1. Socio-technical systems in the context (according to [20])

As communication occurs increasingly electronically due to global work distribution and social media, the technological infrastructure has to be highly reliable and well performing. Hence, the infrastructure has to take into account the resulting complexity and high dynamics of socio-technical systems. They simultaneously operate across multiple loci, whereas socio-ecological systems are more place-bound [20].

2.2 Cloud Systems as Infrastructures

Cloud computing provides application software as service over the Internet by means of 'infinite' hardware resources, eliminating up-front commitments and paying for resources when needed [21]. The economic case for cloud computing has gained widespread acceptance, albeit causing the urgent need for understanding the business issues for the involved stakeholders such as providers and users [22] and discussions about cloud computing security [21].

Public and private clouds bundle computational resources through providers building large data centers at reasonable costs due to concentrating processing power and load balancing. High speed networks form the backbone for on-demand computing and dynamic scaling [23]. It enables applications and data distributed over various clouds, including software development processes. The latter require communication in network structures, albeit the current focus seems to be on data (containers) and remote procedures (cf. Salesforce) rather than the flow of control (exchange of message encapsulating data).

Platform as a Service (PaaS) cf. [24], [25] provides an environment for developers to construct applications using Internet-based connection. The respective services are hosted in a cloud system and accessed via web browser. The tools for building software applications are provided by the platform. Users may use PaaS through preconfigured features by subscribing to them, and include features they need to meet their requirements. In case of process management systems, existing executable models could be identified and run in case they fit the need of an organization. Hence PaaS-based development could comprise utilizing operating system services, server-side scripting, data management, tool and server hosting, and network facilities.

Developers profit from those services, as they can use individual PaaS environments at every stage of the process to develop, test, and ultimately host their results:

- **Focus on development**. Developers need not to invest in physical infrastructure as they rent it to develop and run their applications on a virtual infrastructure.
- **Stakeholder orientation.** PaaS can offer stakeholders to develop an application via their web browser. Examples of this feature are one-click software process installs for ticketing systems for customer relationship management.
- **Individualized design.** Not only the set of tools can be customized, but also requirements can be implemented in a specific way.
- **Adaptation.** The set of tools and thus applications can be modified dynamically.
- **Availability.** Stakeholders in various locations can work together in a straightforward way – they only need Internet connection plus web browser to run distributed processes.

Consequently, PaaS enables an operating environment not only for developing applications for process modeling, but also executing processes in distributed settings. For changing needs it provides an infrastructure that can be used at any time by all involved stakeholders. Hence, concurrent behavior specifications can directly be implemented through service-based architectures, in particular, when featured by corresponding processor infrastructures (see below).

2.3 Parallel Processing Support

Multi-core chips, e.g., from Intel and AMD, offer a dramatic boost in speed and responsiveness, due to sharing RAM and providing fast local memory (Cache). And the number of processor cores in a die is expected to continue to multiply in coming years making multi-core CPUs with large number of cores a commodity item [26]. The resulting opportunities for multiprocessing have led to the proliferation of multi-core processors in general-purpose computing infrastructures. As such they can be accessed by operative stakeholders in organizations. More than ever, multithreading is a requirement for effective use of multi-core processors. This finding has already affected the development of programming languages [27], such as Scala, an object-functional scripting language with concurrent elements, designed to program solutions in an intelligible and type-safe way (www.scala-lang.org).

Using object-functional infrastructures with concurrent elements system behavior can be described in an actor-centered way allowing for parallel activity support in socio-technical systems. Corresponding tools for process support, such as UeberFlow [28] are promising candidates for execution of concurrent processes, in particular when providing open interfaces for importing and exporting model specifications.

3 Open Issues in Socio-Technical System Development

While business process modeling opens up towards semantic representations providing stakeholder-centered modeling requirements (Section 3.1), at the execution level stakeholder support still is an open issue (Section 3.2).

3.1 Social-technical BPM

Business Process Management (BPM) [29] has been recognized to be a key to adaptation, thus enabling organizational agility [30]. Traditional BPM techniques do not support comprehensive socio-technical system (re-)design, neither with respect to modeling or execution when engineering change management (*ibid.*). Even recent approaches such as ontology-based modeling and multi-dimensional representations [31], or communication-oriented BPM [9], [10] do not provide a sufficient, coherent, and holistic representation of organizations as socio-technical system for engineering change.

Models need to be concrete in terms of stakeholder

- involvement, e.g., expressed through roles or role-specific behavior elements
- communication, either mutually or via applications
- tasks to be accomplished in the course of business operations cf. [9], [10]

Models need to tackle both, concrete systems, such as machines or machine parts, and abstract ones, such as software applications or business objects. However, for service or production the allocation of tasks and design of responsibilities have to be specified: 'The less tangible the capability, the more control will be ceded to the customer' [32]. Industry-specific or packaged applications that still are most prominent in organizations have been aligned with organizational 'silos', i.e., internal functions, such as accounting or engineering. They have worked well for well-defined, highly structured processes. Modeling in these cases was based on the assumption of volume, scale, and straight-through processing. Today, the applications resulting from such approaches are not only difficult to change, but rather form a barrier to adapt and innovate, in particular, for organizations in fast-moving industries like automotive (*ibid.*).

The first step to reflect existing processes and build upon existing operational knowledge is to introduce a rigorous defined approach for specification together with involved stakeholders, calling for contextual design of business operations, also enriching the basis for organizational innovation cf. [33].

3.2 Stakeholder Control

Asanovic et al. [34] have claimed that writing programs that scale with increasing numbers of cores in multiprocessor environments should be as easy as writing programs for sequential computers. Such a vision requires to go beyond programming language extensions, and to find efficient thread-to-core assignment policies for applications developed by stakeholders engaged in business operations. As they operate applications they could be engaged in end-users programming. Languages like Visual Basic are intuitive due to their inherent simplicity, requiring little technical expertise to use it. Hence, they allow stakeholders focusing on their domain-specific knowledge – a requirement for stakeholder-centered language design.

Applications developed by stakeholders in the sense of end-user computing require open development cycles, e.g., skipping optimization for the sake of re-engineering. As opposed to industry standard benchmarks and off-the-shelf applications, according to Asanovic et al. [34], stakeholders should be enabled to identify relevant components at run time when triggering change processes.

Although stakeholders could work on a cloud infrastructure in a cost-effective way (based on transparent costing of processes – [35], they might not identify the actual runtime or computer system for processing. As they are less likely to be computer architecture experts, developers need to apply architecture languages, such as ArchiMate (www.archimate.org) for transforming strategic objectives and operational procedures to services in a transparent way. It would allow collaborative fine tuning of application behavior based on the architecture of the underlying infrastructure. However, specifications should hide asymmetries in multi-core architectures, in order to obtain a steady performance.

For application behavior tuning service orientation has been propagated to execution adaptation by Holmberg [36]. He has designed a Service Oriented Business Process architecture managing the separation of the concerns, process, and decision logic, which impedes the flexibility of business information systems for business agility. Business rules can be combined with business process modeling based on a service-oriented architecture for business agility. Separating business logic into process and decision logic became a criterion for bridging the gaps, but simultaneously maintaining logical boundaries between the responsibilities of Subject-Oriented Architectures, business rule applications, and BPM. However, the effects of business logic separation were two separate species of digital services composed to provide business service orientation. They both play a role when stakeholders are enabled to model and execute their processes (see Section 4).

4 Modeling for Execution

After having identified current trends in business operation and limitations of support mechanisms for (process) agility, we discuss how existing deficiencies and needs can be addressed to implement a timely and accurate development tool for socio-technical system design from a stakeholder perspective. We identify modeling as the core activity of designing (Section 4.1) and effective execution support as the key feature of corresponding tools (Section 4.2).

4.1 Modeling

Models can become inherent to the work space of stakeholders and, thus, effective carriers of control information. As models can represent the current knowledge of work processes of stakeholders, they can support them in informed decision making on the future organization of their work. The latter can be triggered by stakeholders, when they feel need of change, or by environmental factors, e.g., changes on the market. In any case model representations help reflecting the current situation and developing various design options. In this way decision making can be grounded on articulated evidence before putting changes into operation.

However, developing alternative process options should be accompanied by direct experience, in order to check the impact before actually changing the organization of work and/or produce alternatives. Consequently, models should be executable and, thus, contain all required control information as applications do. In the remainder of this section we address relevant modeling aspects for stakeholder orientation, direct execution, and effective process organization.

Contextual Modeling and Intelligible Representations

One way to achieve stakeholder-oriented representations is to enrich existing (business process) specifications with contextual knowledge. Le Clair et al. [32] expect a new generation of processes within the next 5 years, designed from the outside replacing heavy packaged applications designed from inside-out, e.g., driven by functional deliveries. Those still drive customer interaction today, but cannot keep up with the demands for change given by dynamic customer needs and the complexity of organizing work.

The quality of models plays an increasing role in adaptive process environments. According to Claes [37] modeling behavior relates to the quality of the process model in several ways. A modeler's structured modeling style, the frequency of moving existing objects over the modeling canvas, and the overall modeling speed is in any way connected to the ease, with which the resulting process model can be understood.

As models need to be communicated, when shared and reflected along organizational learning steps [38], stakeholders need to build up communication skills. Otherwise development projects are likely to fail, as deficiencies in communication are, in fact, among the most frequent reasons for project failure cf. [39]. If not being supported by a BPM technique or tools, a design process is likely to require several iterations until completely understood by involved stakeholders.

Behavior Representation: Key to Agility

Early business modeling approaches mainly focused on static structures of data rather than dynamic aspects. Recent approaches go beyond encapsulating information about data and processes in a single object towards complex multilevel abstraction hierarchies cf. [40]. For instance, multilevel business artefacts extend object representations (m-objects) by associating with each level of abstraction a single life cycle model that defines the permitted execution order of the methods of the class associated with the respective level [41]. Such an artefact encapsulates in single object information about the static and dynamic aspects of multiple levels of abstraction. These approaches go beyond nesting process models, as they refer to the diversity of processing data in processes controlled by stakeholders.

Transactional models, such as DEMO [31], have enriched a standard transaction pattern (order, execution, result phase) with four additional cancellation patterns (Figure 2). As the standard transaction pattern is considered universal and complete with respect to all possible communicative actions, such specifications can serve as a baseline for optimizing application behavior with respect to concurrency.

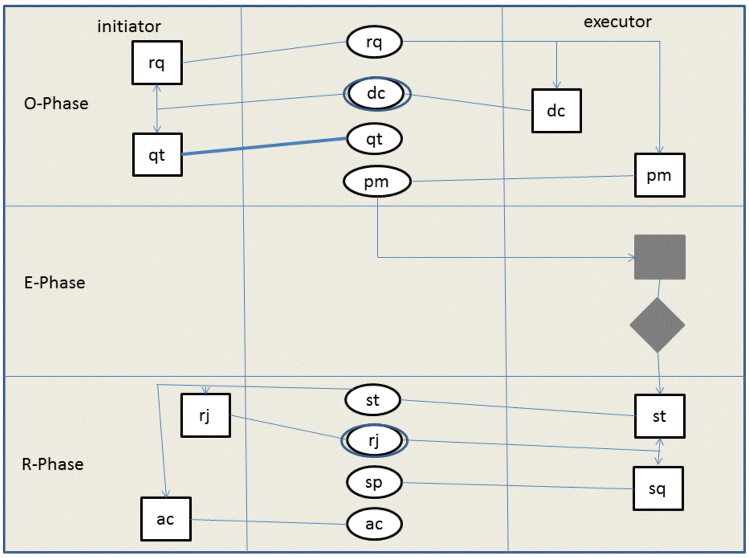


Figure 2. Standard transaction pattern (adapted from [31]) – rq: request, pm: promise, st:state, ac: accept, dc: decline, qt: quit, rj: reject, sp: stop

Such contextual models play a crucial role for implementation. Kutvonen [42] suggested the 'Pilarcos open service ecosystem architecture, allowing stakeholders to manage cross-organizational collaborations collectively, utilizing correctness and acceptability criteria set in ecosystem-wide dynamic processes. The correctness requirements are defined in terms of business network models, policies and service definitions that are dynamically utilized as complex conformance reference points. Failures to conform to these reference points trigger pre-committed recovery behavior' [42, p. 294]. A reference model provides efficient facilities for the design of this kind of complex, adaptive systems. Such approaches serve well for simulating organizational changes cf. [40].

Interfacing Tangibles with Intangibles

Tacit knowledge has been explored as an asset of knowledge management activities [43]. Value Networks [44] have been identified as effective means to represent implicit (intangible) apart from explicit, tangible knowledge. They refer to deliverables from roles not contracted explicitly, such as keeping customers informed when an order is processed, but determine the transactional behavior among stakeholders besides tangibles (i.e., contracted work) - Figure 3.

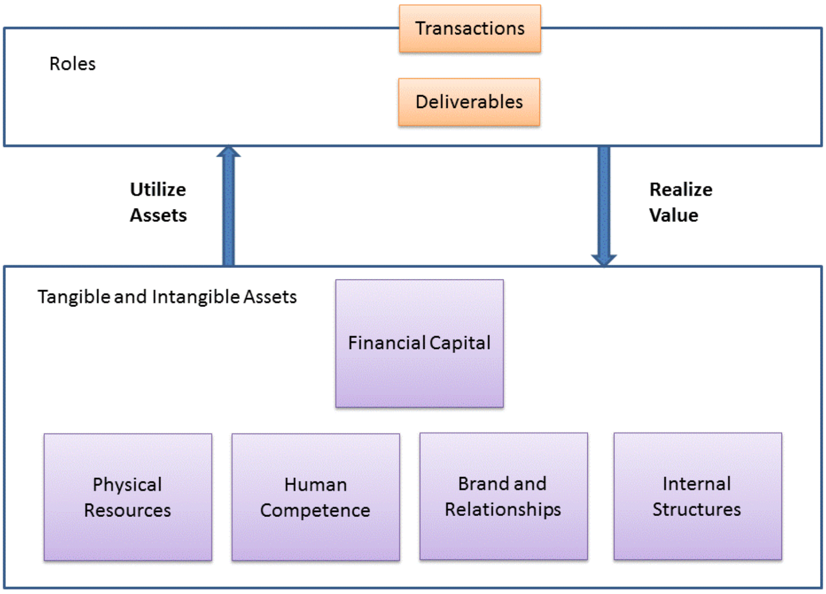


Figure 3. Value Conversion Strategy Model [44]

Relating intangibles to tangible outcomes is relevant when addressing organizational structures [45], and thus, organizational development. However, it requires intervention, in order to convert intangible assets to tangible interactions. Stakeholders may need guidance and background information to specify tangibles for mutual benefit when elaborating the nature of intangible relationship among them cf. [46].

4.2 Execution Support

Dealing with representations of socio-technical systems in the context of organizational agility requires procedures organizing co-creation and adjusted infrastructures for execution, keeping up direct stakeholder engagement in the course of managing change.

Cooperation in Socio-technical Design

Cooperatives have turned out beneficial for crisis and innovation management. As the study by Smith et al. [47] reveals, worker and producer cooperatives have not only benefits during times of economic crises, but also for large and small scale innovations. The latter ‘are contributed by individual members. For worker cooperatives, observations that the workers make in the course of their daily work, whether in the context of building craft products, working on an assembly line, or service work, may be more likely to be mentioned, recorded, and built upon by the cooperative. In this way the cooperative can introduce improvements and new methods of production and organization with the more direct line of communication that their management structure facilitates. This is clearly a comparative advantage of cooperatives over conventional firms’ (ibid., p.11), as long as organizations maintain some sort of exchange between the internal systems of the organization and the external world through bringing in new ideas, resources, and individuals [38]. Consequently, modeling and execution have to be shared in a way that individual models can evolve in mutual context and adjustment.

Reliable and Steady Performance Runtime Cloud

Developers might think about reducing complexity to keep up reliability and performance at the application level through modular intelligent system architectures, such as multi-agent systems, since end-users are less-likely to rewrite or modify their application in order to better utilize the architectural capabilities. Even when organizations utilize advances in technology development to redesign their information and process management systems, the redesigned systems are likely to fail, as strict adherence to prescribed workflows makes it impossible for the system to adapt to unforeseen circumstances [48]. It can be expected that multi-agent systems will be established as workflow enactment mechanisms (*ibid*.). When description languages and their associated design tools are used to specify multi-agent systems, either existing techniques can be enriched or variants of processes can be specified for or by stakeholders.

Buhler [48] shows that Business Process Execution Language for Web Services (BPEL4WS) can be used as a specification language for expressing the initial social order of multi-agent systems, which can be modified to changing environmental conditions.

Ayora et al. [49] propose to collect variants of executable processes. They adopt the idea of process-aware information systems and suggest process model repositories with large collections of related process variants (families). In contrast to previous approaches focusing on the modeling and configuration of process variants, they investigated run-time configuration and re-configuration. According to their findings, such effective handling of process variants requires deferring certain configuration decisions to the run-time, dynamically re-configuring process variants in response to stakeholder requests or contextual changes.

From the cloud computing perspective, dynamically scalable and virtualized resources need to be provided as a service [50]. As Xu et al. could show for the manufacturing industry, it ‘can transform the traditional manufacturing business model, help it to align product innovation with business strategy, and create intelligent factory networks that encourage effective collaboration’ (*ibid*., p.75). Distributed resources are encapsulated into cloud services and managed in a centralized way. Cloud computing has been adopted directly for manufacturing, or as pay-for-IT-as-you-go for scaling production up or down on demand, ranging from product design, manufacturing, testing, management, and all other stages of a product life cycle.

Rethinking transactional systems could help in increasing system performance. As Gramoli [51] points out transaction-based programming can sometimes restrict application concurrency and performance. A thorough understanding of the semantics of an application (as given by semantic models complete with respect to control flow), stakeholders (programmers) could trade simplicity for additional control. In his approach to democratizing transactional programming, neither composition nor correctness need to be compromised: stakeholders can keep concurrency transparent while expert programmers can use multiple synchronization semantics of sequences of shared data accesses to increase performance. Such an approach promotes the coexistence of different transactional semantics in the same application reflecting safety and liveness of operational procedures.

Increasing performance through resource optimization requires such concurrent engineering techniques as overlap, interaction, and iteration of activities [52]. Once sequential process models are replaced, context models are required. They need to express actions and interactions of interrelated internal and external agents (*ibid.*).

Stakeholder-Informed Change Management

Taking into account the efficiency of complex, dynamic, multi scale, and adaptive systems, learning, experimentation, and iteration are essential parts of socio-technical system development, as social processes shape the development and use of technology, but technologies, in turn, open up possibilities for new social practices through active development, linkages, and the alignment of heterogeneous, social, and technical elements into working configurations [20].

Consequently, effective approaches as indicated in the previous sections need to be aligned under the criterion of efficiency. Subject-orientation has turned out to be compatible to both, stakeholder-oriented understanding of processes, and distributed engineering and automated executing process models [9], [10]. For practical use it can be embedded in value network-like organizational learning environments, allowing stakeholders to start with refining their mental work representations to process models [46].

Figure 4 shows the intuitive description language. In Appendix, a brief guide for the creation of subject-oriented models is provided based on the results from recent field studies [54]. Figure 5 provides an order handling example. On the left the standard behavior of the order handling agent is described. The flags mark states in which activities a customer can change his/her order. The middle and the right sides show the behavior in case the customer actually changes the order. In this way, adaptive case management [55] can be implemented – see also Figure 12.

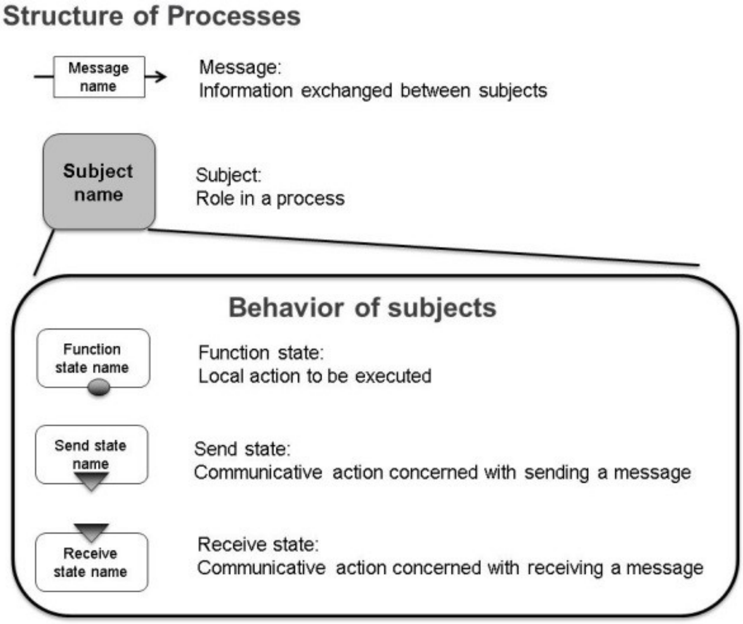


Figure 4. Standard set of symbols in Subject-oriented BPM

Besides constructing models from scratch, business processes can also be designed by restricting communication patterns once all stakeholders or systems involved in a value chain or process have been identified. The procedure reflects the ideas of restricting all-to-all communication as known from e-mail communication to those interactions that are actually required for informed decision making and task accomplishment. It is based on several steps:

1. Specify a generic template according to the number of parties involved in handling a certain business case (cf. Figure 6 for 3 parties involved).
2. Name the subjects according to the application domain.

3. Identify and remove message connections between subjects which are not necessary.
4. Name messages and introduce message types according to the application (domain).
5. Adapt specification to actual subject behavior.
6. Refine the structure of the business objects transmitted by the various messages.

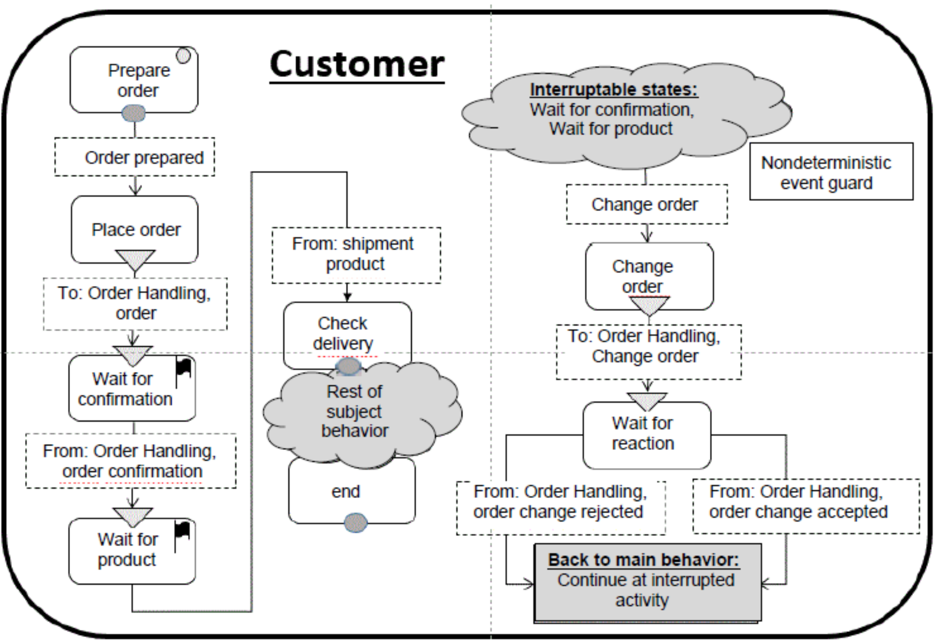


Figure 5. Creating executable process families [53]

In the first step a generic template according to the number of parties involved in handling a certain business case needs to be specified. In principle, each of the parties can exchange messages with another party. Each subject starting message exchange is marked with a 'Play' symbol (small white triangle) in the upper right corner, as for Subject1 in Figure 6.

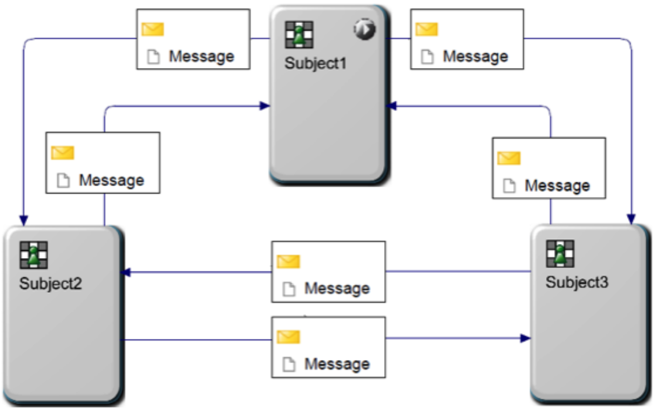


Figure 6. Subject-oriented representation scheme for a 3-party process

As each subject can send messages with the name 'Message' to any other subject any time, the corresponding behavior diagram of each subject can be predefined – see Figure 7 for 'Subject1'. Since Subject1 is the subject, which starts a process, its start state is the state 'select'. Further in the text it will be denoted as 'start subject'. The start state is marked with a 'Play' symbol. The state 'start' and the transitions to the state 'select' will be never executed in the start subject.

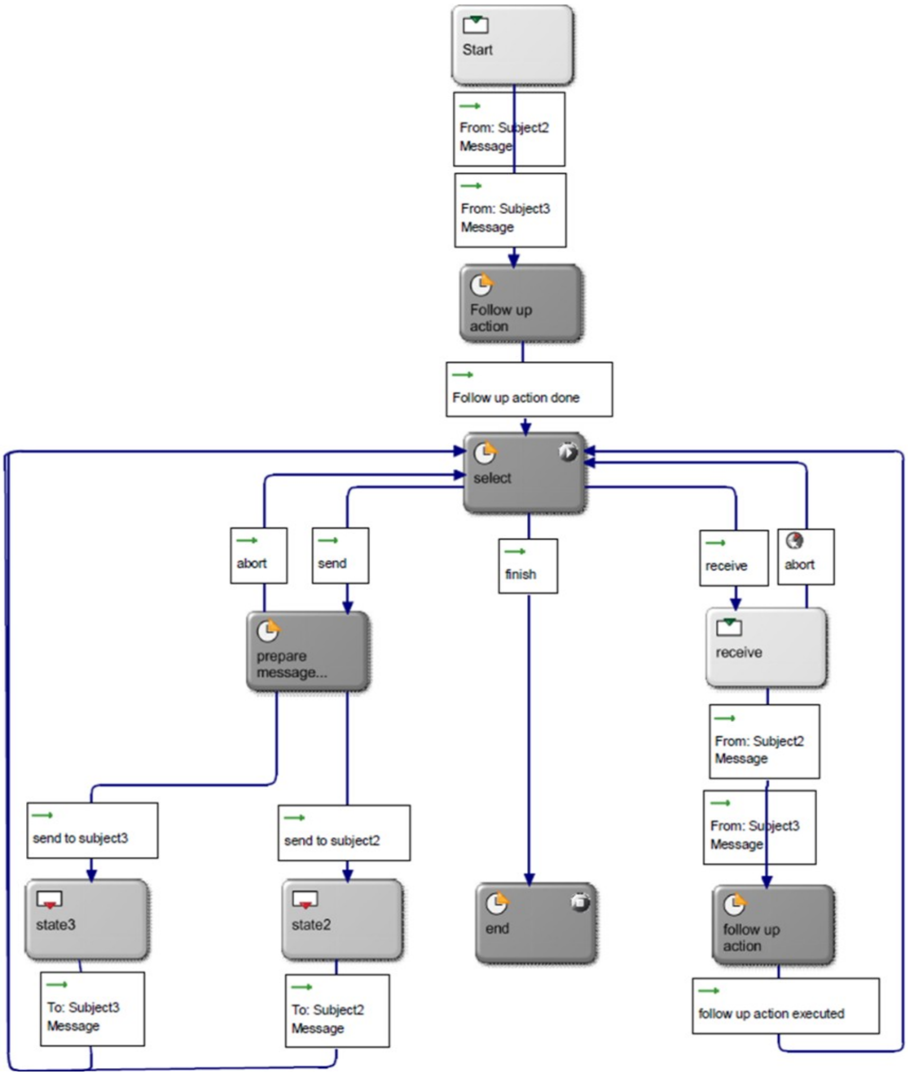


Figure 7. Generic behavior of Subject1 that starts the process

In the behavior specifications of all other subjects the 'start' state is a 'receive' state because they are all waiting for a message of any other subject (see Figure 8).

In this way all subjects that are not start subjects have to receive at least one message before they can start to send messages. The start subject sends a message to any other subject. The receiving subject can now reach the state ‘select’. In that state any subject can decide upon its next action without restrictions. A subject, which is in state ‘select’, can send a message to other subjects, which are still in the state ‘start’. Now these subjects can also reach the select state and can send messages. Finally, all subjects are in the state ‘select’ and can communicate when addressed.

In the 'select' state the start subject decides whether it wants to send or to receive a message. To start a workflow it does not make sense to receive a message because all other subjects are waiting for messages (as mentioned before their start state is a 'receive' state). Consequently, the start subject will start with sending messages and the exchange of messages can begin. Choosing the 'send' transition, the subject moves to the state ‘prepare message and select address’ and fills out the business object (i.e., the data to be manipulated in the course of task accomplishment). That business object is transmitted by the message ‘Message’. After that the subject decides, to which other subject the message with the business object as content will be sent.

In the 'select' state a subject can also decide whether it wants to receive a message - this choice can make sense for a start subject further on when moving into the 'select' state the second time. If there is a message for the subject available, it can be accepted and a follow up action can be executed. It is not specified what the follow up action is. Similarly to receiving an e-mail, the receiver can interpret the content of an e-mail and knows what the corresponding follow up action is. The abort transitions back to the select state enable to step back in case a subject has made the wrong choice.

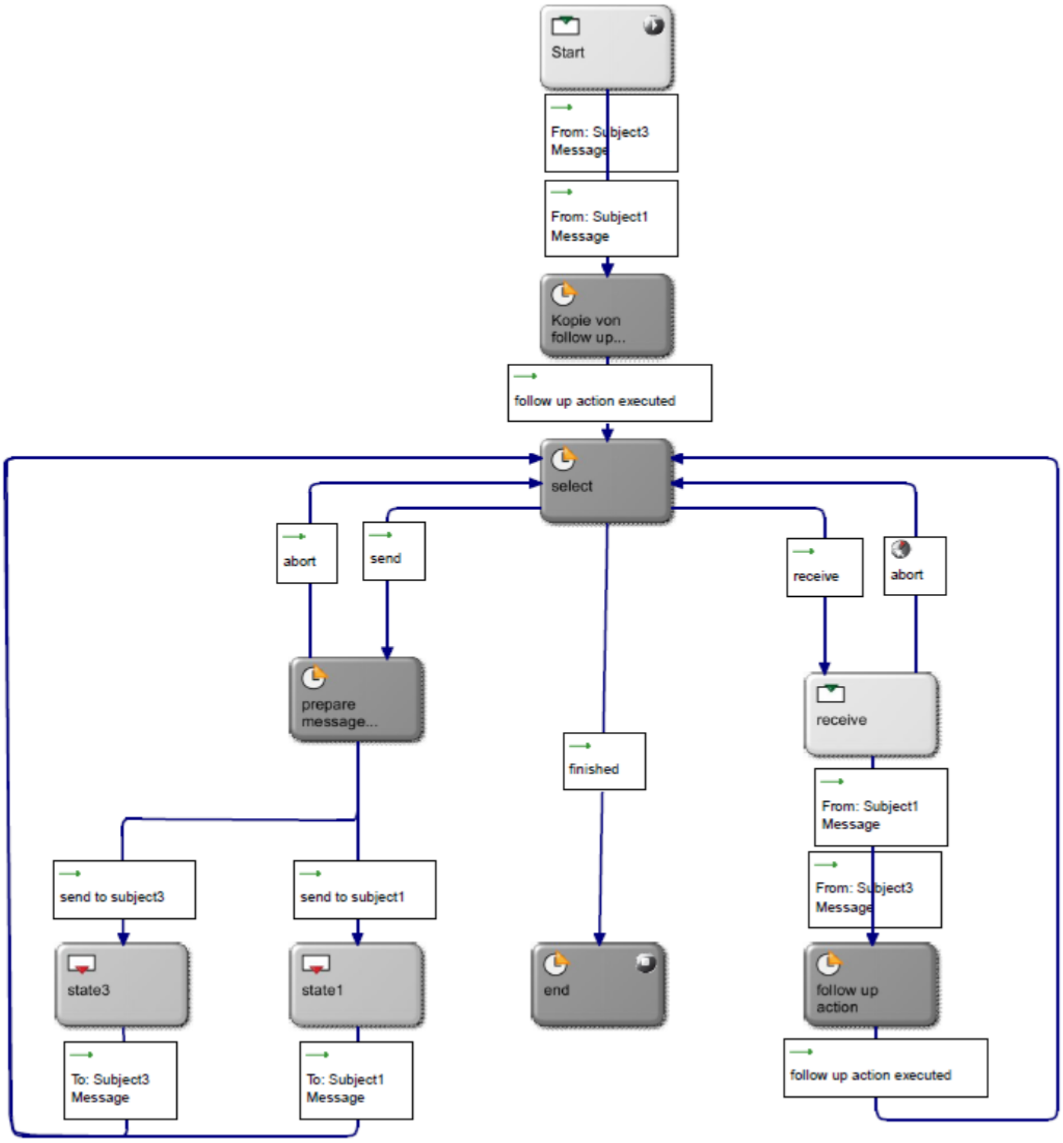


Figure 8. Generic behavior of Subject2

The representation scheme can easily be set up for any number of participants, following the same principles as shown for 3 parties. The behavior of each subject has to be adapted to the number of subjects in a process. In the send area transitions have to be added to send a message to every single new subject, and the same is required for the receive area. Using that extension scheme the behavior for each type of multi-party process can be generated automatically.

With the message 'Message' a business object is sent. The structure of this business object corresponds to the structure of a traditional e-mail with extensions like subject (attention: Here the word 'subject' has a different meaning. It can mean topic, issue, theme etc.), keywords, and signature. Figure 9 details business object 'Message' in terms of attributes and values.

Each step in restricting communication increases the coherence of accomplishing a work task for stakeholders (also termed subject holders), as it becomes more transparent with respect to required inputs for task completion and results. S-BPM guides organizational development starting with a network of mutually communicating stakeholders, and proceeding with reducing their interactions to achieve a focused, actually required communication scheme for accomplishing tasks. Comparing the construction and restriction approach reveals that modeling through restriction does not necessarily result in models identical to those created by modeling through construction. Nevertheless both models need to deliver the requested results of work tasks.

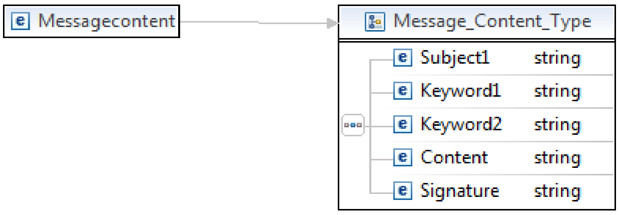


Figure 9. Generic structure of business object 'Message'

The benefits of continuous restriction can be best explained when comparing it with continuous construction. Subjects and communication relationships are specified in a cumulative way in continuous construction. Communications patterns are defined and explored as the respective modelers or stakeholders perceive work procedures. Each model develops over time and represents the current state of business affairs at a certain point in time. It is not linked to a baseline, such as the generic frame of reference for continuous restriction, in order to minimize redundancy or provide a certain structure for design cycles. Consequently, revisiting S-BPM models might cause additional modeling workload for the sake of completeness - the generic message relationships serve as placeholders until being removed in case they are not required for processing the business case at hand (i.e., as soon as they cannot be named according to their task-specific purpose).

Finally, continuous restriction facilitates the automated execution of S-BPM models. The generalized peer-to-peer network (frame of reference) contains all the subjects that are relevant for a business operation at hand. Since it also contains the possible communication relationships between the subjects, this model represents an S-BPM Interaction Diagram (cf. Figure 5). It contains a complete control flow description for generating workflows. Using a corresponding interpreter or BPM suite, such as UeberFlow [28] or Metasonic (www.metasonic.de), S-BPM models can be executed on demand - business processes can be experienced interactively, even when some subjects and messages have not been assigned to concrete actors, systems and message paths.

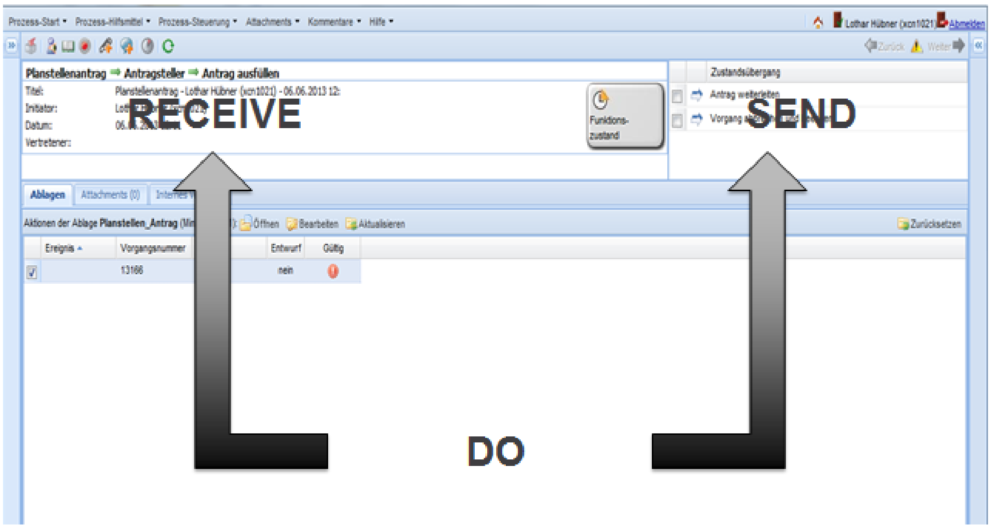
Figure 10 and Figure 11 show interactive experiences created when executing S-BPM models. Figure 10 reveals the screen structure according to the behavior logic for each stakeholder involved in a business process in S-BPM, whereas Figure 11 illustrates the dual view – on the right side the created artefact (in this case the decision situation before approval or rejection) and on the left side — the model including a position marker for the sake of traceability.

Once models are constructed either through continuous construction or restriction they might be modified in the following ways:

- according to the previously described approach, either through construction or restriction
- by enriching them according to the needs of the business case at hand

The latter case is supported in S-BPM on the level of Subject Behavior Diagrams, particularly to create alternative behaviors for each subject when needed. We demonstrate that feature along the communication structure of the order handling process (cf. Figure 5).

Whenever non-standard behavior is included, e.g., due to changing customer needs or emergency situations, processes have to be modified. In particular, in case a customer is able to change orders, adaptations of the models are required. These can be implemented at the interaction and behavior level.



Prozess-Start ▾ Prozess-Hilfsmittel ▾ Prozess-Steuerung ▾ Attachments ▾ Kommentare ▾ Hilfe ▾
Lothar Hübner (xcn1021) Abmelden
Zurück Weiter
Ablagen
Attachments (0)
Öffnen
Bearbeiten
Aktualisieren
Zurücksetzen
Funktions-zustand
RECEIVE
SEND
DO

Planstellenantrag ⇒ Antragsteller ⇒ Antrag ausfüllen

Titel: Planstellenantrag - Lothar Hübner (xcn1021) - 06.06.2013 12:

Initiator: Lot

Datum: 06.

Vertreter:

Zustandsübergang

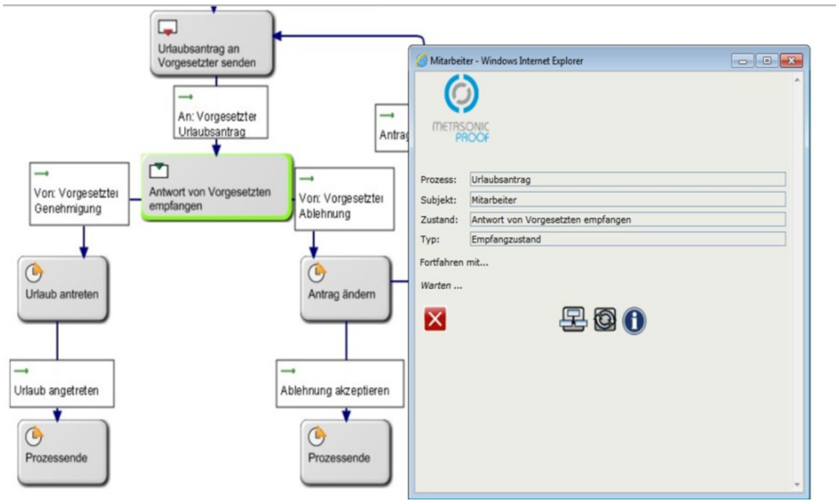
□ Antrag weiterleiten

□ Vorgang

Aktionen der Ablage Planstellen_Antrag (Mi

	Ereignis ▲	Vorgangsnummer	Entwurf	Gültig
☑		13166	nein	●

Figure 10. Executing the behavior of a subject using Metasonic Flow ®



Urlaubsantrag an Vorgesetzter senden
An: Vorgesetzter Urlaubsantrag
Antrag
Von: Vorgesetzter Genehmigung
Antwort von Vorgesetzten empfangen
Von: Vorgesetzter Ablehnung
Urlaub antreten
Antrag ändern
Urlaub angetreten
Ablehnung akzeptieren
Prozessende
Prozessende
Mitarbeiter - Windows Internet Explorer
METASONIC PROOF

Prozess: Urlaubsantrag

Subjekt: Mitarbeiter

Zustand: Antwort von Vorgesetzten empfangen

Typ: Empfangzustand

Fortfahren mit...

Warten ...

Figure 11. Correspondence of execution and model representation on the level of Subject Behavior Diagrams using Metasonic Proof ®

Figure 12 shows the extension of message exchanges as required for changing orders, as it requires approval. In Figure 13 the corresponding customer behavior is detailed, introducing the concept of message guards [56], as it allows continuous refinement according to non-standard business behavior.

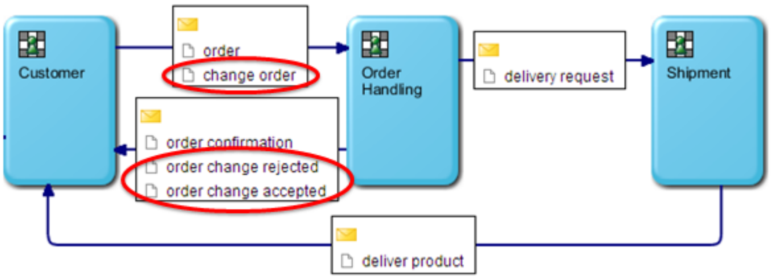


Figure 12. Interaction Diagram including changed requirements

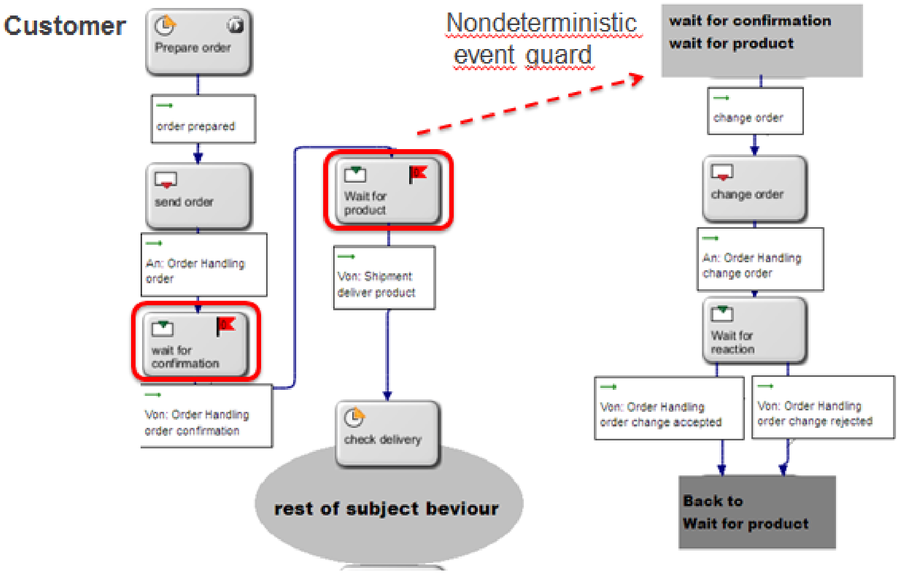


Figure 13. Refinement of modifications in the Behavior Diagrams of the subject 'Customer'

The example indicates how stakeholders can reflect changing requirements in the process model, allowing them to be automatically transformed into the software derived from the model specification. This feature not only minimizes the time spent from articulation of requirements to their implementation in a running system, but also leads to a high level of consistency between the desired and the actual behavior of a software system acceptable as a solution from the involved stakeholders (represented as subjects).

5 Conclusion

Dynamically changing socio-ecological and -technical systems require adaptive designs. Innovation, business and technology developments, in particular, service-oriented architectures and cloud-based platforms, open new opportunities how organizations work and social entities coordinate their work. As involved stakeholders increasingly get engaged in designing their work processes, process agility requires their immediate and accurate involvement. In this contribution we have revisited socio-technical system design and put subject-oriented process specification and execution in that context. Looking at interactions between stakeholders in a business environment, essential parameters and elements of organizing work, in particular, the who, the what, the how, and data can be represented in executable subject-oriented models. Coupling process models tightly with execution allows stakeholders continuously articulating their requirements with respect to process support. The subsequent refinement to executable components of software in terms of subject-oriented representations ensures utmost concurrency due to the asynchronous behavior specification.

It still needs to be investigated in how far such a paradigmatic shift from mainly function-driven BPM modeling towards asynchronously coupled behavior representations can be put into practice consequently. In particular, organizations that are organized in a hierarchical way might experience difficulties with highly parallel structures - subject orientation provides the highest potential when stakeholders individually and in parallel can change their behavior according to their needs, as long as they act along the communication patterns agreed among the subjects (specified through message exchanges).

Finally, S-BPM, due to its capability to precisely describe the execution of process components, is likely to have impact on software engineering. The concept of reactive programming (cf. *http://www.reactivemanifesto.org/#the-need-to-go-reactive*) focuses on easy-to–arrange and -adapt micro services, in order to meet the original idea of real-time adaptation of software (component). In that context, subjects and their fundamental interaction scheme could play a crucial enabling role.

References

[1] R. Turner, “Tackling the DDoS threat to banking in 2014,” in White Paper of Alamai, Ovum Consulting, London, UK, 2014. (download 21.6.2015)

[2] Pymnts, DDOS attacks cost banks up to $100k per hour [Online]. (download 21.6.2015) Available: http://www.pymnts.com/news/2015/ddos-attacks-cost-banks-up-to-100k-per-hour/#.VYPOeEaAWPU.

[3] The Guardian. (2013). Google reveals sharp rise in requests for removal of political content [Online]. (download 21.6.2015) Available: http://www.theguardian.com/technology/2013/dec/19/google-transparency-report-political-content

[4] S. Lall, R. Cardarelli, and I. Tytell. (2006). How do financial system affect business cycles, in: World Economic Outlook. Financial Systems and Economic Cycles, International Monetary Fund, Washington DC [Online]. (download 21.06.2015) Available: https://www.imf.org/external/pubs/ft/weo/2006/02/pdf/c4.pdf.

[5] H. Smith and P. Fingar, “Business process management: the third wave,” vol. 1, Tampa: Meghan-Kiffer Press, 2003.

[6] Y. Chen, Y. Wang, S. Nevo, J. Jin, L. Wang, and W. S. Chow, “IT capability and organizational performance: the roles of business process agility and environmental factors,” in European Journal of Information Systems, 23(3), pp. 326-342, 2014. http://dx.doi.org/10.1057/ejis.2013.4

[7] S. Kemsley, “Business process management and the social enterprise,” in Handbook on Business Process Management 1. Springer Berlin Heidelberg, 2015, pp. 463-474. http://dx.doi.org/10.1007/978-3-642-45100-3_20

[8] R. Seethamraju and D. K. Sundar, ”Influence of ERP systems on business process agility,” IIMB Management Review, 25(3), pp. 137-149, 2013. http://dx.doi.org/10.1016/j.iimb.2013.05.001

[9] A. Fleischmann, W. Schmidt, Ch. Stary, St. Obermeier, and E. Börger, “Subject-oriented Business Process Management,” Springer, Open Book, Berlin, 2012a. http://dx.doi.org/10.1007/978-3-642-32392-8_

[10] A. Fleischmann and Ch. Stary, “Whom to talk to? A stakeholder perspective on business process development,” in Universal Access in the Information Society, 11(2), Springer, Berlin, 2012b, pp. 125-150. http://dx.doi.org/10.1007/s10209-011-0236-x

[11] S. A. Hajimirsadeghi, H. Y. Paik, and J. Shepherd, “Social-network-based personal processes,” in Asia Pacific Business Process Management, Springer International Publishing, 2015, pp. 155-169. http://dx.doi.org/10.1007/978-3-319-19509-4_12

[12] S. Aghaee and C. Pautasso, “End-user programming for web mashups,” in Current trends in web engineering. Springer, Berlin, 2012, pp. 347-351. http://dx.doi.org/10.1007/978-3-642-27997-3_38

[13] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, B. Myers, M.-B. Rosson, G. Rothermel, M. Shaw, and S. Wiedenbeck, “The state of the art in end-user software engineering,” ACM Computing Surveys (CSUR), 43(3), 21, 2011. http://dx.doi.org/10.1145/1922649.1922658

[14] B. Shneiderman, “Leonardo’s Laptop: Human Needs and the New Computing Technologies,” MIT Press, Cambridge, MA, 2002.

[15] L. G. Terveen, P. G. Selfridge, and M. D. Long, “Living design memory: framework, implementation, lessons learned,” Human-Computer Interaction, 10(1), pp. 1-37, 1995. http://dx.doi.org/10.1207/s15327051hci1001_1

[16] M. A. Lyles, “Organizational Learning, knowledge creation, problem formulation and innovation in messy problems,” European Management Journal, 32(1), pp. 132-136, 2014. http://dx.doi.org/10.1016/j.emj.2013.05.003

[17] A. Rugman, “The end of globalization,” Random House, NY, 2012.

[18] A. Giddens, “The consequences of modernity,” in Choice Reviews Online, Stanford University Press, Stanford, CA, 1990. http://dx.doi.org/10.5860/choice.28-1843

[19] St. Roth, “The Fairly Good Economy: Testing the Economization of Society Hypothesis against a Google Ngram View of Trends in Functional Differentiation (1800-2000),” in The Journal of Applied Business Research, 29(5), pp. 1495-1500, 2013.

[20] A. Smith and A. Stirling. (2010). “The politics of socio-ecological resilience and sustainable socio-technical transitions,” in Ecology and Society, 15(1), 11 [Online]. (download 21.6.2015). Available: http://www.ecologyandsociety.org/vol15/iss1/art11/.

[21] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” in Communications of the ACM, 53(4), pp. 50-58, 2010. http://dx.doi.org/10.1145/1721654.1721672

[22] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, “Cloud computing – the business perspective,” Decision Support Systems, 51(1), pp.176-189, 2011. http://dx.doi.org/10.1016/j.dss.2010.12.006

[23] L. Kappelman, E. McLean, J. Luftman, and V. Johnson, “Key issues of IT organizations and their leadership: The 2013 SIM IT trends study,” MIS Quarterly Executive, 12(4), pp. 227-240, 2013.

[24] S. Kolb and G. Wirtz. “Towards application portability in platform as a service,” in IEEE 8th International Symposium on Service Oriented System Engineering (SOSE), 2014, pp. 218-229. http://dx.doi.org/10.1109/sose.2014.26

[25] R. Krebs, M. Loesch, and S. Kounev. “Platform-as-a-service architecture for performance isolated multi-tenant applications,” in IEEE 7th International Conference on Cloud Computing (CLOUD), June 2014, pp. 914-921. http://dx.doi.org/10.1109/cloud.2014.125

[26] D. Geer, “Industry trends: Chip makers turn to multicore processors,” in Computer, 38(5), pp. 11–13, 2005. http://dx.doi.org/10.1109/mc.2005.160

[27] J. Reinders, “Intel threading building blocks: outfitting C++ for multi-core processor parallelism,” O'Reilly, San Francisco, 2010.

[28] WIN-CE, UeberFlow, 2015 [Online]. (download 21.6.2015) Available: http://www.i2pm.net/interest-groups/ueber-flow/home

[29] M. Weske, “Business process management: concepts, languages, architectures,” Springer, Berlin, 2012. http://dx.doi.org/10.1007/978-3-642-28616-2

[30] G. Rimassa and B. Burmeister, “Achieving business process agility in engineering change management with agent technology,” in WOA, 2007, pp. 1-7.

[31] J. Dietz, “Enterprise Ontology: Theory and Methodology,” Springer, New York, 2006.

[32] C. Le Clair, C. Moore, A. Peters, D. Miers, and E. Fowler-Cornfeld, “Stuck in Cement: When Packaged Apps Create Barriers to Innovation,” Forrester Research, Cambridge, MA, 2012.

[33] E. Mturi and P. Johannesson, “A context-based process semantic annotation model for a process model repository,” in Business Process Management Journal, Emerald, 19(3), pp. 404-430, 2013. http://dx.doi.org/10.1108/14637151311319888

[34] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, and K.Yelick, “A view of the parallel computing landscape,” in Communications of the ACM, 52(10), pp. 56-67, 2009. http://dx.doi.org/10.1145/1721654.1721672

[35] F. Barachini and Ch. Stary, “Better Balanced Economies — A management challenge for the 3rd millennium,” in Journal of Information & Knowledge Management, 11(4), 2012. http://dx.doi.org/10.1142/s0219649212500256

[36] N. Holmberg, “The purity of separation of concerns: The service oriented business process-a design approach for business agility,” Ph.D. dissertation, Lund Univ., Lund, SE, 2014.

[37] J. Claes, I. Vanderfeesten, H. A. Reijers, J. Pinggera, M. Weidlich, St. Zugal, D. Fahland, B. Weber, J. Mendling, and G. Poels, “Tying process model quality to the modeling process: the impact of structuring, movement, and speed,” in Business Process Management, Springer, Berlin, pp. 33-48, 2012. http://dx.doi.org/10.1007/978-3-642-32885-5_3

[38] J. Frishammar, U. Lichtenthaler, and A. Richtnér, “Managing process development: key issues and dimensions in the front end,” in R&D Management, 43(3), pp. 213-226, 2013. http://dx.doi.org/10.1111/radm.12011

[39] K. Bergener, J. vom Brocke, S. Hofmann, A. Stein, and C. vom Brocke, “On the importance of agile communication skills in BPM education: Design principles for international seminars,” in Knowledge Management & E-Learning: An International Journal (KM&EL), 4(4), pp. 415-434, 2013.

[40] M. Laguna and J. Marklund, “Business process modeling, simulation and design,” CRC Press, 2013.

[41] C. Schütz, M. Schrefl and L. M. Delcambre. “Multilevel business process modeling: motivation, approach, design issues, and applications,” in Proc. the 5th Ph.D. workshop on Information and knowledge, ACM, 2013, 2012, pp. 91-94. http://dx.doi.org/10.1145/2389686.2389705

[42] L. Kutvonen, “ODP RM reflections on open service ecosystems,” in Computer Standards & Interfaces, 35(3), pp. 294-312, 2013. http://dx.doi.org/10.1016/j.csi.2012.01.005

[43] I. Nonaka and H. Takeuchi, “The Knowledge-Creating Company,” Oxford University Press, NY, 1995.

[44] V. Allee, “Value network analysis and value conversion of tangible and intangible assets,” in Journal of Intellectual Capital, 9(1), pp. 5-24, 2008. http://dx.doi.org/10.1108/14691930810845777

[45] R. S. Kaplan, “Strategy maps: Converting intangible assets into tangible outcomes,” Harvard Business Press, 2004. http://dx.doi.org/10.5465/ame.2004.13835877

[46] Ch. Stary, “Non-disruptive knowledge & business processing in knowledge life cycles – aligning value network analysis to process management,” in Journal of Knowledge Management. 18(4), pp. 651-686, 2014. http://dx.doi.org/10.1108/JKM-10-2013-0377

[47] St. C. Smith and J. Rothbaum, “Cooperatives in a global economy: Key economic issues, recent trends, and potential for improvement,” IZA Policy Paper No. 68, Institute for the Study of Labor, Bonn, DE, 26p, 2013.

[48] P. A. Buhler and J. M. Vidal, “Towards adaptive workflow enactment using multi-agent systems,” in Information technology and management, 6(1), pp. 61-87, 2005. http://dx.doi.org/10.1007/s10799-004-7775-2

[49] C. Ayora, V. Torres, M. Reichert, B. Weber, and V. Pelechano. “Towards run-time flexibility for process families: open issues and research challenges,” in Proc. BPM’12 Business Process Management Workshops, Springer, Berlin, 2013, pp. 477-488. http://dx.doi.org/10.1007/978-3-642-36285-9_49

[50] X. Xu, “From cloud computing to cloud manufacturing,” in Robotics and Computer-Integrated Manufacturing, 28, pp. 75-86, 2012. http://dx.doi.org/10.1016/j.rcim.2011.07.002

[51] V. Gramoli and R. Guerraoui, “Democratizing transactional programming,” in Communications of the ACM, 57(1), pp. 78-85, 2014. http://dx.doi.org/10.1145/2541883.2541900

[52] G. Urrego-Giraldo and G. L. Giraldo, “A process model representation for supporting concurrent engineering,” in Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment, J. Stjepandic, G. Rock, C. Bil, Eds., Springer, London, 2013, pp. 399-410. http://dx.doi.org/10.1007/978-1-4471-4426-7_35

[53] A. Fleischmann, U. Kannengiesser, W. Schmidt, and Ch. Stary. “Subject-oriented modeling and execution of multi-agent business processes,” in Proc. Int. Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT’13, ACM/WIC/IEEE, 2013a. http://dx.doi.org/10.1109/wi-iat.2013.102

[54] A. Fleischmann, W. Schmidt, and C. Stary. (2015). “S-BPM in the Wild,” in Practical Value Creation, Springer Open, Berlin [Online]. (download 21.6.2015) Available: http://link.springer.com/book/10.1007/978-3-319-17542-3. http://dx.doi.org/10.1007/978-3-319-17542-3

[55] J. Koehler, J. Hofstetter, and R. Woodtly, “Capabilities and levels of maturity in IT-based case management,” in Business Process Management, Springer, Berlin, 2012, pp. 49-64. http://dx.doi.org/10.1007/978-3-642-32885-5_4

[56] A. Fleischmann, W. Schmidt, C. Stary, and F. Strecker. “Nondeterministic events in business processes,” in Business Process Management Workshops, Springer Berlin, 2013b, pp. 364-377. http://dx.doi.org/10.1007/978-3-642-36285-9_40

APPENDIX

SUBJECT-ORIENTED SPECIFICATION (according to [53])

(1) A subject represents the behavior of an active entity.
(1.1) A specification of a subject does not imply any actor or technology that could be used to execute the described behavior.
(1.2) Subjects communicate with each other by exchanging messages.

(2) A Message has a name and a payload.
(2.1) The name should express the meaning of a message informally.
(2.2) The payloads are the data (business objects) transported.

Subjects have internal behavior representations.
(3) Internally,
(3.1) a subject executes local activities
(3.2) a subject sends messages to other subjects
(3.3) a subject expects messages from other subjects
(3.4) a subject performs all these activities in sequences – they are defined in a subject's behavior specification.

(4) Subject-oriented process specifications are embedded in some context.
(4.1) Context is defined by the business organization and environment it is part of.
(4.2) Context is provided by the technological infrastructure by which a business process can be executed.

The construction of a subject-oriented business representation is based on the behavioral entities or abstract resources involved in a business-relevant process termed subjects, and their exchanges of messages (interactions).

(5) Subject-oriented modeling requires several activities, namely, the specification of
(5.1) the business case
(5.2) the subjects involved in a process
(5.3) interactions they are part of
(5.4) the messages they send or receive through each interaction
(5.5) the behavior of each subject encapsulating functions and interactions

(6) Subject Interaction Diagram (SID)
(6.1) (5.2) to (5.4) constitutes a Subject Interaction Diagram.
(6.2) A Subject Interaction Diagram is the most abstract diagrammatic level of describing processes in S-BPM.
(6.3) For each subject of a Subject Interaction Diagram a Subject Behavior Diagram (7) needs to be constructed for a complete and coherent S-BPM model.

(7) Subject Behavior Diagram (SBD)
(7.1) (5.5) for each subject constitutes a Subject Behavior Diagram.
(7.2) After the step (6), the behavior of each subject is defined.
(7.3) A subject's behavior is described by three states (send, receive, internal function) and transitions between these states. Hence, when specifying the behavior of each subject, a sequence of sending and receiving messages, and activities to be set for task accomplishment need to be represented.
(7.4) The description of a subject defines the sequence of sending and receiving messages, or the processing of internal functions, respectively. In this way, a subject specification contains the sequence of predicates.
(7.5) A Subject Behavior Diagram is the most concrete diagrammatic level of describing processes in S-BPM.
(7.6) Each Subject Behavior Diagram should detail a specific subject of a Subject Interaction Diagram (6) for a complete and coherent S-BPM model.

(8) The states of a Subject Behavior Diagram represent operations.
(8.1) They are active elements of the subject description.
(8.2) States are implemented by services.
(8.3) State transitions are necessary to exchange and manipulate business objects.

(9) (Business) Objects are
(9.1) data and/or applications affected by operations of a subject
(9.2) data and/or applications processed through services.