

Pattern-Based Engineering of Systems-of-Systems for Process Execution Support

Albert Fleischmann¹, Werner Schmidt^{2(✉)}, Christian Stary³,
and Christian Fichtenbauer⁴

¹ Interaktiv Expert, Pfaffenhofen, Germany

Albert.fleischmann@interaktiv.expert

² Technische Hochschule Ingolstadt, Ingolstadt, Germany

Werner.schmidt@thi.de

³ Johannes Kepler University, Linz, Austria

Christian.stary@jku.at

⁴ C-FC, Mortsel, Belgium

Fichtenbauer@icloud.com

Abstract. Globalization and digital transformation increasingly drives agility requirements of business processes, hand-in-hand with execution support by highly flexible, interactive and distributed socio-technical systems. The design of such systems spans the conceptual level of business process models and the implementation level with regard to the organizational environment (assignment of tasks to units/people) and the IT environment (assignment of tasks to applications or services). The organizational and IT environment are often quite heterogeneous, particularly in large, globally operating organizations. The latter may also face specific regulations in different countries, which impacts the way of doing business there. As a consequence, multiple business process variants are developed, both with respect to the organization of work, and IT support. However, such a business reality limits the desirable homogeneity of solutions. It rather requires design concepts allowing for technical and organizational flexibility while keeping mutual coherence. We present ideas for a pattern-based concept ensuring both. It combines System of Systems (SoS) thinking and Subject oriented BPM (S-BPM).

Keywords: System of systems · BPM · S-BPM · Process variants

1 Introduction

Globally operating enterprises are doing business in many countries. While following the same objective, like selling cars, operations at different locations usually vary depending on the organizational environment, the IT support for accomplishing tasks, and other factors, e.g., legal regulations.

Business Process Management (BPM) is a widely accepted approach to align different operational aspects. It comprises several activities like process discovery, design, implementation and execution. One way to capture the different business realities is to identify process variants embedded in a specific target environment.

Thereby, tasks specified in a process model are assigned to units/members of the organization and their IT support is realized by integrating existing with newly implemented applications and services.

The result is a set of Locally Executable Business Processes (LEBP), being considered as combinations of work procedure descriptions (process models) and their embedding into the local organization and IT infrastructure.

This perspective in many ways refers to the notion of System of Systems (SoS) as ‘a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities’ [16]. As systems of systems in the mentioned context can be seen:

- the overall structural organization as the sum of all local, subsidiary parts of the organization, in the end representing the human resources (e.g., departments, teams, places, people)
- the overall process organization as the sum of (partial) process models
- the overall IT infrastructure as the sum of all distributed IT resources, which can be integrated in order to support operation (e.g., workflow engines, enterprise resource planning systems, enterprise content management systems, enterprise service bus etc.)
- the overall operational business system of the enterprise as the sum of all LEBPs, representing the enterprise architecture. Each LEBP itself can also be seen as an SoS, as a whole providing a particular solution for running the respective business process.

The major challenge for practical use is to effectively and efficiently design and maintain those systems of systems. Here some of the characteristics of SoS come into play as there are operational and managerial independence of elements, evolutionary development, emergent behavior and geographic distribution. The exchange of messages and the agreements upon their content facilitate the interconnection and interoperability of the SoS parts. They are a prerequisite for the contribution of the single, autonomous systems to the achievement of higher-level goals, like jointly accomplishing business process instances, as well as for emergent behavior of the SoS. The independence of the single systems and their loose coupling to a SoS allow changing, replacing and adding pieces without affecting other parts. This is exactly what agile behavior of an enterprise requires.

Following the System of Systems notion we present a pattern-based process framework for specifying building blocks of highly flexible yet consistent and coherent process execution support systems. This is of particular interest not only for engineering internal but also cross-enterprise systems-of-systems, e.g., along supply chains or networks. The framework is based on the subject-oriented approach to Business Process Management (S-BPM), which puts the communication of active entities and their individual behavior in a business process in the center of interest. This perspective on work procedure level finds its analogy on IT level with the concept of communicating micro-services, which, however, is not part of this paper.

The remainder of the article is structured as follows: After this introduction we look at related work. In Sect. 3 we present our concept for a Subject-oriented Process System of Systems, before we conclude in Sect. 4.

2 Related Work

Subject-Oriented Business Process Management (S-BPM). S-BPM is a communication-oriented paradigm focusing on subjects as acting parties in business processes (people, systems) [7]. These parties are loosely coupled and synchronize their work on the common goal of executing process instances by exchanging messages. Each subject has a behavior defining the work procedure to accomplish its contribution to the process result. Subject representatives can model subject behaviors independently as long as the messages (interfaces) remain untouched or are agreed upon newly. This behavior encapsulation allows highly flexible self-organization, with the communication structure assuring overall coordination. It also facilitates exchanging behavior descriptions without affecting other parts of the process model. On the implementation level the notion of subjects decouples their occurrence in a process model from people or systems executing their behavior at runtime. This feature fosters implementation and execution flexibility, e.g., identical process activities can be performed by different organizational units and supported by different IT solutions at various locations (e.g., headquarter and subsidiaries).

Systems of Systems. [3] in their recent report introduce ‘platform’ to abstract from concrete elements, such as vehicles, in a universe of discourse depending on a certain context. A ‘platform architecture’, consequently, ‘may reflect the separation of payload from platform’ (p. 3). Extending the traditional notion of industry platform they identify a SoS platform as provider of ‘common information models (semantics) and common communication mechanism (ibid.)’. It may also prescribe patterns or sequences of interaction for certain SoS functions’ (ibid.). In analogy to that understanding, S-BPM as a SoS platform supports interoperation among systems, as its subject diagrams scope business-relevant information models that can be shared and mutually interact based on a common communication mechanism, namely exchanging messages.

With respect to designing an SoS architecture from scratch [3] identified top-down decomposition as one way to accomplish interoperability. Their empirical findings indicate that developers recognize the claim for architectural ‘purity’ which cannot be achieved in a straightforward way. As a consequence not all systems of SoS are changeable. In those cases, rather than rethinking modularity based on a common communication mechanism, only interfaces can be managed. From DoD’s SoS experience, the developers concluded the focus of design should be on runtime with an integrator for the infrastructure and the constituent systems. They warn from the effort to integrate differently architected systems for autonomous operations to collaborate without a common information exchange pattern (p. 15f.)

[8] recognized decomposition and bottom-up construction to be complementary along spiral development of SoS, as it reflects the dynamics of accomplishing change. In addition, constituent systems of SoS should be self-aware, at least on the goal level, in order to meet challenges on a level of single systems (cf. [2]). Explicit representation of goals could help to adjust a system in operation during build time in the sense of identifying a ‘best fit’ for each component handling a global task before actually executing a SoS. However, such a perspective needs revisiting Requirements

Engineering and the embodiment of fundamental SoS categories, such as directness, acknowledged objectives, collaboration, and virtuality with respect to control (cf. [14]).

Recognizing the resulting need of structured tool support [4] have developed COMPASS aiming for collaborative SoS development (see also [10]). In that context, they came up with a modeling language representing the particularities of SoS. However, they were focusing on abstract SoS modeling concepts rather than a particular execution logic controlling SoS behavior, as e.g., the hypergraph approach for constraint satisfaction [11].

The COMPASS language abstracts from SoS as a collection of constituent systems. They are specified by SoS behavior which is very likely to be incomplete (as emergence hinders capturing all cases). However, in line with S-BPM, each constituent system has

- to have a specification of its behavior relevant to the SoS
- to provide sufficient state information to allow the constituent system to be verified and executed
- to represent a consistent abstraction of the specification, however, its internal state need not be exactly the same as the actual system specification
- to have a specification, that is completely independent of the overall SoS. If it is not possible to create such a specification, then it may not be a system in its own right.

In order to capture operation of SoS, a SoS model should include an infrastructure specification, modelled as constituent systems. ‘These parts of the overall specification are used to either bind the component systems into the SoS, or to provide a bit of extra functionality beyond what is provided by any of the constituent systems. Full specifications of these constituent systems must be provided in detail to all members of the SoS’ [4, p. 452].

From a managerial perspective, SoS projects benefit from a layered approach to define dynamic behavior capabilities in the course of analysis, e.g., as provided by [19]. Their Base Level addresses networked actors in terms of human agents. They utilize information and resources to perform (project) activities. The Activity Level contains networked activities abstracted from the actions (including interaction) of the networked actors on the Base Level. The Process Level networks processes as aggregates of Activity Level’s activities. Finally, the Project Level captures all emerging properties from the network on the Process Level.

Modelling Process Variants. This topic has been investigated for quite a while. Approaches range from the so-called fragmented- or multi-model concept to the consolidated- or single-model concept [5, 9]. However, [13] state ‘Striking a trade-off between modelling each process variant separately versus collectively in a consolidated manner is still an open research question.’ We do not discuss this question in more detail, but rather resolve this issue pragmatically.

3 Subject-Oriented Process System of Systems

3.1 Drivers for Locally Executable Business Processes

The number of various Locally Executable Business Processes necessary to cover operation on all markets depends on the diversity of the respective environment. To identify LBEPs we suggest a three-dimensional model, inspired by [1, 6, 12]. It comprises the following, partially interdependent dimensions:

- **Market.** This dimension represents variation caused by the location where a process is executed, and its consequences. Factors like differing legal regulation or habits can limit standardization of business logic or let it seem not reasonable.
- **Organization.** Organization refers to the question of whether tasks of a process being operated in different markets are executed by similar organizational units or roles.
- **IT.** The IT dimension serves to differentiate applications and services in place to support processes at various locations.

As each dimension can comprise an arbitrary number of characteristics and therefore can be split differently, the resulting body is a cuboid. It can be partitioned repeatedly into sub-cuboids, all of which represent an LEBP, suitable for the particular combination (see Fig. 1). Process variants not necessarily differ in all dimensions, allowing for standardization with patterns wherever possible and reasonable. On the other hand, not all combinations may actually occur, meaning there can exist empty sub-elements. Nevertheless, the cuboid covers all enterprise operations.

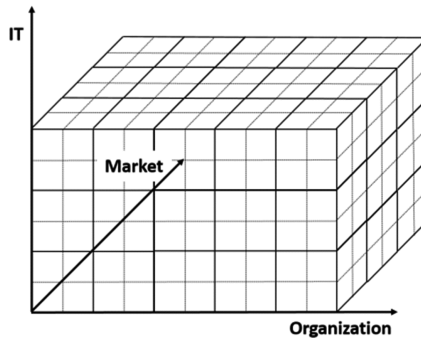
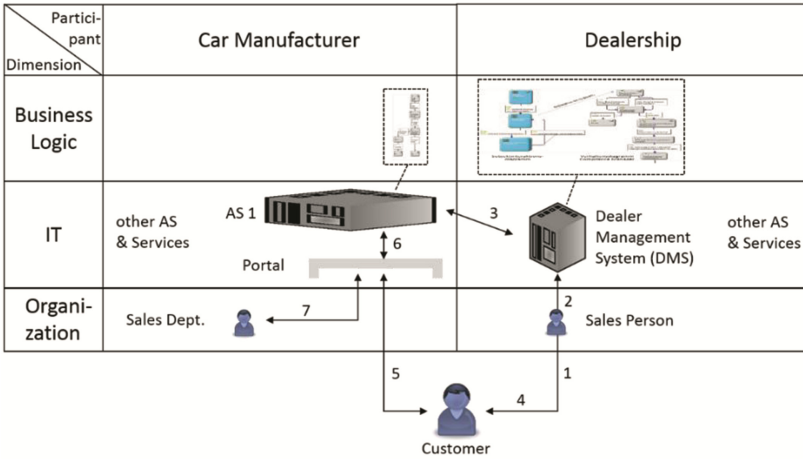


Fig. 1. Dimensions of LBEPs

An example from the automotive industry shows how IT infrastructure causes variants and LBEPs. The following scenario is about arranging an appointment for car maintenance in a garage, which is part of the overall car service process. The way the process works for the garage, belonging to a dealership, and the car manufacturer, mainly depends on whether the dealership has an own Dealer Management System (DMS) or not. Figure 2 depicts the resulting variants with the effect on the distribution of the business logic.

Variant 1: Dealership has Dealer Management System



Variant 2: Dealership does not have Dealer Management System

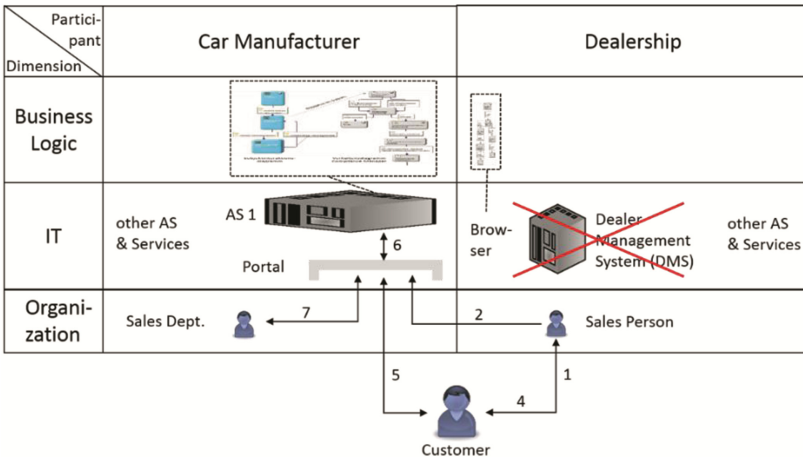


Fig. 2. Process variants

In variant 1 the customer articulates an appointment request to the dealer (1). There, an employee there uses the DMS to manage customer data, dates etc. (2), to connect to an application system AS 1 of the car manufacturer (3), to synchronize data, and to get access credentials to be passed on to the customer (4). Then, the customer can access this system via a portal (5, 6). Data in AS 1 is also maintained by the sales department on the manufacturer’s side (7, 6). In this variant main parts of the business logic is represented in the DMS, only a few steps for handling the data access are executed in AS1.

In case of variant 2 the customer either directly accesses AS1 through the portal (if registered before) (5, 6), or the sales person at the Dealer’s does that by proxy, using a browser (1, 2, 4). The overall business logic is different, the processes on both sides need to be designed accordingly.

3.2 Concept of the Process Framework

To cope with the sketched variety, the objective is to model process fragments, i.e. subsystems, ensuring their maximal independence and interoperability, as well as constructing largely generic patterns where these systems can be plugged in. As mentioned in the previous sections the System of Systems philosophy and the S-BPM approach as an SoS platform seem to provide conceptual ground for a process framework meeting these requirements.

S-BPM considers humans and systems as subjects, having an encapsulated behavior and exchanging messages in order to coordinate their activities. Systems in this context can be processes, IT applications or services, robots etc. This notion allows building SoS solutions for process execution support, both decomposing top-down and constructing bottom-up as suggested by [3, 8], in any case consisting of elements loosely coupled by message exchange in order to accomplish interoperability.

While decomposition splits processes from the business perspective horizontally and vertically (hierarchically) into pieces or phases linked by messages, construction based on behavior of distinct systems lets the communication structure evolve, when they need to be linked in order to accomplish a (sub) process.

Any of the dimensions in Sect. 3.1 can give evidence for how to crop process parts during initial design and when changes occur over time. They also determine whether independent building blocks need to be modified, replaced, added etc.

Figure 3 depicts the frame of the already mentioned car service process, being decomposed in subsystems representing phases of the overall process. Subject orientation considers a business process to be a system of communicating entities. They produce results that add to accomplishing the task of the overall process system. Following this notion, each subsystem contains the involved subjects, their interaction and their individual behavior. Messages between subjects in different phases connect the parts loosely, however, deliver results. Although linear with a back loop in the example, the communication pattern can be networked. The figure also shows three variants of the ‘Arrange Appointment’ phase, each of which can be plugged in the frame depending on their fit to the concrete situation. The only constraint is to assure compatibility or adaptation of the connecting messages.

The coupling by messages facilitates easy variation both horizontally regarding the phases, and vertically over multiple levels of detail. This allows situative design of LBEPs, where business and system architects can select or develop the variant fitting to each phase or, within phases, the relevant subjects (i.e., their encapsulated behavior). This is not only valid for initial design, but also for maintenance.

Architects can create patterns with identical building blocks for situations with similar characteristics of the cuboid dimensions, and add or replace only those with deviating facts. In our sample, variant 1 of ‘Arrange Appointment’ can be applied as a pattern for all cases (countries, markets, etc.) where the dealership has a DMS, assuming there are no differences with respect to the other dimensions. In cases without DMS, the same process frame (pattern) is used, but with variant 2 another pattern is plugged in, specifying different subjects, communication structure and behavior (see Fig. 3).

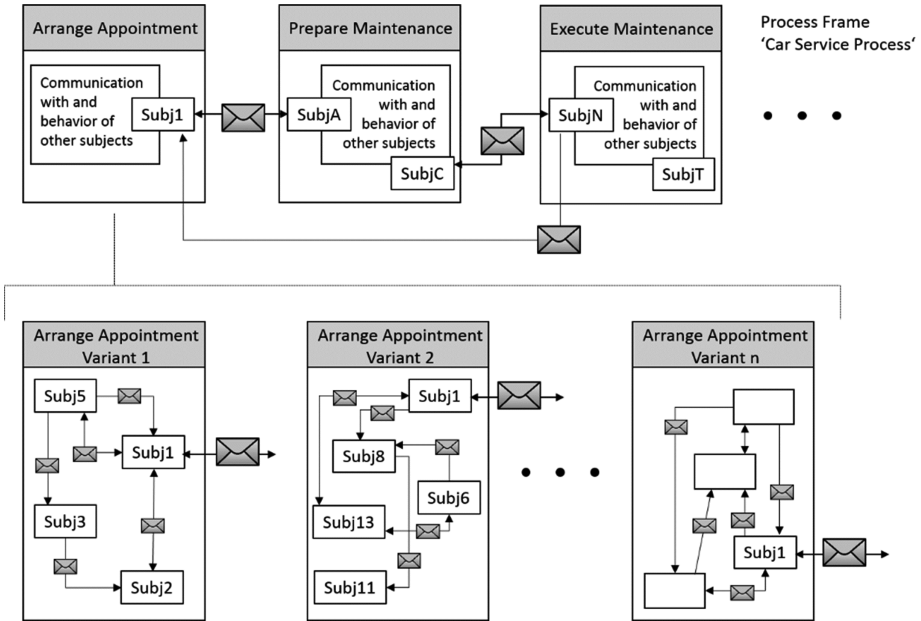


Fig. 3. Process frame and variants

Case-specific design and plugging building blocks into the frame on demand is a compromise of existing ways to generate process variants, as there are behavior-based and structural configurations [17]. The first starts with a comprehensive model covering all known aspects. Subsequently, variants are derived by hiding and blocking model parts. Structural configuration uses a base model and pre-specified changes (e.g., insert, delete, move activities) to be applied at adjustment points. In case of many variants, both approaches may cause extensive effort.

Major benefits from our concept can be expected w.r.t. reducing complexity while increasing flexibility, due to encapsulating and loosely coupling of building blocks. Replacing elements is possible with minimal effect on other parts, either at design time or in the course of deployment. Such straightforward reconfiguration facilitates emergence of new behavior.

4 Conclusion and Further Research

We combined the SoS philosophy and the S-BPM approach to a concept for engineering comprehensive yet flexible process execution support systems. The concept particularly aims for globally acting enterprises requiring many different Locally Executable Business Processes.

The concept is in an early stage of development. It is currently tested, revised and detailed in a practical setting in industry. Revisions will undergo evaluation efforts in future. For instance, detailing processes refers to IT implementation in terms of LBEPs

using micro-services. This is a logical consequence as the presented concept is their counterpart on the level of business process models. Micro-services as small, decoupled pieces of software, implemented in different programming languages, and representing certain encapsulated capabilities correspond to the representatives of the IT dimension of the cuboid in Fig. 1 [15]. By communicating via language-independent interfaces, they allow building complex applications following a modular LBEP architecture whose elements can easily be replaced. Strategic alignment of business process variants (e.g., see [18]) is another topic to be investigated to see how the presented approach can be related to business strategies.

References

1. Abbott, M.L., Fischer, M.T.: *The Art of Scalability*. Addison-Wesley, Boston (2015)
2. Cavalcante, E., Batista, T., Bencomo, N., Sawyer, P.: Revisiting goal-oriented models for self-aware systems-of-systems. In: 2015 International Conference on Autonomic Computing (ICAC), pp. 231–234. IEEE Press, New York (2015)
3. Cohen, S., Klein, J.: State of practice report: essential technical and nontechnical issues related to designing SoS platform architectures. Technical report CMU/SEI-2015-TR-007 (2015)
4. Coleman, J.W., Malmos, A.K., Larsen, P.G., Peleska, J., et al.: COMPASS tool vision for a system of systems collaborative development environment. In: 7th International Conference on System of Systems Engineering (SoSE), pp. 451–456. IEEE Press, New York (2012)
5. Dumas, M.: Consolidated management of business process variants. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM Workshops 2011, Part II*. LNBP, vol. 100, p. 1. Springer, Heidelberg (2012)
6. Fichtenbauer, C., Fleischmann, A.: Three dimensions of process models regarding their execution. In: *Proceedings of S-BPM ONE 2016*. ACM DL (in print)
7. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: *Subject-Oriented Business Process Management*. Springer, Heidelberg (2012)
8. Gomez, A., Fonck, B., Ayoun, A., Inzerillo, G.: Concurrent system engineering in air traffic management: steering the SESAR program. In: *Proceedings of the Posters Workshop at CSD&M*, p. 25 (2013). <http://ceur-ws.org/Vol-1085/04-paper.pdf>
9. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and management of process variants. In: Brocke, J., Rosemann, M. (eds.) *Handbook of Business Process Management*, pp. 237–255. Springer, Heidelberg (2010)
10. Hallerstede, S., Hansen, F.O., Holt, J., Lauritsen, R., Lorenzen, L., Peleska, J.: Technical challenges of SoS requirements engineering. In: 7th International Conference on System of Systems Engineering (SoSE), pp. 573–578. IEEE Press, New York (2012)
11. Khalil, W., Koubeissi, A., Merzouki, R., Conrard, B., Ould-Bouamama, B.: Contribution to system of systems modeling. In: 10th System of Systems Engineering Conference (SoSE), pp. 182–186. IEEE Press, New York (2015)
12. Milani, F., Dumas, M., Matulevičius, R.: Identifying and classifying variations in business processes. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) *EMMSAD 2012 and BPMDS 2012*. LNBP, vol. 113, pp. 136–150. Springer, Heidelberg (2012)
13. Milani, F., Dumas, M., Matulevičius, R.: Decomposition driven consolidation of process models. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) *CAiSE 2013*. LNCS, vol. 7908, pp. 193–207. Springer, Heidelberg (2013)

14. Ncube, C., Lim, S.L., Dogan, H.: Identifying top challenges for international research on requirements engineering for systems of systems engineering. In: 21st International Requirements Engineering Conference (RE), pp. 342–344. IEEE Press, New York (2013)
15. Newman, S.: Building Microservices. O'Reilly, Sebastopol (2015)
16. Office of the Deputy Assistant Secerary of Defense: SoS Systems Engineering. http://www.acq.osd.mil/se/initiatives/init_sos-se.html. Accessed 9 Feb 2016
17. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems. Springer, Heidelberg (2012)
18. Zellner, P., Laumann, M., Appenfeller, W.: Towards managing business process variants within organizations - an action research study. In: Proceedings of the 2015 48th Hawaii International Conference on System Sciences, pp. 4130–4139. IEEE Press, New York (2015)
19. Zhu, J., Mostafavi, A.: Towards a new paradigm for management of complex engineering projects: a system-of-systems framework. In: 8th Annual IEEE Systems Conference (SysCon), pp. 213–219. IEEE Press, New York (2014)