Requirements Elicitation and Specification using the S-BPM Paradigm

Christian Stary

Johannes Kepler University, Linz, Austria
Christian.stary@jku.at

Abstract. Recognizing the essential role of eliciting stakeholder needs and expectations when specifying system requirements from an interaction perspective, corresponding research has been triggered in subject-oriented requirements engineering. Thereby, requirements elicitation from a functional and interactional perspective overcomes essential deficiencies in behavior-centered process and application development: the lack of structuring elicitation and modelling, and the lack of prescribed deliverables. In this paper, the rationale of the subject-oriented approach is detailed and its application is exemplified. The cases show how system requirements evolve along multi-dimensional elicitation and specification activities. The approach is effective at delivering semantically coherent, and consistent requirements specifications. A crucial benefit seems to be scaffolding, in particular, when communication between stakeholders and/or interaction between systems is considered in addition to a purely functional or task-centered perspective. Finally, the generated representations can be used throughout development, thus supporting continuous evolvement.

Keywords: communication requirements, stakeholder perception, interaction, messaging, behavior encapsulation, subject-orientation, systems engineering.

1 Introduction

Today, Requirements Engineering is understood as dynamic communication and interaction process between different stakeholders (cf. Dybå et al., 2013), in particular when taking into account existing expertise and different mental models on organizing work and technologies (cf. Rosenkranz et al., 2014). Requirements elicitation, analysis, and negotiation are considered crucial for non-disruptive system development, and a topic to be better understood, albeit various reference models and tool developments (cf. Sharma et al., 2014; Seyff et al., 2015). Stakeholders require more consideration, in particular supporting their capability to articulate their requirements at the beginning of a project (cf. Hess et al., 2013, Burnay et al., 2014). Incremental and early approaches to requirements elicitation engage stakeholders in detailing or narrowing design- and implementation-relevant information (Miller et al., 2014, Oppl, 2016, Schneider et al., 2013). Commitments along articulating situation-sensitive knowledge, structuring needs, and negotiating requirements help for later stage developments (Seyff et al., 2015, Vitharana et al. 2016).

Metaphors or paradigms can support elicitation, structuring, and specification, such as demonstrated recently for agent-oriented requirements engineering (Miller et al., 2014). Subject-orientation (Fleischmann et al., 2012a) has been introduced as a paradigm and IT-supported methodology in the field of Business Process Management (BPM) to enable stakeholders expressing their knowledge of work not only in terms of how they perform tasks according to their formal role and technical knowledge, but also in terms of their interactions, in particular when and with whom or which system they exchange business-relevant information (cf. Fleischmann et al., 2012b, 2013). However, as recent empirical studies (Fleischmann et al., 2105, Neubauer et al., 2017) reveal, cognitive bias at design time (cf. Razavian et al., 2016), including early focus on cognitive user interface issues (cf. Eberle et al., 2011) seem to hinder effective subject-oriented requirements elicitation and representation.

Although several subject-oriented modeling and engineering support tools have been proposed over the last years (cf. Fleischmann et al., 2014, Krenn et al., 2017), requirements elicitation and specification still lacks support acquiring and representing information considered relevant by stakeholders from an interaction/communication *and* task perspective. Methodological approaches do not describe in a systematic way how to address and check both perspectives in the course of elicitation and presentation in a balanced way. Thus, the acquisition of system-relevant requirements remains an arbitrary process, depending on the facilitator and concerned stakeholders at hand. For instance, eliciting knowledge focusing on a purely functional (task-specific) perspective means using the behavioral abstraction 'subject' as isolated task representation, e.g., for accounting. When the interaction with other stakeholders or other subjects (subjects may also represent technical systems) is not tackled, only a part of the subject-oriented paradigm is addressed.

In line with that, focusing on the modelling notations may put principles fundamental for the paradigm to the background. For instance, capturing only those data or business objects that are actual part of interactions between systems or stakeholders, is highly relevant for effective and efficient development of subject-oriented applications. Conveying such underlying principles to modeling and design adequately engages stakeholders, as they can develop explicit commitment to them. However, they need to be informed in the course of requirements elicitation and specification. Referring to principles of a paradigm rather than to notational or executable modeling elements promotes a design space capturing diverse behaviors.

In the following work an incremental approach to requirements elicitation and modelling is pursued to engage stakeholders, acknowledging that requirements elicitation is effectively supported with semi-structured methods, as it that narrows down choices when progressing. Instead of removing issues considered important or reflected immediately, stakeholders can keep it, until they feel confident how to proceed. The approach builds upon findings on visualizing requirements, e.g., through feature diagram hierarchies (Slavin et al., 2014). It makes use of experiences with scaffolding (cf. Oppl, 2016) and tangible support technologies when articulating and sharing knowledge (cf. Oppl et al., 2014). Section 2 gives a brief overview of the paradigm and subject-oriented models allowing for automated execution. Section 3 introduces the semi-structured support for identifying subject-oriented behavior encapsulations and checking them from a functional action and an interactional perspective. Section 4 concludes the paper summarizing the results and giving an outlook to future research.

2 Subject Orientation

Subject-oriented Business Process Management has established a modeling approach encapsulating communication-oriented behavior and refining it according to services to be processed for accomplishing tasks. These services either refer to directly performed actions (system functions or manually accomplished tasks), or to sending or receiving messages (including business objects). Hence, behavior is constituted by performing actions, sending messages (after preparing their content), and receiving messages (by analyzing their content). Validated models of this kind can be executed without further transformation.

When creating a representation in a subject-oriented way, and thus eliciting and specifying requirements, several aspects influence this process:

1. The baseline are entities that interact, either through exchanging messages or objects, such as data for task accomplishment. Hence, the list of requirements refers to interacting entities.
2. These interacting entities can be considered as systems from a generic perspective (cf, Stary, 2017), as they represent roles, persons, artefacts, app's, i.e. technical or social entities. From a stakeholder perspective, the scope and thus, the granularity is subjective. And this bias induces a second one: The scope and thus, granularity is very likely to depend on how a situation at hand is perceived.
3. Persons acting as modelers need to be aware they are observers, i.e. a part of the observed world that is represented in a subject-oriented model.
4. Elements of the observed reality are represented as web of interacting entities without representing the observer and its relations explicitly.

The rationale for defining subjects are entities (systems) carrying or taking intentional roles in processes leading to or requiring interaction(s) with other entities (task carriers of this kind), and being part of an organizational setting. A technical system is viewed as emerging from both the interaction between subjects and their specific behaviors encapsulated within the individual subjects. Like in reality, subjects (systems) operate in parallel and can exchange messages asynchronously or synchronously. It is a view of reality as autonomous, concurrent behaviors of distributed entities. A system (subject) is a behavioral role assumed by some 'actor', i.e. an entity that is capable of performing actions. The entity can be a human, a piece of software, a machine (e.g., a robot), a device (e.g., a sensor), or a combination of these. Subjects can execute local actions that do not involve interacting with other subjects (e.g., calculating a price and storing a postal address), and communicative actions that are concerned with exchanging messages between subjects, i.e. sending and receiving messages.

In diagrammatic S-BPM models, subjects are one of five core symbols used when specifying designs. Based on these symbols, two types of diagrams can be produced to conjointly represent a system: Subject Interaction Diagrams (SIDs) and Subject Behavior Diagrams (SBDs). SIDs provide a global view of an observed reality, comprising the subjects involved and the messages they exchange. The SID of a simple ordering process is shown in Figure 1. Subject Behavior Diagrams (SBDs) provide a local view of the process from the perspective of individual subjects. They include sequences of states representing local actions and communicative actions including sending messages and receiving messages. State transitions are represented as arrows, with labels indicating the outcome of the preceding state (see Figure 2 & 3).

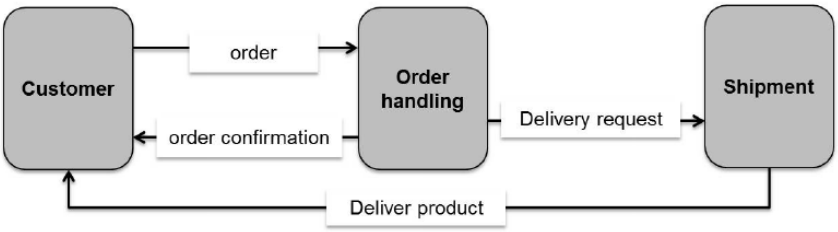

Figure 1. Order Handling – Subject Interaction Diagram

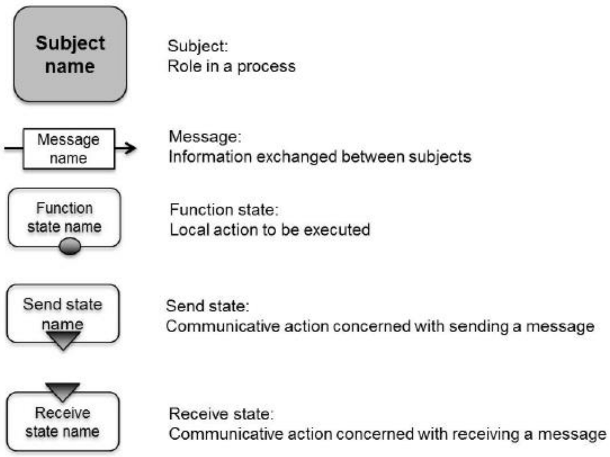

Figure 2. Diagrammatic elements in subject-oriented models

Figure 3 exemplifies a customer – order-handling department relationship through a role-specific interaction pattern, e.g., confirming an order to a customer, and actions to be performed on either side to prepare an order and to deliver a service or product according to the order. While in such an application case, subject-oriented requirements specification is straightforward due to existing actor or system roles, complex tasks, such as workforce planning, require specific preparation of stakeholders – see e.g., Stary (2014) for such an endeavor in healthcare.

As validated behavior specifications can be executed without further transformation, stakeholders can put their subject-oriented specifications to operation. This features is of importance for continuous requirements engineering, as seamless roundtrips in development are enabled - stakeholders may change specifications and run them.

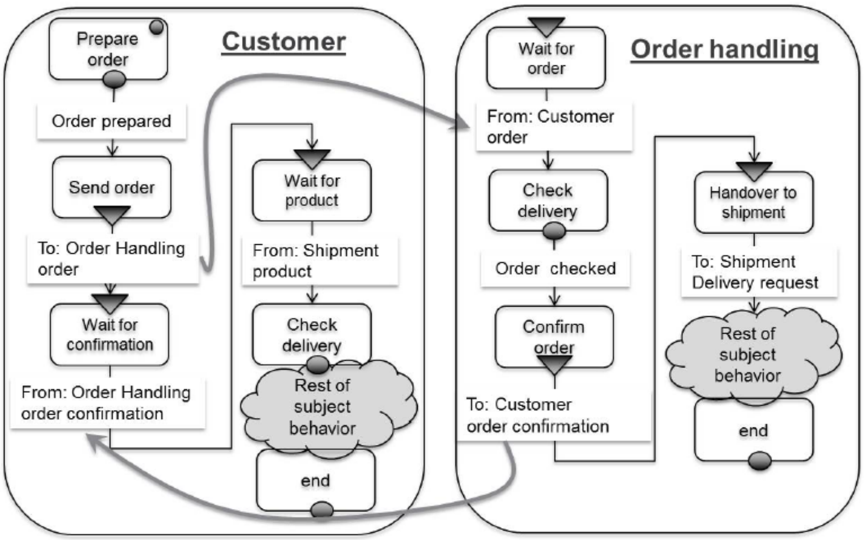

Figure 3. Order Handling – Subject Behavior Diagrams 'Customer' and 'Order Handling'

3 Multi-Perspective Requirements Engineering

In this section we provide the procedure and representation of the approach we have tested in the field, namely in service industries, in particular in consulting and banking. In line with our experiences in stakeholder elicitation and specification (Neubauer et al., 2017), this approach should help overcoming historically established patterns of eliciting and documenting, e.g., using Data Flow diagrams (cf. Gane et al., 1979), and thus, focusing on a certain perspective (in this case technical process aspects of an information system). Furthermore, the approach is designed to require minimal training for the facilitator, e.g., a system developer who would like to gather requirements from members of an organization.

3.1 Procedure

The proposed subject-oriented requirements elicitations and specification procedure distinguishes three groups of activities composed of development steps which require the participation of the facilitator, stakeholders (e.g., users) or both. Those activities and steps are requirements a system has to satisfy. The first activity contains *required preparation activities (1)*. The facilitator undertakes the following steps:

1.1 Set up a space for requirements elaboration: As a first step, the facilitator creates some space for elaboration (e.g., a topic structure to be discussed in a table-top articulation session) with an appropriate title for the project at hand. Project participants are invited.

1.2 Provide project portfolio: The facilitator structures the project space and puts in required start information, e.g., a project description, folders for articulation protocols. It serves as development repository, in order to ensure availability of documentations and traceability of the development process.

The subsequent set of activities concerns *requirements elicitation, specification, and negotiation activities (2)*. The following tasks are mainly performed by the participating stakeholders, guided by the facilitator.

2.1 Trigger a group meeting: The facilitator invites stakeholders to an elicitation meeting. Depending on the project, the facilitator may ask to acquire additional information and put it into the project portfolio.

2.2 Participants scope requirements: The facilitator invites stakeholders to begin specifying requirements according to a certain diagrammatic block structure to their perception of the current situation, their ideas, needs, and concerns.

2.3 Participants present their requirements: As soon as blocks have been specified, stakeholders can invite others and present them according to their specification. They can ask questions for clarification and issue concerns. The facilitator should guide the presentation through time budgeting and filtering/complementing various categories of information.

2.4 Participants acknowledge requirements: Participants provide their acknowledgement in this step for requirements. It can help to guide further developments.

The third set of activities leads to *requirements consolidation and refinement (3)*, and is performed by the group:

3.1 Check elements and relations: Stakeholder contributions may vary in encapsulating behaviors and providing complete information entries. However, they can be checked for functional and interaction relations, ensuring flow of information and control. The facilitator keeps track of the progress. A Subject Interaction Diagram (SID) can be constructed based on the encapsulated behavior specifications.

3.1 Detail required business objects: Stakeholders provide the attributes for each business (data) objects according to the interaction relations. Again, the facilitator keeps track of the progress.

For the design of a subject-oriented system, each subject of the SID needs to be detailed in terms of its internal functions and send/receive activities subsequently.

3.2 Requirement Bricks

Creating a development space is based on creating blocks representing service requirements. The fundamental element is the requirements block (Req-Brick). They focus in elicitation (cf. Sharma et al., 2014) and codify topics as they occur in conversation (cf. Burnay et al., 2014). In addition, it recognizes supportive haptic qualities of specification in interactive settings (Oppl et al., 2014). The latter has been recognized in educational settings so far (cf. Zuckerman et al., 2005), but not elaborated in the context of Requirements Engineering.

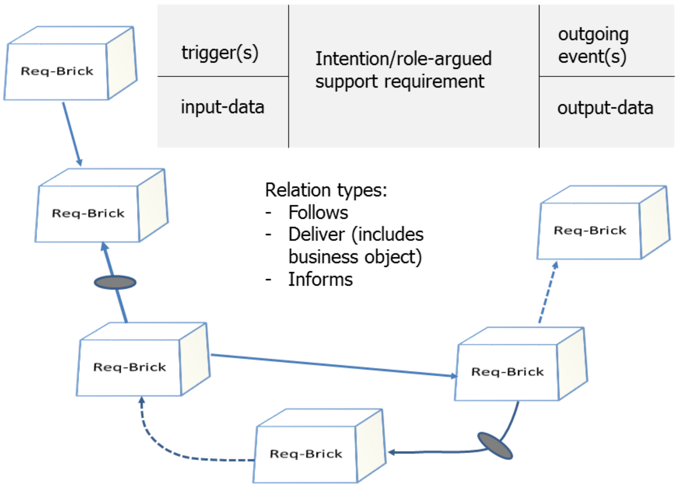

Figure 4. Requirement bricks: Inlay and relations

Each requirement block Req-Brick has a certain structure, as shown in Figure 4 and exemplified in Figure 6. It consists of

1. ***Incoming information***: Hereby we distinguish trigger from input (data) for a role-specific behavior requirement. Both help understanding a certain support requirement, while differentiating interaction from functional action.
2. ***Core Requirement***: It specifies the user story in terms of activities to be set, and the role in which an activity is intended to be performed. In this way, the context of actions can be represented.
3. ***Outgoing / delivered information***: Thereby, we distinguish outcome in terms of outgoing events from output data. This combination, like for incoming information, facilitates to distinguish communication interaction from resulting data from action.

The core requirement is based on a basic structure of a user story:

- TITLE as identifier of the user story
- Role identifier (AS A <role carrier>)
- Intentional action (I WANT TO <to set an activity>)
- Goal (IN ORDER TO <achieve a goal>)

The following user story addresses a sales requirement:

- TITEL: Sales innovation
- AS A Senior sales person
- I WANT TO have timely access to customer data (i.e. before meeting the customer) including the latest movements
- IN ORDER TO update my prepared offers

In addition, a validation criterion should be provided. It has the form

- ONCE I <set this activity> THE EFFECT IS <effect of activity >

For instance, the senior sales person could specify the following effect:

- ONCE I have timely access to customer data (i.e. before meeting the customer) including the latest movements THE EFFECT IS higher customer binding through better information

In this way, criteria for testing the successful implementation of a requirement can be provided. The core requirement is framed by the event(s) triggering the behavior and being caused when meeting the core requirement, as well as the data to be provided for meeting the requirement and becoming available when the requirement can be met. For instance, the trigger of the sales innovation is the set of scheduled events for customer contacts, induced by profile changes (i.e. input data). The output is a timely customer offer, enabling to approach customers with timely information.

The definition of Req-Bricks is supported by 3D manipulatives (Oppl et al., 2014) on a tabletop device. They represent requirements and can be related according to the categories shown in Figure 4. The process of specification is supported by documenting it along time stamps (see snap shot tool in Figure 5), set for recording the pattern on the table top. The manipulatives can be opened, serving then as containers of additional information. This information can be nested requirements, files, links, application functions, and is encoded using markers (shown in Figure 5). The eraser and circle device allow deleting relations and going back to previous Req-Brick constellations.

Figure 5. Tool support through table-top elements representing Req-Bricks

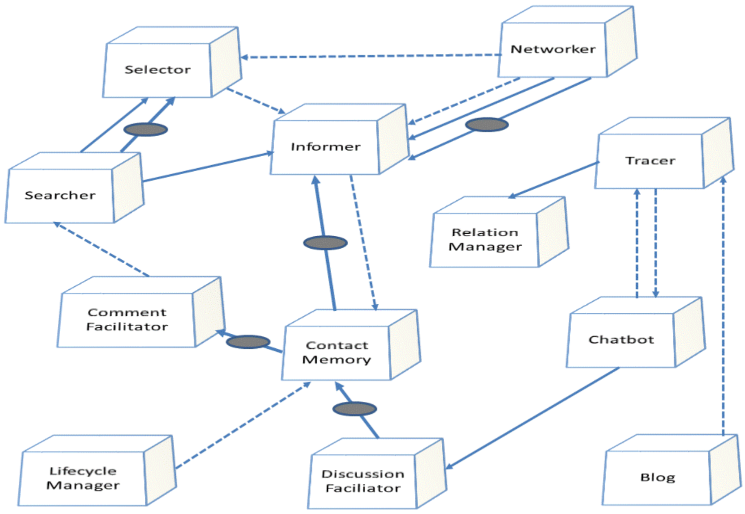

Figure 6. Sample Req-Brick pattern of a sales person

Figure 6 shows for the functional role of a sales person several requirements in terms of blocks when accomplishing CRM tasks. The displayed sales requirement scenario provides detailed requirements in terms of how the domain expert features the information management for the sales innovation. The ‘Informer’ Req-Bricks plays a central role for coordinating the search and selection of data. It is also related to the ‘Contact Memory’ which in turn receives data from the ‘Discussion Facilitator’, bundling social media track data. The setting also reveals that beside data-specific relations the information of others play a dedicated role for sales.

When developing requirements maps using the Req-Brick approach, requirements need to be analyzed according to the different perspectives as encoded in the relations. Of particular importance are filters for (i) the flow of functional control to complete tasks, (ii) data exchange to identify the required information structures, and (iii) information relation, as they may occur in parallel to functional control. Once the data exchanges (ii) are filtered, a SID can be constructed, as each Req-Brick involved in a ‘Deliver’ relation corresponds to a subject, and each business object is part of a data exchange. In addition, each ‘Informs’ (iii) relation leads to subjects related by a message exchange, however, without having to detail a business object. The ‘Follows’ relations can either be refined to data or message exchanges, or indicate internal functional behavior of a subject. Then, it becomes part of a Subject Behavior Diagram.

Once a specification has been validated, it can be executed, and the involved stakeholders receive behavior feedback, both from the perspective of the individual subjects involved (reflecting on action (doing) and interaction (receiving and sending messages of business objects)), as well as from the perspective of organizational behavior (reflecting on overall patterns of interaction between subjects in a SID).

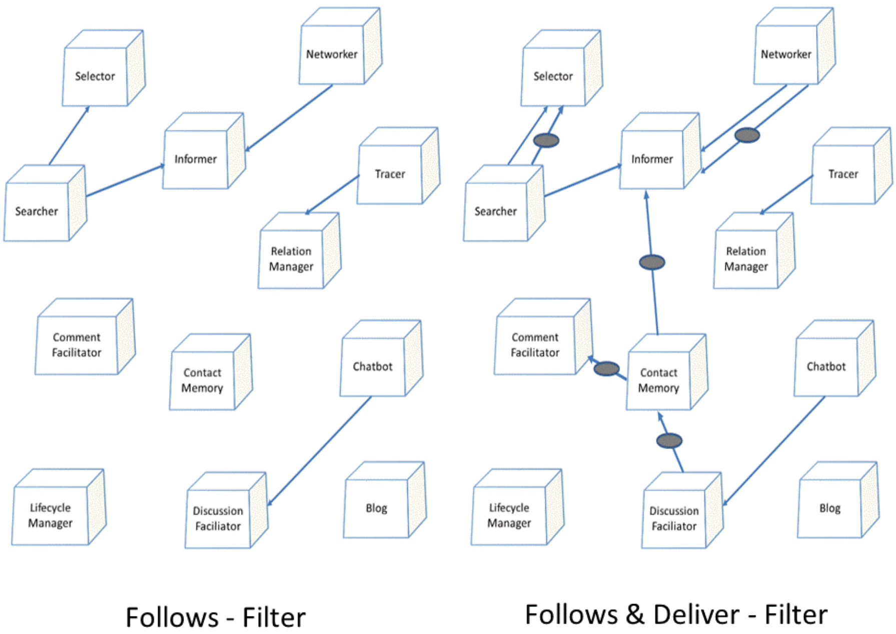

Figure 7. Filtering by relation categories

Latest field tests of the approach have been performed in a sales setting of a method service provider in knowledge-intense business consulting. Hereby, the 5 sales specialists were asked to innovate their sales activities optimizing their interaction with customers. Each of them was observed and supported according to the procedure. Although they were able to articulate their requirements in a coherent and consistent way, the ex-post interviews after completing the requirements engineering phase revealed potential in improvements w.r.t. to the provided categories of relations – stakeholders may not be familiar with identifying particular relations and using them consistently in the course of specification. In addition, the transition to design, constructing SIDs, and refining each subject by means of SBDs requires step-by-step explanations for stakeholders.

4 Conclusion

Once stakeholders are involved in system development their perception of situations, needs, and expectations needs to be captured. Utilizing the subject-oriented paradigm requirements elicitation is performed from a functional *and* interactional perspective. It allows overcoming deficiencies in behavior-centered process and application development due to incomplete specifications. The presented approach shows how to guide elicitation and modelling and finally, to set up a multi-dimensional development space. The procedure to follow contains a preparation, an articulation/representation, and a consolidation phase. Various tools can be applied to support elicitation and specification. In the field tests, the overall goal, namely semantically coherent and consistent requirement specifications from a stakeholder perspective, could be achieved with the help of facilitators, scaffolding and diagrammatic specification support. Future tests and evaluations have to be performed to achieve more practical insights before rethinking the proposed structure of how to convey the S-BPM paradigm and the procedure of the approach.

References

1. Burnay, C., Jureta, I. J., & Faulkner, S. (2014). What stakeholders will or will not say: A theoretical and empirical study of topic importance in Requirements Engineering elicitation interviews. *Information Systems*, *46*, 61-81.
2. Dybå, T., & Cruzes, D. S. (2013). Process research in requirements elicitation. In *Proceedings 3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*, IEEE, 36-39.
3. Eberle, P., Schwarzinger, C., & Stary, C. (2011). User modelling and cognitive user support: towards structured development. *Universal Access in the Information Society*, *10*(3), 275-293.
4. Fleischmann, A., & Stary, C. (2012b). Whom to talk to? A stakeholder perspective on business process development. *Universal Access in the Information Society*, *11*(2), 125-150.
5. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., & Börger, E. (2012a). *Subject-oriented business process management*. Springer Publishing Company, Incorporated.
6. Fleischmann, A., Schmidt, W., & Stary, C. (2013). Subject-Oriented BPM= Socially Executable BPM. In *IEEE 15th Conference on Business Informatics* (pp. 399-407). IEEE, New York.
7. Fleischmann, A., Schmidt, W., & Stary, C. (2014). Tangible or not tangible–a comparative study of interaction types for process modeling support. In *International Conference on Human-Computer Interaction* (pp. 544-555). Springer International Publishing.
8. Fleischmann, A., Schmidt, W., Stary, C. (2015). *S-BPM in the Wild. Practical Value Creation*. Springer Publishing Company, Incorporated.
9. Gane, C., & Sarson, T. (1979) *Structured System Analysis*. Englewood Cliffs, Prentice-Hall, New Jersey.
10. Hess, J., Randall, D., Pipek, V., Wulf, V. (2013). Involving users in the wild—participatory product development in and with online communities. *Int J Human-Computer Stud* 71(5), 570–589.
11. Krenn, F., Stary, C., Wachholder, D. (2017). Stakeholder-centered process implementation support: Assessing S-BPM tool support. *Proc. S-BPM ONE 2017, 9th Int. Conference on Subject-oriented Business Process Management,* ACM.
12. Miller, T., Lu, B., Sterling, L., Beydoun, G., & Taveter, K. (2014). Requirements elicitation and specification using the agent paradigm: the case study of an aircraft turnaround simulator. *IEEE Transactions on Software Engineering*, *40*(10), 1007-1024.
13. Neubauer, M., & Stary, C. (2017). *S-BPM in the production industry. A stakeholder approach*. Springer Publishing Company, Incorporated.
14. Oppl, S., & Stary, C. (2014). Facilitating shared understanding of work situations using a tangible tabletop interface. *Behaviour & Information Technology*, *33*(6), 619-635.
15. Oppl, S. (2016). Towards scaffolding collaborative articulation and alignment of mental models. In: *Proceedings ICKM 2016,* eds: Barachini, F., Hawamdeh, S., Stary, C., Procedia Computer Science 99, Elsevier, 125-145.

16. Rosenkranz, C, Vranesic, H., & Holten, R. (2014). Boundary interactions and motors of change in requirements elicitation: a dynamic perspective on knowledge sharing. *Journal of the Association for Information Systems*, *15*(6), 306.
17. Razavian, M., Türetken, O., Vanderfeesten, I. (2016). When Cognitive Biases Lead to Business Process Management Issues. In: Proceedings of the *9th Workshop on Social and Human Aspects of Business Process Management* (BPMS2'16) @BPM 2016
18. Schneider, F., Bruegge, B., & Berenbach, B. (2013). The unified requirements modeling language: Shifting the focus to early requirements elicitation. In *Proceedings of International Workshop on Comparing Requirements Modeling Approaches (CMA@ RE)*, pp. 31-36, IEEE.
19. Sharma, S., & Pandey, S. K. (2014). Requirements elicitation: Issues and challenges. In *Proc. International Conference on Computing for Sustainable Global Development (INDIACom)*, 151-155. IEEE.
20. Slavin, R., Lehker, J. M., Niu, J., & Breaux, T. D. (2014). Managing security requirements patterns using feature diagram hierarchies. In: *Proceedings 22nd International Requirements Engineering Conference (RE)*, 193-202, IEEE.
21. Seyff, N., Todoran, I., Caluser, K., Singer, L., & Glinz, M. (2015). Using popular social network sites to support requirements elicitation, prioritization and negotiation. *Journal of Internet Services and Applications*, *6*(1).
22. Stary, C. (2014). Non-disruptive knowledge and business processing in knowledge life cycles–aligning value network analysis to process management. *Journal of Knowledge Management*, *18*(4), 651-686.
23. Stary, C. (2017). System-of-Systems Design Thinking on Behavior. *Systems*, *5*(1), 3.
24. Vitharana, P., Zahedi, F., & Jain, H. K. (2016). Enhancing Analysts' Mental Models for Improving Requirements Elicitation: A Two-stage Theoretical Framework and Empirical Results. *Journal of the Association for Information Systems*, *17*(12), 1.
25. Zuckerman, O., Arida, S., & Resnick, M. (2005). Extending tangible interfaces for education: digital montessori-inspired manipulatives. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 859-868, ACM.