SeeMe* – a Process Modelling Notation for socio-technical Requirements-Engineering

Mandana Modiriasari[1], Thomas Herrmann[2]

[1] *paluno-The Ruhr Institute for Software Technology, University of Duisburg Essen, Gerlingstr. 16, Essen, 45127, Germany*
[2] *Institut für Arbeitswissenschaft, Ruhr-University of Bochum, Universitätsstr. 150, Bochum, 44780 Germany*

Abstract

Modeling social-technical systems' work processes as a basis for requirements engineering is a challenging issue. One of the most important pre-conditions for designing a socio-technical system is that system analysts know and understand how the system should support a company's work processes, and what kind of intentions of the involved stakeholders influence each other and substantiate the requirements to be met. The goal of this paper is to provide a modelling method that helps the system analysts to develop this knowledge, and to evaluate the system's features.

Modelling notations for socio-technical systems –such as SeeMe– support the definition of activity sequences as well as the representation of contingency, explicit incompleteness, and flexibility. Other notations – such as i* – allow for representing goals and intentions of actors and the dependencies between these actors.

The focus of this research is to support requirements engineering (RE) for socio-technical systems by merging these two modelling approaches. The result is a new modelling notation (SeeMe*) that extends the process-oriented view by modeling the agent-oriented paradigm of the i*-framework that covers agent properties such as intentionality, autonomy, sociality and boundaries.

Furthermore, the integration of goals into process models supports socio-technical RE. With a case study in the context of a pharmacy, we demonstrate how SeeMe* enables analysts to systematically generate the specifications of solution-oriented requirements in early phases. We conclude that this approach can avoid determining too early whether the solution is based on socio-organizational measures or on technical components and infrastructures.

Keywords

Socio-technical systems, socio-technical modelling, Requirements-Engineering, Business Process Management

1. Introduction

IT-Applications become increasingly complexer with respect to their involvement with social interaction, not only within organizations and between them but also with respect to the inclusion of people in various roles such as relatives, community, members friends etc. (Herrmann et al., 2017). Furthermore, more and more available apps for mobile devices include functionality for social networking. Accordingly, health care applications have to react on the fact that patients are part of social relationships that help to support their recovering. These kinds of increased social involvement mirror the emerging relevance of taking socio-technical aspects into account for designing software solutions, developing organizational measures, making people competent, and for integrating all of these elements. Successful IT-applications are a result of deliberate socio-technical design (Baxter & Sommerville, 2011; Fischer & Herrmann, 2015).

Socio-technical design is not only about software design but also about designing and managing social relationships by establishing coordinative conventions, agreements on rights and duties of roles, communication channels, organizational practices and processes etc. (Baxter & Sommerville, 2011; Herrmann et al., 2021; Mumford, 1995). This inclusion of social aspects has decisive

8th International Workshop on Socio-Technical Perspective in IS development, August 19-21, 2022 (STPIS'22)
EMAIL: mandanamodiri@yahoo.com (A1); thomas.herrmann@rub.de (A2)

relevance for requirements engineering (RE). RE has to be focused on socio-technical design (Jones & Maiden, 2004). That means that RE has not only to develop requirements that can be met by software solutions but also requirements that deal with the social side of a socio-technical solution. A procedure is needed that helps to develop socio-technical requirements that are in the early phases of STD neutral with respect to the question whether the solution is based on technology or on establishing certain social and organizational practices.

Therefore, we suggest that procedures of RE should not only be based on functions or objects or data to be handled, but should take processes as a basis since within processes, technical functions and organizational measures are immediately interwoven (Cardoso et al., 2009; Herrmann & Hoffmann, 2005). Thus, we take socio-technical process-modelling and-analysis as an appropriate method to support early phases of RE. The perspective of "early phases" refers to those states of design where we do not exactly know whether or which problems are solved by technical or by organizational means. As a means to support this kind of socio-technical RE, we take process modeling methods as basis. However, most common process modelling methods such as BPMN (Recker et al., 2006) are not intended to support the socio-technical perspective (Koliadis et al., 2006). Thus, we seek to answer the following research questions with this paper:

1. How does an appropriate process modelling notation for RE within socio-technical design look like, and
2. how can socio-technically oriented process models be used to derive requirements?

The basic idea for answering this question is to combine SeeMe (Herrmann, 2009) –a process-oriented, semi-structured socio-technical modeling method– with i* (Yu, 2009) that focuses on the intentions of social actors and the dependencies between them.

In the literature review (section 2), we identify possible approaches to be combined into such a socio-technical modelling notation. We demonstrate the usage and usefulness of the new modelling method for the case of delivery of medicine in a pharmacy. This case and its relevance for socio-technical design is described in section 3. Section 4 clarifies how we have constructed a new process modelling notation and give examples of how it looks like and how requirements can be derived. The concluding section discusses the advantages and limitations of the method.

2. Background: socio-technical requirements-engineering and the relevance of processes, intentions and dependencies

2.1 Socio-technical design and RE

The goal of requirements engineering is to produce a set of requirements which, as far as possible, is complete, consistent, and relevant and reflects what the customer actually wants. Requirement engineering procedures aim on defining and communicating all business, customer and system requirements (Sommerville & Sawyer, 1997). This includes understanding the system's environment and considering the organizational and social context in which a new system would function (Nuseibeh & Easterbrook, 2000). Developing a socio-technical system as a whole, implies an engineering process where not only software but also hardware, system interactions with human users, and various constraints imposed by organizational and social policies and regulations must be considered (Sommerville, 2007, 2014). If these aspects are partially ignored, the requirements engineering process may result in systems with irrelevant or inappropriate features that do not meet the stakeholders' goals. Consequently, if the business environment is not correctly analyzed, the system may not meet expectations, and business/IT alignment would thereby not be achieved. To focus on this alignment and on the goals of the users and affected stakeholders, a best match between the technological and social aspects has to be pursued (Mumford, 1994). Therefore, in the early stages of the development of a socio-technical system, modeling and analytical techniques should help understand the organizational environment in terms of its goals and the interdependencies between them as well as mutual constraints of different actors, and support taking the appropriate design principles into account (Clegg, 2000; Herrmann et al., 2021).

In accordance with Bryl (2009), in a socio-technical system, human actors, organizational units and software rely on each other in order to fulfill their respective objectives. Therefore, at the early stages of a socio-technical system development, there is a need for modeling and analysis techniques that help understand the organizational environment in terms of goals, interdependencies and mutual constraints of various types of actors.

One prominent approach of supporting RE methodologically is the usage of modelling notations that help to represent as-is situations as well as the as to-be models. UML (Fowler, 2004) is the most prominent set of modelling notations to support software-engineering. However, its focus does not take business processes into account but use cases, classes of objects, interactions, control flows etc. In UML, the description of use cases with tables include sequences of actions and conditions whether they take place or not. However, this consideration of sequences of actions is handled much more explicit in the BPM world. Furthermore, the social side is neglected by UML, for example vagueness, close collaboration, rights and duties etc. are not systematically considered (Goedicke & Herrmann, 2007). Thus, Fischer and Herrmann (2011) propose SeeMe as a means that helps to create diagrams of socio-technical processes that also support participation of affected users and stakeholders before and during the use of socio-technical systems. Compared to other methods, such as FRAM (Salehi et al., 2021) or the soft-systems approach (Checkland, 1999; Checkland & Poulter, 2020), SeeMe focuses on process-oriented, flexible sequences of activities, collaborating roles and assigned resources.

2.2 SeeMe vs. BPMN

Process modeling is acknowledged as a means of supporting the development of software (Becker et al., 2000). Being graphical in nature, it is easy for communication with stakeholders, to represent relationships among entities and activities as well as to understand the overall operations.

A "universal" notation for business process models does not exist. Each notation has been created for specific purposes. Thus, they concentrate on those aspects that have been considered as more important by their designers. In addition, each method has its own advantages and disadvantages, and each method is restricted with respect to the view on the world that it can present. However, the Business Process Modeling Notation (BPMN) is specifically developed to integrate various views and all levels in the organization (Recker et al., 2006). It is designed to provide an intuitive and easy way for non-expert users in BPM to understand the notation and represents the semantics of complex processes easily and in an intelligible form.

However, from the viewpoint of practitioners, the limitations of BP-modeling methods might be summarized as follows (Coelho, 2005, p. 122):

- "Process improvement is focused on methods and tools improvement, forgetting sometimes the people dimension,
- …
- The role of the business people is focused in the transfer of information for the team project and not in provoking a rethinking of the organization,
- The approach is only based on individual interviews,
- The task of documenting methods and procedures is too heavy, making it difficult to run the workshops and interviews with enthusiasm on the part of business people."

Regarding these problems, research has been carried out to overcome the shortcomings and improve BPMN by extending it. For instance, in a study BPMN is extended with a goal-oriented approach to obtain a model that reflects the continuous changes checking between model and reality (Santos et al., 2010). All in all, BPMN and it's extensions is not designed to address issues that are typical for social structures (Zarour et al., 2019) such as intuition, improvisation, flexible task allocation, coordination on the fly, as already has been outlined in the early discussions on workflow management systems (Herrmann & Hoffmann, 2005).

To overcome these limitations the semi-structured, socio-technical modeling method, SeeMe, was established based on the concepts of uncertainly, explicit incompleteness, multiplicity of perspectives, and self-referential meta-relations. It is especially designed for representing socio-technical work processes and structures and can help people to understand the specific features and requirements of 'their' socio-technical systems (Herrmann et al., 2004).

SeeMe acknowledges that the psychological and social factors affect process efficiency and effectiveness and that organizations are made up of people who work with IT systems in order to achieve goals. It is appropriate to use a modeling notation to build diagrams that consider incompleteness and vagueness when some details are not yet clear during the early phases of developing socio-technical systems or processes (Goedicke & Herrmann, 2007): Furthermore, this modeling notation should be able to represent the dynamic intertwinement between humans, people, and technical components, while also integrating both formal and informal processes.

In summary, the SeeMe-method is the more appropriate process notation for modeling systems that include the interaction of humans and technical components and include incompleteness in early phases of the modeling process. SeeMe is a modeling notation that extents the functionality of other notations by adding many features for communication and presentation of vague content. Most important, in comparison to BPMN, it allows for flexible task-allocation that is decided on-the-fly by collaborating roles. SeeMe helps to model situations where roles can flexibly start and terminate activities independently from pre-planned sequences, or where they can flexibly include resources, and SeeMe does not require the explicit representation of the collaborative decision-making behind this flexibility (Herrmann et al., 2004). Fig. 1 gives an example by representing aspects of the analyzed pharmacy case. However, SeeMe cannot represent the goals and interests of stakeholders. In social environments, the intentional dimension of actors and dependencies among them should not be underestimated. Therefore, an extension of SeeMe is reasonable in order to elicit requirements that takes the intentions of involved actors into account.

2.3 Dependency modelling with i*

"The i* modeling approach is an attempt to bring social understanding into the system engineering process by putting selected social concepts into the core of the daily activity of system analysts and designers" (Yu, 2009, p. 100).

According to YU et al. (2011), the conventional requirements analysis adopts a mechanistic view of a world that consists of fully known entities and activities. By contrast, with a social worldview, we see that the world has intentionality, interests, motivations etc. behind behavior. Therefore, it makes sense to shift the focus away from the usual focus on activities and information flows – as is the case with business process modeling – to questions such as:

- "What does each actor want?
- How do they achieve what they want?
- And who do they depend on to achieve what they want." (Yu, 2009, p. 100)

In addition, the i* modeling framework integrates some aspects of social modeling into information system engineering methods, especially at the level of requirements. I* emphasizes the primacy of social actors, in contrast to traditional systems analysis, which tries to abstract away from the social aspects of systems. The analysis focuses on how well the objectives of different actors are achieved, given some configuration of relationships between human and system actors, and what reconfigurations of those relationships can help actors advance their strategic interests.

The i* framework offers modeling notations for two types of models, each one corresponding to a different level of abstraction: the Strategic Dependency (SD) model represents the intentional level and the Strategic Rationale (SR) model represents the rationale behind it (Yu et al., 2011). SD models represent actors and a set of dependencies that represent the relationships among them. Dependencies express that an actor (depender) depends on some other (dependee). Furthermore, the depender depends on a dependum (a resource, task or goal) that has to be contributed by the dependee. If the dependee does not provide the dependum, the depender is negatively affected with respect to its goals. Secondly, with the SR model the intentionality behind an actor is represented by a hierarchy of intentional elements delimited by its boundary (Yu et al., 2011). The dependencies of the SD model are linked to intentional elements inside the actor boundary (Grau et al., 2008). The Strategic Rationale model examines "inside" actors to model internal intentional relationships at a more detailed level (Castro et al., 2000). Fig. 2 gives an example, by representing aspects of the analyzed pharmacy case.

The i* approach not only differs significantly from business modeling, but also contrasts with common approaches of modeling mostly software-oriented systems, such as UML and its basic notations, including use case and class diagrams. Whereas UML specifications describe what a system should do, I* models include cross-references to goals and soft-goals that explain why it should do so.

2.4 Comparison between SeeMe and i* – strengths and weaknesses

As argued above, SeeMe is an appropriate modeling method for socio-technical systems and processes that allows indicating incompleteness and vagueness. Subsequently, in early phases of modeling, it allows the modeler to leave parts of the modeled reality intentionally unconsidered and to focus on those details that need support or improvement. By contrast, i* models seek to give a complete picture of the actors' intentions and dependencies and therefore tend to become large and hard to handle in the case of real systems. The only kind of vagueness within i* is the differentiation between hard goals (its achievement can be clearly determined) and soft-goals (e.g. good consulting, see Fig. 2)

Furthermore, SeeMe modeling covers aspects of workers' actions and the sequence between them. It is related to BPM in that dependencies between activities and thus control flows are represented, and logical ramifications can also be modeled depending on conditions. SeeMe can also represent information flows and communicative exchange and therefore helps to build bridges to the relevant aspects of software development. The absence of control and information flows within i* is a consequence of its focus on social relations but it is not obvious why the representation of intentionality and dependencies could not be combined with representations of sequences of action and their dependency on varying conditions as the availability of information and resources. Dependency diagrams are not process-oriented and cannot directly be used to enable a step-by-step improvement of functionality and forms of interaction with the technical system.

On the other hand, it is not a necessity that SeeMe neglects aspects such as goals and the dependencies between actors and dependencies between goals, conditions, tasks, etc.

According to their characteristics, SeeMe is the only modelling notation allowing for representing incompleteness, which is typical for socio-technical systems (Cherns, 1987), and i* the only one that helps present the interdependencies between intentions of actors. We argue that the integration of SeeMe and the i* framework can alleviate the weaknesses of each other and help organizations to understand their requirements from different perspectives. The SeeMe modeling notation is an activity-based model that helps in understanding dependencies between tasks: A task in a process is enabled after the completion of the preceding tasks. However, SeeMe models do not include any direct information why the included roles may want to execute the tasks or on what social pre-conditions the quality of task executions is based on. Consequently, the combination of both modeling methods can proof fruitful as a basis for socio-technical RE. We are aware that there are a variety of approaches to RE in software engineering. However, we do not now any literature so far about RE methods that are open to technical as well organizational measures that help to meet the requirements.

3. The pharmacy case – delivery of medicine

To demonstrate how the new modeling notation –SeeMe*– will work and to provide plausibility for its usefulness and usability, we refer to a case study. The case is covered by the business of a pharmacy and focuses the complex process of medication delivery to the customer, denoted as the patient. This is a rich process since it includes various conditions such as patients with or without a prescription; whether medication is immediately available, has to be ordered or has to be substituted; additional consulting with the doctor is needed or not; etc.

Furthermore, various goals have to be considered such as fast delivery vs. thorough information of the patient or building trustful relations. One might question whether this case is a representative example for a socio-technical system. Pharmacy processes are part of the healthcare industry which is very different from other industries because of the intensity of the personal interactions. Health care is all about people: patients and their families and friends, and the various healthcare professionals and workers who all have a variety of interests and motivations. Therefore, when analyzing, designing, implementing and improving healthcare systems, the people dimension should be at the forefront, and a socio-technical approach is specifically feasible (Ackermann et al., 2017).

We use the case study as a means to provide an empirical basis for dealing with our research questions. The empirical investigation helps understand how the social factors such as goals, dependencies, and motivation of actors can influence the process modeling of a socio-technical system and support the elicitation of socio-technical requirements.

The participants of this case study include three pharmacists, seven pharmacy technicians, and one pharmacy assistant from a modern pharmacy which is located in a small city of North Rhine Westphalia, Germany. The ages of participants range from 24 to 60 years old; two are male, and nine are female. On average, the participants have over 15 years of work experience.

3.1 Modeling examples

This sub-section provides modelling examples of a small detail of the medication delivery process where a patient, who has no prescription, seeks advice to deal with a health problem. To demonstrate the difference between the two modelling approaches the two exemplary diagrams cover the same situation with in the overall process. Furthermore, the two examples of Figure 1) and 2) will help to understand the result that is achieved by merging the two different modelling notations (Figure 3).

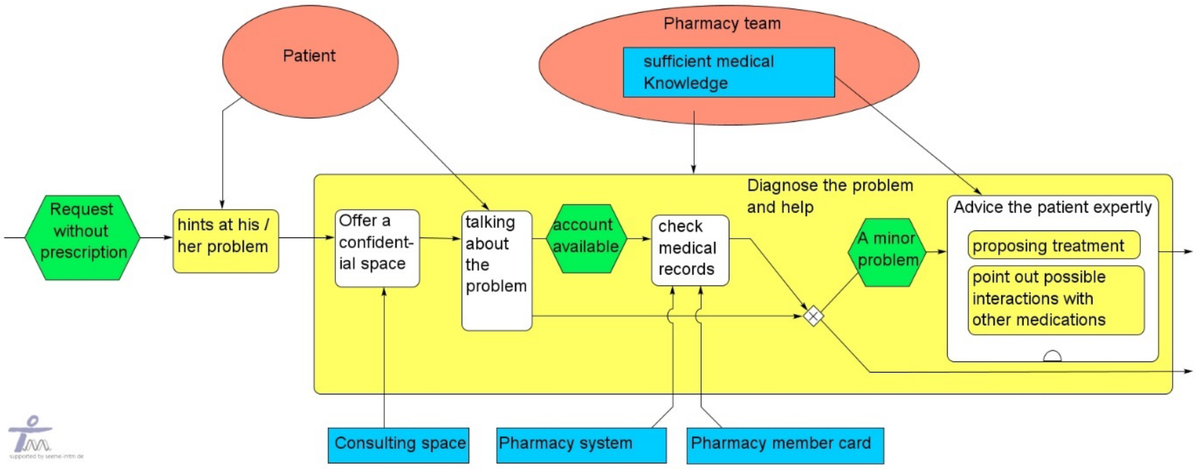

Figure 1: A process section of the pharmacy case modeled with SeeMe

In the upper part of the SeeMe-diagram (Fig. 1) the active roles are depicted. The activities are represented with rounded rectangles. They can be embedded into each other. The relation that points from “pharmacy team” to “diagnose the problem and help” expresses that the team takes care of all the sub- activities. By contrast, the patient is only involved into “talking about the problem”. Since the starting point of the relation cuts into the boundary line of the role “pharmacy team”, it indicates that not all members of the team will become active. This is a kind of vagueness that indicates that the team decides on the fly who will do what. Another example of vagueness is the semi-circle at the bottom of “Advice the patient expertly”. This symbol expresses that there might be more sub-sub-activities than “proposing treatment” etc. The control flow expresses that a confident place is offered before the consulting starts. The exclusive-OR-connector indicates that the pharmacy gives only advice if minor problems have to be dealt with. Otherwise, other measures might be taken as the open-ended arrow on the bottom-right depicts. On the bottom of the diagram, entities are shown as rectangles that mainly represent the technical equipment, tools, documents etc. So far, we can only assume why a "consulting space" is offered, the reasons behind this offer are not clearly depicted. Knowledge owned by roles is also represented by an entity, however, this kind of entity is embedded into the role to indicate the ownership.

By contrast, the i*-diagram (Fig. 2) includes goals that help to understand these kinds of reasons. The empirical investigations revealed that the pharmacy team intends that the patient "feels comfortable to talk …". Consequently, this soft-goal is depicted within the boundary of the pharmacy team. There are other soft-goals, such as "efficient consulting", "increasing patient satisfaction", and on the patient's side "high quality of consulting" and "will receive a medicine" what is a hard goal. Activities or tasks are represented with hexagons. It is shown by the help-relation to which goals the tasks contribute. Thus, it becomes clear "offer a confidential space" is carried out to make the patient "feel comfortable …". Sequences between tasks cannot be seen. Embedment of tasks and the alignment of resources (rectangles) is expressed by crossed lines. Obviously, it is not clear on a first glance what really happens and how the delivering of a service to the patient is conducted. Therefore, it might become difficult to detect when and how support by software or managerial interventions etc. might be helpful. For example, it could be required that the patient can enter their hints with a software while waiting for the consulting and that this information is used for "talking about his/ her problem". On the other hand, it is easy to detect that an additional task is needed that contributes to the increasement of sufficient medical knowledge. This task could just be inserted somewhere within the boundary of the pharmacy team, while such a task is hard to insert into the sequence of the SeeMe-diagram.

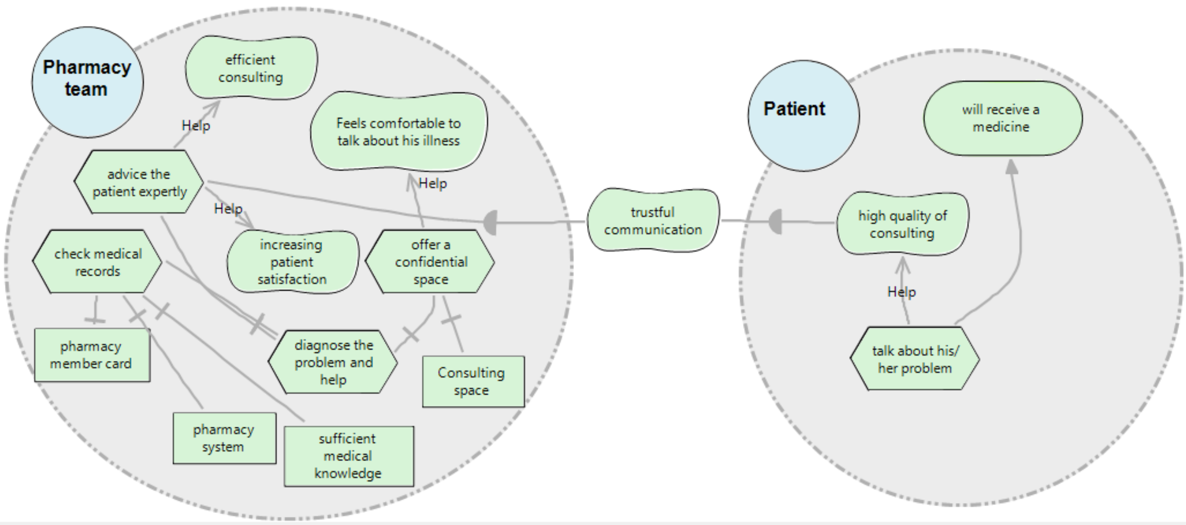

Figure 2: Goals and tasks, and their relationship shown with the i*-notation

The i*-diagram represents one example of a dependency. The pharmacy team depends on the patient when they try to give advice; more exactly, they depend on the patient's having an interest in "high quality of consulting" (instead of seeking some kind of a fast, superficial treatment). The dependency is related to a dependum, the soft-goal of "trustful communication" that has to be pursued by both sides, if "advice the patient expertly" should be carried out successfully. Having this soft-goal explicitly depicted is –for instance – a valuable guidance for considering privacy issues when a socio-technical solution is specified.

3.2 The empirical basis

The diagrams shown in this paper are based on systematical empirical work. Two main series of interviews and three meetings with participants in the pharmacy were conducted. The results of the interviews and meetings were documented in written notes and sometimes in recorded voices. The meetings were held to validate the diagrams and collect feedback from the pharmacy team. The first series of interviews were conducted to generally understand current processes in the pharmacy. Its results have been used as a starting point to compare SeeMe- and i*-models. The second series of interviews provides an in-depth view of a selected process in the pharmacy. It helped to understand those goals that are relevant to detect the need for improvement and the related socio-technical requirements, and thus served as a basis to develop the SeeMe*-to-be-models. The detailed results of the empirical investigation can be found in Modiriasari (2022), while this paper is limited to the presentation of exemplary aspects.

4. SeeMe* - a merged modelling notation

4.1 The elements of SeeMe*

As described above, i*-models represent goals and intentions, and answer the question who does what why. An SR model allows modeling of the reasons associated with each actor and their dependencies and provides information about how actors achieve their goals and soft-goals. For example, the patient's soft goal of receiving high quality consulting depends on the pharmacy team's activity of "advising the patient expertly" (as part of the overall activity of "diagnose the problem …"), while "trustful communication" can be annotated (see fig. 2) as a precondition for the communication taking place. These aspects are the main elements we intend to add to the SeeMe notation, since they help to understand why the pharmacy is doing what. The method of creating the new modelling notation was to develop comparable meta-models of both notations and to systematically merge them on this meta-level (Modiriasari, 2022). However, this is a lengthy and detailed procedure that cannot be presented for the purpose of this paper. Instead, we present the result of this meta-model-merging with Table 1), by describing the elements and types of relations that are combined. The formal definition of SeeMe* is given by a meta-model that is based on the UML-notation of classes in Modiriasari (2022) and is extended by a set of patterns how the notation can be used.

According to Table 1, the SeeMe*-notation consists of 20 different semantical types of elements where 15 are taken from SeeMe and 12 taken from i* (with 7 overlapping, similar semantical constructs). To keep the types of elements manageable, we have not included every type of elements from the merged notations. For example, we did not take meta-relations from SeeMe with which one can express that an activity determines the rights and duties of a role or who belongs to a team; and we neglected – for instance – the differentiation of types of actors (role, agent, position) as it is possible with i*, and the differentiation between SD- and SR-models.

4.2. A modelling example

Fig. 3 gives an example of modeling with SeeMe* and how the two models of Fig. 1 and Fig. 2 can be combined. To allow for keeping an overview, five swim lanes are introduced with the help of dividers so that goals, roles, activities, dependencies and entities can be clearly recognized. The numbers of the types of elements of Table 1 are used to indicate examples of these types in Fig. 3 and Fig. 4. Most important is that the soft- and hard-goals can be identified as well as their relation to roles and activities. Thus, it can be seen why a role is doing what, for instance, that the patient gives hints about his/her problem, since s/he wants to receive medicine; or the pharmacy team advices the patient with the goal of increasing his / her satisfaction. Empirically, this sub-goal has been derived from the manger's request that the whole socio-technical pharma system should help to increase the patients' loyalty to the pharmacy. Fig. 3 includes one dependency taken from Fig. 2: The pharmacy team depends on the patient's willingness to talk about the problem when they try to successfully diagnose the problem. The dependum here is that both sides manage to meet the goal of "trustful communication".

Table 1
Elements derived for SeeMe*

#	SeeMe Element	I* Framework Element	SeeMe*-Element
1	Role	Actor	Role and embedded sub-roles
2	Activity	Task	Activity connected to roles
3	Entity	Resource	Entity
4	-----	Soft-Goal	Soft-goal
5	-----	Hard- Goal	Hard-goal
6	Embedment: activities into activities	Internal element relationship: Decomposition Link	Embedment of activities
7	Embedment of roles into roles or entities into entities	-----	Embedment of roles into roles or entities into entities
8	Embedment: All possibili-ties of embedding different types into one another	-----	Competencies of roles are embedded as entities into roles (8a) Task delegated to systems are shown as activities being embedded into entities (8b).
9	-----	Actor boundaries indicating which tasks, goals and resources belong to an actor	Relations between roles and tasks, and between roles and goals
10	Relation: Role has expectation to a role	Dependency relationship between actors and between their tasks or the goals	Dependency relation between roles, or between the assignments of tasks to roles or of goals to roles, with attached dependum of four possible types
11	Relation: role carries out activity	Task in an actor boundary	Relation: role carries out activity
12	Relation: Activity follows on activity	------	Relation: activity follows on activity
13	Relation: activity modifies entity	------	Relation: activity modifies entity
14	Relation: Entity is used by activity	Entity as a Task decomposition	Relation: entity is used by activity
15	------	Internal element relationship: Means-Ends Link to a hard-goal	Means-Ends Link to a hard-goal
16	------	Internal element relationship: Contribution Link to a soft-goal	Contribution Link to a soft-goal
17	Vagueness/ Incomplete-ness with semi-circles or cutting relations	------	Vagueness/ Incompleteness with semi-circles or cutting relations
18	Logical connectors	------	Logical connectors
19	Conditions or events as modifiers	------	Conditions or events as modifiers
20	Dividers	------	Dividers that help differentiate between roles, activities, dependum and entities

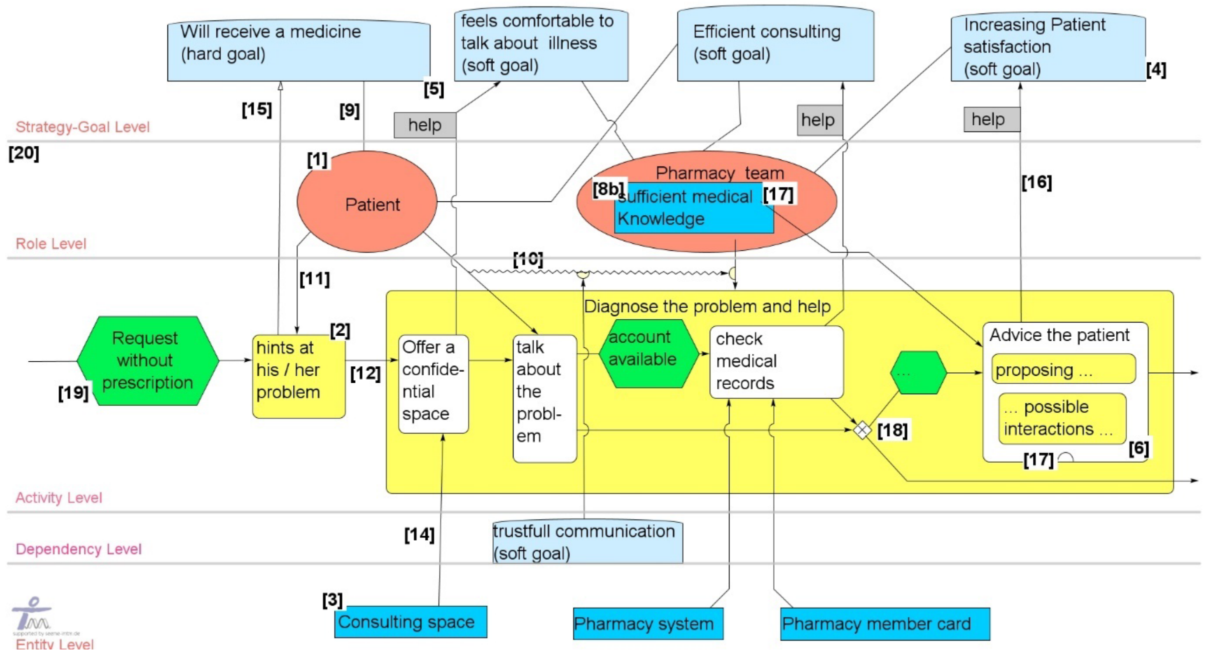

Figure 3: Merging the SeeMe- and i*-diagrams with SeeMe*

4.3 Deriving requirements from SeeMe*

The SeeMe*-model as shown in Fig. 3 is already a to-be model that includes data elicitation about problems and requirements. In our case study, we interviewed the manager of the pharmacy to collect his needs and proposals for improvement. This resulted in a set of 9 "user stories". A typical example is: "As a pharmacy manager, I need an efficient service time, so that patients are not consulted too long." The case study was focused on the needs of the pharmacy team as a basis to identify examples that help to demonstrate the use and usefulness of the new modelling notation. In a real project of socio-technical design, a participatory approach would have included the patients' perspective. However, from the feedback that the pharmacy team gave, it is quite obvious that patients are also interested in an efficient procedure.

Taking a list of these kinds of needs and requirements as a starting point, a systematic procedure can evolve to develop solution-oriented requirements of a socio-technical nature. In a first step, we look for each requirement whether it leads to new tasks, goals, dependencies or resources that are not sufficiently represented in the SeeMe*-diagram that is already available. The result may look as follows (see Table 2).

Table 2
Deriving new SeeMe*-aspects from requirements

No.	Gathered Requirements	New Task	New Goal	New Dependency	New resource as skill
1	As a pharmacy manager, I need an efficient service time, so that patients are not consulted too long.	Make consulting more efficient	Efficient consulting		

The new goal "Efficient consulting" is already depicted in the SeeMe*-diagram (see Fig. 3). Alternatively, instead of using a notation element for goals, one might also try to include an activity like "Make consulting more efficient" into the process diagram. However, this is not feasible since this would be an ongoing activity, even more an attitude. Thus, it is hard to decide, where such an activity would have to be systematically positioned in the process diagram since it is interwoven with all activities of the pharmacy team. Thus, it is much more appropriate to represent this kind of behavior by a soft-goal. Such a decision of how to model the situation is an instructive demonstration of the advantages achieved if goals of the included actors can be represented and considered.

In the second step, we analyze which solution requirements will support a newly found -goal. Table 3 demonstrates the result of such an analyzes for one example. For the patients it is feasible that they do not have to tell twice what the pharmacy needs to know to help them (Table 3, 1a). For the patient it is not relevant how the pharmacy meets this requirement, by organizational measures (such as one face to the customer) or by a technical information system. However, it should be noted that recalling this knowledge might be a subject of vagueness and incompleteness. For the pharmacy team, it might be most efficient to work with an information system (Table 3, 1b), however, to meet this requirement, the agreement of the patient is needed as a precondition.

Table 3
Deriving socio-technical solution requirements

No.	New Tasks or goals from goal-oriented to-be SeeMe*	Socio-Technical Solution Requirement	Vagueness/Incompleteness	Precondition
1a	Make consulting more efficient	As a patient, the socio-technical pharmacy system knows the information that I have already shared with the pharmacy	It is not sure how far the pharmacy team can recall all relevant knowledge (note that the arrow is cutting into "… medical knowledge" to express this vagueness.)	
1b		As a pharmacy team, I can access information about the patient		The patient agrees to the usage of a personal account.

In the third step, it is reasoned about how the solution requirement can be fulfilled. Table 4 gives an example for the requirement 1b) from Table 3.and demonstrates the differentiation between a technical and a social solution for this requirement. Most important, a definite point within the process of handling a patient has to be identified where the pharmacy team addresses the topic of the advantages of an account. From the viewpoint of the pharmacy, it appeared reasonable to mention the advantages of such an account in the context of the medical advice. This is an organizational solution. With respect to this solution, the relevance of integrating i* into a process view becomes apparent: for the pharmacy team it has to be clearly recommended when they should refer to the possibility of an account. Furthermore, a technical solution –providing an App for the patients with which they can access their account– can help to increase the acceptance.

Table 4
Specifying technical or social solutions that fulfill requirements

No.	Solution Requirements	How	Who/ What
1b	As a pharmacy team, I can access information about the patient	To every patient the advantages of agreeing to an account are explained during the activity "advice the patient expertly"	Social / organizational
		Provide an App for the participants with which they can monitor their account	technical

The solutions described in the column "how" can be integrated in a new SeeMe*-model as shown in Fig. 4. This model can be used to support bringing the solutions into reality or new aspects can be modeled as shown in Fig. 4. Such an extension is helpful to decide whether a new cycle of the three steps described above is reasonable. For example, Fig. 4 adds in comparison to Fig. 3 that the pharmacy team has the goal that the account of every patient is established and maintained. Achieving this goal depends socially on the patient's agreement; and this dependency can be conditionally influenced by the patient app. Thus, the dependencies can be used to demonstrate how the intertwinement between social aspects (e.g. the activity "explain the advantages of a personal account") and technical functions (e.g. the patient app) can be specified. Just offering an App for monitoring the personal account might not be sufficient to convince a patient. For example, the app could additionally integrate a function that reminds the patients when they should take their medicine – as already depicted in Fig. 4. Furthermore, privacy issues have to be considered. The new diagram could be used to discuss whether the activity "explain the advantages of a personal account" needs further support and additional features. Therefore, new (sub-)goals and ways of how to reach them as integral part of the pharmacy process can be considered. Starting and repeating this kind of new cycles can be considered as a basis for agile procedures.

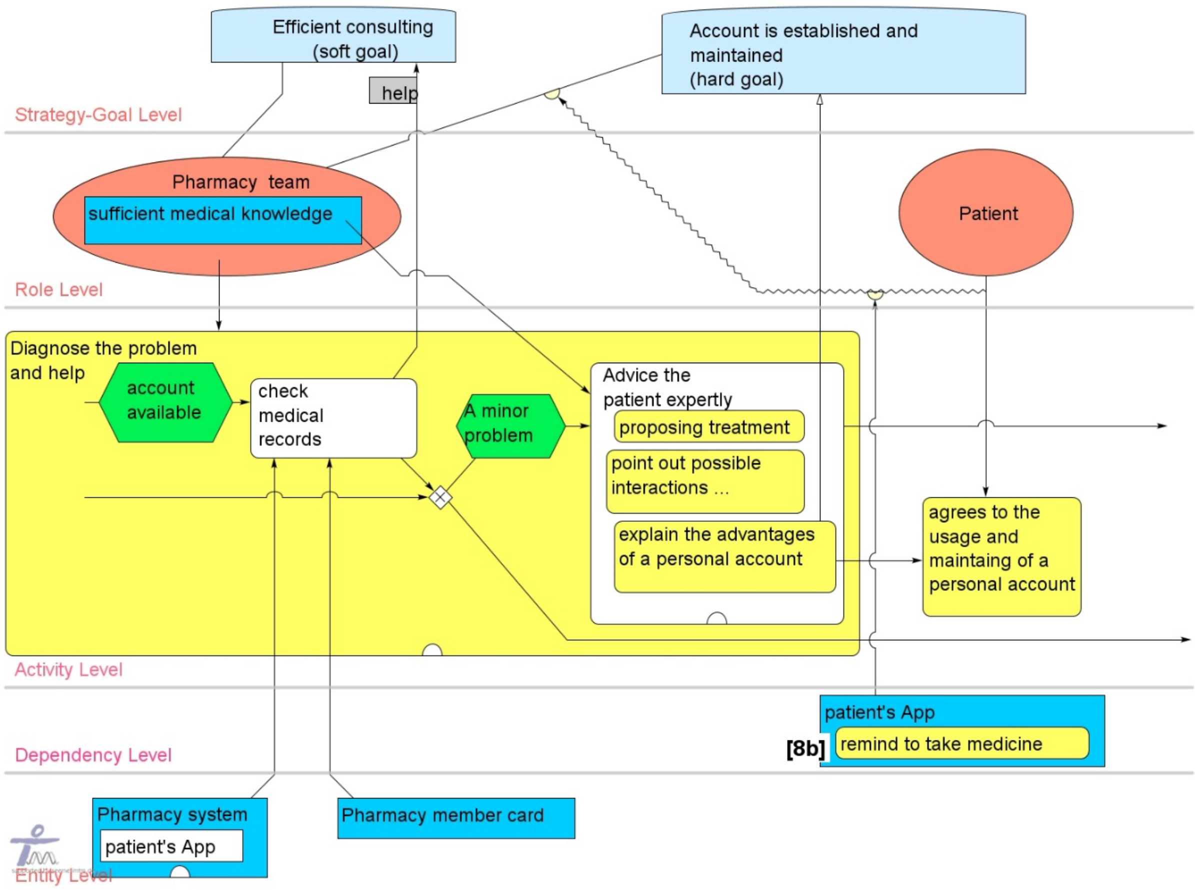

Figure 4: Modeling the derived solutions

5. Conclusion: Strength and limitations of the approach

With respect to the first research question of how an appropriate socio-technical modelling notation will look like, we suggest that the merging of SeeMe and i* lead to a feasible solution. Obviously, the notation of social dependencies between stakeholders' intentions can be integrated into process diagrams, and it can be demonstrated how the support of goals can be integrated into a process model. SeeMe* can clearly be considered as being socio-technical: It is not only derived from two methods that are dedicated to socio-technical systems design but it is open for socio-organizational as well as technical solutions and it obviously demonstrates how both can be integrated. The three steps of how to derive requirements are a proposal to answer the second question, and we explain how the development of socio-technical requirements can be repeated in agile cycles. Thus, an iterative step-by-step procedure of requirements elicitation can be supported with SeeMe*-diagrams, that make social-relationships and dependences on interests of involved people comprehensible. If those stakeholders who are involved in the process of RE can look on the goals, and can perceive interests and potential conflicts between them, they are supported to review the as-is process with respect to the question whether the modification of activities or certain types of software-support would be adequate to achieve their goals. Agile procedures can be realized by an alternation between drafting to-be-models and deriving requirements and solutions. Allowing for agility complies with the evolutionary character of goals that have to be revisited from time to time.

Since real systems include flexibility, such as who of the team will serve the customer, which case is considered as a minor problem, whether the available knowledge will be sufficient to advice a customer etc., it is not possible to guide the employees with deterministically defined sequences of activities and task allocations (as proposed by BPMN). Thus, guiding people with noting down the goals to be achieved represents an appropriate completion to a kind of modelling of processes that includes flexibility. Furthermore, SeeMe*-models can be used to evaluate the appropriateness of a socio-technical solutions by checking the to-be-SeeMe* diagram step-by-step along the sequence of activities whether every goal is addressed.

Furthermore, the intertwinement between technical components and social measures can be analyzed and improved with respect to goals and interests that have to be balanced. For example, the pharmacy's interest in efficiency –based on an information system that stores information about patients– can be in conflict with the patients' feeling comfortable to talk about their illness (see Fig. 3). Additionally, goals can be used to demonstrate intentions that are not related to a single sub-activity of a more complex process but are holistically present in every activity or at least most of them, as discussed for the goal of efficiency.

One of the most challenging limitations of the proposed approach is the complexity of diagrams that present real cases as a whole. While the example used for this paper has 21 elements and 27 relations (Fig. 4), the diagram that was developed during the empirical work includes 121 elements and 142 relations of the types that we specified above in Table 1. On the one hand, this type of large diagrams helps to gain an overview and to see the overall relationships and reciprocal influences. On the other hand, focusing on smaller diagrams, as presented in the paper, helps to consider problems in detail and to increase the chances for a deep understanding for specifying creative solutions by an in-depth analysis. Eventually, it might be reasonable to focus on the most relevant goals or – at least– the possibility of a hierarchical embedding of sub-goals into goals should be provided. Relations between goals and roles or tasks can then be reduced to the goals on the upper hierarchical level. Adding a dependum to the dependency relations and to present them on the dependency level adds additional structural complexity to the diagrams. It might be worth considering whether a dependum can be replaced by a normal element and a its support of soft- or hard-goals, by just using the means-end link-relation (Table 1, no. 15) or the contribution-relation (Table 1, no. 15). All in all, the complexity issue is also the decisive reason why it would not be feasible to represent a certain case, where RE takes place just by two diagrams, one based on SeeMe and one based on i*. The design of SeeMe* also includes a number of decisions of reducing the set of types of elements and aspects that are presented in diagrams.

One could argue that the two types of views of SeeMe and i* need to be presented separately because the objectives and basic assumptions behind the two modeling notations are very different. However, if these differences involve conflicting assumptions about how to design a solution for a socio-technical system, these conflicts must be addressed in any case, and the earlier the better in the course of the RE.

As for now, the notation needs a careful validation performed by practitioners in real working settings. Obviously, further studies are needed specifically with respect to the understandability of the SeeMe*-diagrams for those stakeholders who should participate in RE. Furthermore, there should be further studies to check whether the approach of deriving requirements with the help of SeeMe* can be applied in other areas than healthcare and whether using SeeMe* or only i*, SeeME or BPMN makes a relevant difference.

References

Ackermann, M. S., Goggins, S. P., Herrmann, T., Prilla, M., & Stary, C. (2017). Designing Healthcare That Works – A Socio-technical Approach. Academic Press.

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. Interacting with Computers, 23(1), 4–17. https://doi.org/10.1016/j.intcom.2010.07.003

Becker, J., Rosemann, M., & Uthmann, C. von. (2000). Guidelines of business process modeling. In Business process management (pp. 30–49). Springer.

Bryl, V. (2009). Supporting the design of socio-technical systems by exploring and evaluating design alternatives. University of Trento, Trento.

Cardoso, E., Almeida, J. P. A., & Guizzardi, G. (2009). Requirements engineering based on business process models: A case study. 2009 13th Enterprise Distributed Object Computing Conference Workshops, 320–327. https://doi.org/10.1109/EDOCW.2009.5331974

Castro, J., Alencar, F., & Cysneiros, G. (2000). Closing the GAP between organizational requirements and object oriented modeling. Journal of the Brazilian Computer Society, 7(1), 05–16. https://doi.org/10.1590/S0104-65002000000200002

Checkland, P. (1999). Systems Thinking, Systems Practice (Hauptbücherregal). John Wiley & Sons.

Checkland, P., & Poulter, J. (2020). Soft systems methodology. In M. Reynolds & H. Sue (Eds.), Systems approaches to making change: A practical guide (pp. 201–253). Springer.

Cherns, A. (1987). Principles of Sociotechnical Design Revisted. Human Relations, 40(3), 153–162.

Clegg, C. W. (2000). Sociotechnical principles for system design. Applied Ergonomics, 31(5), 463–477. https://doi.org/10.1016/S0003-6870(00)00009-0

Coelho, J. (2005). BPM and Continuous Improvement. Search of BPM Excellence: Straight from the Thought Leaders, 119–130.

Fischer, G., & Herrmann, T. (2011). Socio-technical systems: A meta-design perspective. International Journal of Sociotechnology and Knowledge Development (IJSKD), 3(1), 1–33.

Fischer, G., & Herrmann, T. (2015). Meta-design: Transforming and enriching the design and use of socio-technical systems. In Designing Socially Embedded Technologies in the Real-World (pp. 79–109). Springer London.

Fowler, M. (2004). UML distilled: A brief guide to the standard object modeling language (3rd ed). Addison-Wesley.

Goedicke, M., & Herrmann, T. (2007). A case for viewpoints and documents. Monterey Workshop, 62–84.

Grau, G., Franch, X., & Maiden, N. A. M. (2008). PRiM: An i*-based process reengineering method for information systems specification. Information and Software Technology, 25.

Herrmann, T. (2009). Systems design with the socio-technical walkthrough. In Handbook of research on socio-technical design and social networking systems (pp. 336–351). Hershey: Idea Group Publishing.

Herrmann, T., Ackermann, M. S., Goggins, S. P., Stary, C., & Prilla, M. (2017). Designing Health Care That Works – Socio-technical Conclusions. In Designing Healthcare That Works. A Socio-technical Approach. (pp. 187–203). Academic Press.

Herrmann, T., & Hoffmann, M. (2005). The Metamorphoses of Workflow Projects in their Early Stages. Computer Supported Cooperative Work, 14(5), 399–432.

Herrmann, T., Hoffmann, M., Kunau, G., & Loser, K.-U. (2004). A Modeling Method for the Development of Groupware Applications as Socio-Technical Systems. Behaviour & Information Technology, 23(2), 119–135.

Herrmann, T., Jahnke, I., & Nolte, A. (2021). A problem-based approach to the advancement of heuristics for socio-technical evaluation. Behaviour & Information Technology, 1–23. https://doi.org/10.1080/0144929X.2021.1972157

Jones, S., & Maiden, N. A. M. (2004). RESCUE: An integrated method for specifying requirements for complex socio-technical systems. In J. S. Mate & A. Silva (Eds.), Requirements Engineering for Sociotechnical Systems (pp. 245–265). Idea Group. http://www.sociotech-lit.de/JoMa04-Rim.pdf

Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., & Ghose, A. (2006). Combining i* and BPMN for Business Process Model Lifecycle Management. In J. Eder & S. Dustdar (Eds.), Business Process Management Workshops (Vol. 4103, pp. 416–427). Springer Berlin Heidelberg. https://doi.org/10.1007/11837862_39

Modiriasari, M. (2022). An Improved Method for Socio-Technical Process Design – the Development of Requirements in the Case of Pharmacy Systems. University of Diuisburg - Essen (to appear).

Mumford, E. (1994). New treatments or old remedies: Is business process reengineering really socio-technical design? The Journal of Strategic Information Systems, 3(4), 313–326. https://doi.org/10.1016/0963-8687(94)90036-1

Mumford, E. (1995). Effective systems design and requirements analysis: The ETHICS approach. Macmillan.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: A roadmap. Proceedings of the Conference on the Future of Software Engineering, 35–46.

Recker, J., Indulska, M., Rosemann, M., & Green, P. (2006). How Good Is BPMN Really? Insights From Theory And Practice. 14th European Conference on Information System (ECIS 2006). Gothenburg (Sweden), 1–12.

Salehi, V., Veitch, B., & Smith, D. (2021). Modeling complex socio-technical systems using the FRAM: A literature review. Human Factors and Ergonomics in Manufacturing & Service Industries, 31(1), 118–142. https://doi.org/10.1002/hfm.20874

Santos, E., Castro, J., Sanchez, J., & Pastor, O. (2010). A Goal-Oriented Approach for Variability in BPMN. 17–28.

Sommerville, I. (2007). Software Engineering (8th ed.). Pearson Studium.

Sommerville, I. (2014). Designing for the Don't Cares: A story about a sociotechnical system. https://doi.org/10.48550/ARXIV.1411.3948

Sommerville, I., & Sawyer, P. (1997). RE: a good practice guide. John Wiley & Sons.

Yu, E. (2009). Social Modeling and i*. In A. T. Borgida, V. K. Chaudhri, P. Giorgini, & E. S. Yu (Eds.), Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos (pp. 99–121). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02463-4_7

Yu, E., Giorgini, P., Maiden, N., & Mylopoulos, J. (2011). Social modeling for requirements engineering. MIT press.

Zarour, K., Benmerzoug, D., Guermouche, N., & Drira, K. (2019). A systematic literature review on BPMN extensions. Business Process Management Journal, 26(6), 1473–1503. https://doi.org/10.1108/BPMJ-01-2019-0040