A Coarse-grained Comparison of Modelling Languages for Business Motivation and Intentional Distribution

Rando Tõnisson and Raimundas Matulevičius

University of Tartu, Estonia
randotonisson@gmail.com and rma@ut.ee

Abstract. Goal modelling is an important activity to reason *why* different software decisions are taken, or architecture solutions are implemented. Currently there exist a number of goal-oriented modelling approaches. In this paper we apply the semiotic quality framework to compare quality of the business motivation model (BMM) and *i** modelling languages at the coarse-grained level. The study report on the BMM and *i** language quality and model quality. The study also presents observations on how the BMM and *i** modelling languages could be used to reason on and support construction of the business process model expressed in business processes model and notation.

Keywords: BMM, the *i** framework, SEQUAL, business process management, language quality, model quality

1 Introduction

Goal-oriented modelling is one of the most important research developments in the RE field. During goal modelling one needs to shift the focus from *what* and *how* (i.e., data and processes) as addressed by traditional analysis, to *who* and *why* (i.e., the actors and the goals they wish to achieve). With recent increase of importance of business process management (BPM) and BPM modelling approaches it became necessary to understand the rationale of one or another business process decision or system architecture solution. Therefore, combination of the goal-oriented approaches and business process modelling becomes rather a necessity than an option.

On one hand, *business motivation model* (BMM) supports identifying factors that (*i*) motivate business plans and (*ii*) define means of how these plans are related to each business goal [15]. On another hand traditional goal-oriented modelling addresses the early analysis of requirements. But they can also be applied to support and reason on decisions taken during the business process management. Among various goal-oriented approaches, such as GBRAM [2], KAOS [10], Lightswitch [17]), the *i* framework* [19] (and its multiples extensions, such as GRL [7], TROPOS [4], NRF [5]) is one of the most widely used goal modelling language [3]. The *i** language supports systematic understanding of the system actors, their goals and reasoning of achieving these goals by the available means.

In this paper we apply the semiotic quality framework (SEQUAL) [9] to compare both modelling, i.e. BMM and *i**, languages. The goal of this study is threefold. It includes (*i*) comparison of BMM and *i** language quality, (*ii*) comparison of BMM and *i** model quality, and (*iii*) identification of the mappings of the BMM and *i** models to the business process model and notation [16].

The remaining of this paper is structured as follows: in Section 2 we will present the SEQUAL framework, BMM and *i** modelling languages. In Section 3 we compare both languages using the SEQUAL guidelines. Section 4 presents the related work. Finally, in Section 5 we conclude our study.

2 Background

This section introduces the semiotic quality framework used to guide analysis of the selected modelling languages. Next, the major principles of BMM and *i** are presented.

2.1 SEQUAL

The semiotic quality framework (SEQUAL) [9] argues for the constructivistic world-view that recognises model creation as part of a dialog between participants whose knowledge changes as the process takes place. It separates between several quality types to assess information systems models. For instance the *physical quality* addresses how tools are used for making the model. *Empirical quality* deals with error frequencies when reading or writing model, as well as coding and ergonomics when using modelling tools. *Semantic* and *syntactic quality* types are used to evaluate the semantic/syntactic correctness and completeness of the model. *Pragmatic* quality deals with model understandability and interpretation both by the modelling tools and the audience (i.e., users) of the model. Social quality seeks agreement among the participants' interpretation of the model. Finally, the *deontic* quality type focuses on the financial aspects and the goals of the model.

The SEQUAL framework also separates among the six appropriateness types [9] to evaluate quality of the modelling language. *Domain appropriateness* (*DA*) evaluates the basics of the modelling language and how useful they are to use. *Comprehensibility appropriateness* (*CA*) is used to understand the social interpretations. *Participant appropriateness* (*PA*) is used to evaluate the participants' knowledge on the language. *Modeller appropriateness* (*MA*) evaluates the knowledge of the modeller with the modelling languages. *Tool appropriateness* (*TA*) is used to evaluate the languages tools. Finally, *organisational appropriateness* (*OA*) shows the relations between the language and the organization using it for work.

In Section 3 we will apply the SEQUAL framework to assess quality of BMM and *i** languages and models.

2.2 Business Motivation Model

Business motivation model (BMM, version 1.3, 2015) [15] contains a set of built-in concepts, which define the elements of business plans. All the elements of BMM are developed from a business perspective. The visual constructs are summarised in Fig.

1. The major constructs include *Ends*, *Means*, *Influencers*, and *Assessments* areas. The *Ends* characterise the goal that the business wants to succeed in, and the *Means* are the processes that are employed to get to achieve this goal. *Influencers* are the cause to do something in the business. They shape the elements in the business plan and also are the base for *Assessments* that impact both the *Ends* and the *Means*. All of those are related to each other by some fundamental questions, which the BMM user needs to answer – "what is needed to achieve in order to achieve what the business wants to achieve?" (answered by completing the *Means*) and "why does each element of the business plan exist?" (answered by analysing the *Ends*, i.e., business goals).

There are three concepts are defined as external BMM references. Firstly, *organization unit* plays participates in defining boundaries of the enterprise being modelled. This means, organisational unit is used to express *ends*, *means*, *assessments*, *influencers* and *strategies*. Secondly, *business process* is a part of *courses of action*. It provides the steps, sequences, structure, interactions, and connections to events that are part of the process. Hence the *organisational unit* is also responsible for the business processes. Finally, *business rules* provide a specific, practicable way to implement *business policies*. Business rules are derived from *business policy*, they may guide the policy and may affect the *tactics*.

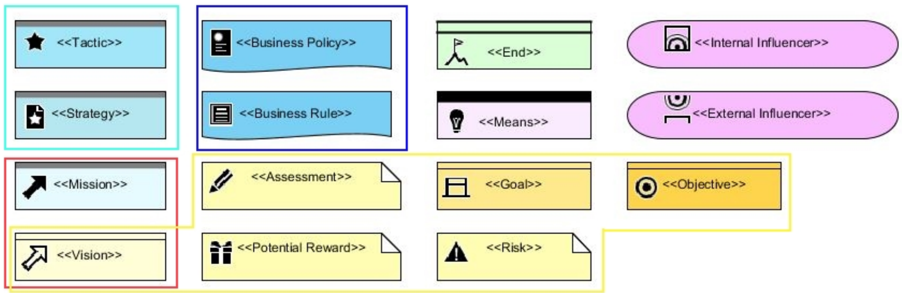

Fig. 1. BMM constructs

Fig. 2 presents a simple BMM model. Here the *assessment* indicates that Warehouse is not suited for delivery. Therefore the Warehouse needs rework to support delivery as indicated by the *internal influencer* construct. Then the argumentation is moved to the means section, where *mission* for warehouse reworked to support delivery contribute to *vision* (on the *ends* area) of warehouse suited for delivery. Similarly the *tactics* to hire a construction company and to plan the rework are defined to establish a *strategy* for reaching the *goal* of warehouse rework completed.

In the current example the *organization unit* is the retail chains management. The *business process* is given in Fig. 6. Finally the *business rules* are not explicitly defined, however, for example, the retail chain management will have to respect the country laws.

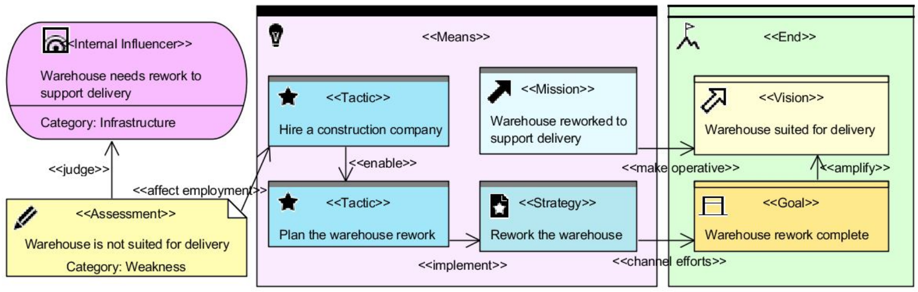

Fig. 2. BMM constructs

2.3 The *i** Framework

The *i** framework uses two diagram types to separate between two levels of abstractions [19]: (*i*) strategic dependency model (i.e., intentional level) to identify actors treating them as the "black boxes", and to define actor dependencies, and (*ii*) strategic rationale model (i.e., rational level) to define internal rationale and intentions of each actor. Language constructs are listed in Fig. 3.

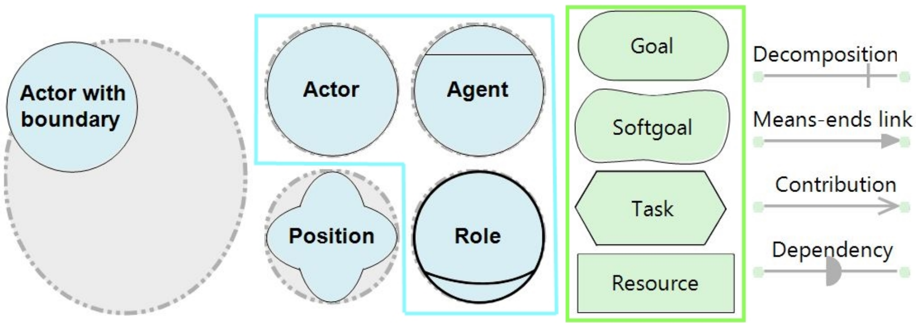

Fig. 3. The *i** constructs

In the strategic dependency model, an *actor* describes an entity that has strategic goals and intentions within the system or within the organisational setting. *Dependency* between two actors indicates that one actor (the *depender*) depends for some reason (*dependum*, expressed using goal, softgoal, task, or resource construct) on another actor (the *dependee*).

In the strategic rationale model, an *actor* is a holder of intentions and characterises active entities, who want goals to be achieved, tasks to be performed, resources to be available, and softgoals to be "satisficed". A *goal* is a condition or state of affairs that the stakeholders would like to achieve. Like a goal, a *softgoal* is a condition that the stakeholder wants to achieve, but there are no clear-cut criteria to determine whether this condition is achieved. A *task* describes a particular way of doing something. A *resource* is an entity for which the main concern is whether it is available. To combine constructs, relationships, like means-ends, decomposition, and contribution, are applied. *Means-ends* are used to describe how goals are achieved, typically through tasks. A *decomposition* link defines the subcomponents of a task, typically (but not limited to) the subgoals that must be accomplished. A *contribution* describes the impact that one element has on another by design.

Fig. 4 presents a simple *i** strategic dependency model. It includes two actors (i.e., Retail chain management and Construction company) and two dependencies: goal (i.e., Warehouse reworked) and resource (i.e., Funds) dependency. In Fig. 5 the strategic rationale model is represented. To achieve goal Warehouse reworked, one needs to Hire a construction company. The later task is decomposed to a goal (i.e., Warehouse suited for delivery) and a task (i.e., Give funds for the rework). To achieve this goal the Retail chain management depends on a Construction company (for the goal Rework warehouse). However Construction company depends on the Retail chains management for the Funds (i.e., resource dependency).

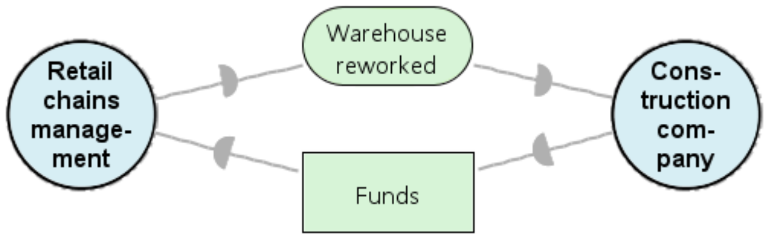

Fig. 4. The i* strategic dependency model

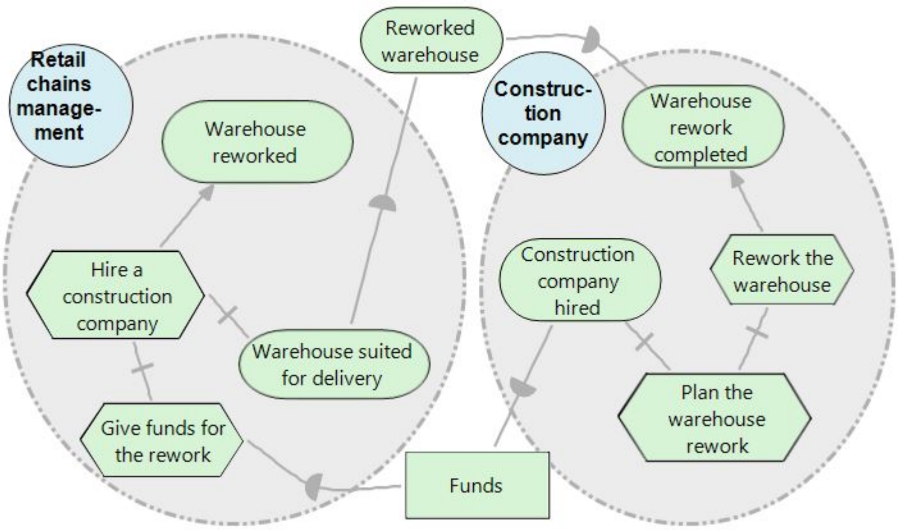

Fig. 5. The *i** strategic rationale model

3 Comparison

In this section we define our research method and overview the validity threats. Then the evaluation results received when comparing the BMM and *i** languages and the BMM and *i** models are presented. Next, the section is continued with the discussion on how the BMM and *i** models are used to support preparation of the business process model.

3.1 Research method

Our research method includes three steps resulting in the coarse-grained analysis of BMM and *i**. Firstly, we analyse the language quality. This means that we select a set of evaluation criteria from [11] that allows comparing some characteristics of the language appropriateness to the *domain* (DA), *comprehensibility* (CA), and *participants* (PA). In the second step we evaluate the BMM and *i** models. The list of evaluation criteria is defined following the SEQUAL framework [9] and contributes to the assessment of the *physical*, *empirical*, *semantic* and *syntactic* quality.

In the third step we map the BMM and *i** models to the business process model. Hence we observe the language concepts and constructs used to reason on and support the construction of the business process model.

3.2 Threats to Validity

Before presenting our findings we will overview some validity threats. The following should be noted:

- *Subjective assessment*. Both language and model evaluation is performed by the authors of this paper. Potentially, the results could be different if a group of respondents were performing these assessments in the controlled experimental settings (e.g., like in [11]. However to mitigate this threat, we were trying to deal with the evaluation criteria which were possible to measure objectively;
- *Few quality characteristics assessed*. This is direct impact from the above issue. We tend to assess a smaller amount of characteristics, but to base our assessment on the objectively measured ones;
- *Limited versions of the languages*. One could argue that in the comparison limited versions of the BMM and, especially, *i** languages are taken. Potentially if comparing various extensions of these languages, the study could result in different observations. However this is not included in the scope of our study;
- *Limited model size*. In the current study were have analysed models of the limited size. Potentially the models could be much larger if modelling some other problems. Our purpose was to provide the illustrative and yet targeted examples of the languages and models use, thus, avoiding the unnecessary details (e.g., duplication of the usage principals) that might be brought by the larger scope cases.

3.3 Languages Evaluation

The evaluation of the BMM and *i** languages is performed along *number of constructs* (contributing to *DA, CA,* and *PA*), *construct semantic similarity* (*DA*), *modelling perspective* (*DA*), *expressive power* (*DA*, *CA*, and *PA*), *well-defined constructs* (*PA* and *CA*), *graphical representation* (*CA*), and *perceptual discriminability* (*CA*) criteria [11]. The summary of this evaluation is presented in Table 1 and discussed below.

Table 1. Evaluation of BMM and *i** languages

Evaluation criteria		**BMM language**	***i** language**
Number of constructs		15 (all constructs)	13 (9 constructs and 4 relationships
Construct similarity		27% of similarity to *i**	31% of similarity to BMM
Modelling perspective	**Behavioural**	Partially	Partially
	Goal and rule	Yes	Yes
	Actor and role	No	Yes
Expressive power	**Behavioural**	3	6
	Goal and rule	9	6
	Actor and role	0	12
Well-defined constructs		Partially	Partially
Graphical representation		79%	65%
Perceptual discriminability		Very low	Low

Number of constructs. The BMM modelling language contains *15* graphical syntax elements, listed in Fig. 1 (only constructs are presented); the *i** language – *13* listed in Fig. 3 (9 constructs, e.g., *actor, goal, task* etc., and 4 relationships*,* e.g., *dependency*, *means-ends*, etc.).

Construct semantic similarity. Although both languages could be used to understand rationale for the business decisions and process management, there are only few semantic concepts, which have similar meaning (see Table 2). For instance, we observe sematic similarities between BMM *vision* and *i* goal*, BMM *goal* and *objective* and *i* softgoal*, BMM *course of action* and *i* task*, and BMM *influence* and *i* actor*. However, this makes only approx. *27%* (received as similar constructs divided by the number of all considered language constructs) of BMM and *31%* of *i** language similarity one to another.

At the same time it should be noted that there are few constructs, which have the same name, but has different semantics. For example, the BMM *end* is something that/where the enterprise wants to be; the *i* end* is something to be achieved (goal), performed (task) or produced (resource). Another example is *means:* in BMM they are steps needed to get to the end, in *i* mean-ends* is a link between two nodes to show how end node will be achieved.

Modelling perspectives. In [9] Krogstie separates between seven modelling perspectives: behavioural, functional, structural, goal and rule, object, communication, and actor and role. Four modelling perspectives – functional, structural, object, and communication – are not supported or their support is very minimal in BMM and *i**. Below we will discuss how the analysed languages support the behavioural, goal and rule, and actor and role perspectives.

Table 2. BMM and *i** construct semantic similarity

BMM constructs		***i** constructs**	
Construct	**Definition**	**Construct**	**Definition**
Vision	An image of the desired end	Goal	A specific condition that actor wants to achieve.
Goal	Narrow version of the vision to make it easier too follow	Softgoal	Narrow version of goal to make it easier to understand.
Objective	Shows the steps towards the goal		
Course of action	Action taken towards achieving the end	Task	Specific way of doing something to achieve the goal
Influencer	Somebody or something to cause the change	Actor	Somebody, who want to achieve the goal(s)

In behavioural perspective the main phenomena are states and transition between states. In BMM the *tactics* and *strategies* (i.e., event element) are used to achieve the *mission* (i.e., the goal element representing state). Similarly, in *i** the performance of *task* contributes to satisfaction of *goal* or *softgoal* (thus expressing the desired change of the state). In general, although in both languages we could observe the implicit elements of the behavioural perspective, this perspective is not explicitly supported by the visual language constructs.

In the goal and rule perspective the main phenomenon are goals (i.e., a condition or state of affairs that is wanted to be achieved) and rules (i.e., something that influences the action of a set of actors). Both BMM and *i** belong to this perspective. For instance, using the BMM language, one needs to define *visions*, which are then divided to goals, and *business rules* and *policies*, which needs to be followed in the organisation. Hence the *means* will show how to achieved the defined *goals*. Similarly in *i**, *goals/softgoals* can be decomposed to lower *goals/subgoals*. Here *tasks* are used to define the rules and conditions to achieve these *goals*.

In the actor and role perspective the main phenomena are actors (or agents) and their roles. BMM does not support this perspective. However, the major principle of the *i** framework is to understand the major system actors (potentially their roles, agents and positions) and to define their dependency relationships, as illustrated in Fig. 5.

Expressive power is defined as the relationship between the number of constructs and the number of modelling perspective [11]. Table 3 shows that BMM has 3 constructs and *i** 3 constructs for the behavioural perspective; and BMM - 9 and *i** - 6 appropriately for the goal and rule perspective. Expressive power also shows a limitation of the language constructs regarding the *well-defined constructs*.

Well-defined constructs mean that constructs have a clear (but possibly informal) semantics. As illustrated in Table 3 both BMM (e.g., *tactics*, *strategy*, and *mission*) and *i** (e.g., *goal*, softgoal, task, and others) constructs could be used for different purposes (i.e., they have various meaning), thus, they could be misinterpreted by various model readers.

Table 3. Expressive Power of the BMM and *i** constructs

Modelling perspective	BMM constructs		The *i* * constructs	
Behavioural	Tactics, strategy, mission	3	Goal, softgoal, task, decomposition, means-ends, contribution	6
Goal and rule	Means, tactics, strategy, mission, vision, goal objective business policy business rule	9	Goal, softgoal, task, decomposition, means-ends, contribution	6
Actor and role	-	-	Actor (also agent, position, and role), goal, softgoal, task, resource, dependency, means-ends, decomposition, and contribution	12

Graphical representation criterion means that modelling language should possess a graphical representation of each construct [9]. The graphical syntax was confronted to the language abstract syntax definitions found in the literature. For instance, the core BMM concepts are defined in [15]. Hence we observe that all graphical constructs listed in Fig. 1, have their corresponding definitions in the BMM abstract syntax (approx. *79%* of BMM concepts). However, in addition, there exist a number of abstract BMM concepts, such as *directive* (generalises *business policy* and *business rule*), *course of action* (generalises *strategy* and *tactics*), *desired result* (generalises *goal* and *objective*), and *potential impact* (generalises *reward* and *risk*), which do not have their explicit graphical representations.

A comprehensive representation of the *i** abstract syntax is given in [3]. Hence, the visual constructs, listed in Fig. 3, have their correspondences in the abstract syntax. This makes approx. *65%* of the defined *i** concepts. But, like in BMM, there exist a number of abstract concepts, such as *intentional elements*, (generalises *dependum* and *intentional elements*), *internal elements* and *dependum* (both generalises *goal*, *softgoal, task* and *resource*), *dependable node* (generalises *actor* and *internal element*), *relationship* (generalises *means-ends*, *decomposition*, and *contribution*), and few others.

Perceptual discriminability. Refers to the ease and accuracy with which symbols can be differentiated from each other [13]. Both BMM and *i** constructs have discriminability limitations. As discussed in [14], the *i** language has similar construct shapes (e.g., actors, agents, and roles are of the same shape) and limitations in relationship discriminability (e.g., contribution are differentiated only by their labels, etc.). Similar limitations are observed regarding the BMM constructs. Basically they all are of the rectangle shape, mostly differentiated only by colours and icons, placed on the top left corner of the rectangle (see Fig. 1).

3.4 Model Evaluation

Table 4 presents the evaluation of the BMM and *i** models. It adapts criteria [9] for *electronically stored* and *reusability* (contributes to *physical quality*), *use of colour, font size* and *construct size* (*empirical quality*), *syntactic invalidity* and *syntactic* *incompleteness* (*syntactic quality*), and *semantic completeness, semantic validity* and *consistency* (*semantic quality*). The model evaluations are discussed below.

Table 4. Evaluation of BMM and *i** models

Evaluation criteria		BMM model	*i** model
Physical quality	**Electronically stored**	YES, Visual Paradigm	Yes, OpenOME
	Reusability	Opening, copying, adapting, etc.	Opening, copying, adapting, etc.
Empirical quality	**Number of colours**	11	4
	Model size	10 constructs (2 compounds)	12 constructs (2 compounds)
	Font size	Arial, 12 size	Arial, 12 size
	Construct size	Different size constructs	Different size constructs
Syntactic quality	**Syntactic invalidity**	Links used to connect constructs	No errors
	Syntactic incompleteness	No errors	No errors
Semantic quality	**Semantic validity**	Valid	Valid
	Semantic completeness	Incomplete	Incomplete
	Consistency	Consistent	Consistent

Physical quality. The BMM model is stored electronically in the Visual Paradigm format. Use of the tool also enables reusability (e.g., opening, copying, adapting, etc.) of previous model components. The *i** model is also stored electronically using the OpenOME tool. Like BMM, the *i** model components can be reused from model to model (e.g., opening, copying, adapting, etc.).

Empirical quality is assessed though the number of colours, model size (number of constructs) font size, and size of objects in the model. As advised in [9], model should potentially include from four to seven *colours*. In addition same colours should not be used for different objects and same objects should not have different colours. In the BMM model 11 different colours are used. Also, some colours used are very similar to one to another (e.g., different shades of yellow are used for *objective, goal and risk and vision*, different shades of blue are used for *mission, strategy* and *business rules*). In the *i** model four different colours are used.

The *font style* and *size* should be easy to read for most people [9]. Both in the BMM model and in the *i** model we have applied the Arial style, 12 size font.

Finally, the *element size* and *orientation* of the elements placed in diagrams could be used to make emphasis or indicate reading start points. Also the objects in the middle are mostly noticed first by the model readers. In the BMM model the first two elements seen on the model are *means* and *end*. They are way bigger in size than other objects and even contain smaller objects inside. This can be both, good and bad. Good in the aspect that the goal of the model is in the end object and the way to achieve it is defined in the means. But the cause of the modelling is defined in the influencer nodes that are smaller and will not be seen at first. The cause or motivation to do something is the main aspect of the model and with such visuals, it may not look like it. In the *i** model the size of the different actors differs a lot. This is caused by the number of their internal elements. This can be considered good, as the management is the most important actor, that controls the flow of the work and has dependencies on all other actors and all the other actors depend on the management. The sizes for other actors and elements are quite similar.

Syntactic quality is the correspondence between the language used in the model and the language defined in the concepts of the modelling language [9]. Two kinds of errors – syntactic invalidity and syntactic incompleteness – could be observed. *Syntactic invalidly* means that model includes elements that are not part of the modelling language. *Syntactic incompleteness* means that the model does not have constructs, which are required by the used language.

All the constructs used in the BMM model are a part of the BMM language, thus, this makes the model syntactically correct. However the model includes the links between the constructs; these links are not the part of the BMM abstract syntax [15], however it is not possible to combine the language constructs without them. The model is also syntactically complete as it has all the four main concepts including *influencers*, *assessments*, *means*, and *end*.

All the constructs used in the *i** models are the part of the modelling language, thus making the model syntactically correct. The *i** model is also syntactically complete as it includes all syntactic constructs in the way as defined in its abstract syntax.

Semantic quality is the correspondence between the model and the modelling domain [9]. While a lot of properties can be defined for semantic quality, the two main ones are semantic validity and semantic completeness. To be valid, the statements in the model must be defined correctly and be relevant to the domain. To be complete, the model has to contain all the statements that are correct and relevant for the problem. To be valid and complete, the model has to be consistent. Consistence requires the objects not to conflict with each other.

The BMM and *i** models are semantically valid as their all the statements are correct and carry over information about the warehouse construction problem. However, both models are not semantically complete, because additional information about the problem could potentially be introduced. For example, it is possible to define more business rules and policies about the working conditions, required worker expertise, etc. Both models do not contain inconsistences.

3.5 Creating Business Process Model

Fig. 6 presents the business process model expressed in business process model and notation (BPMN, version 2) [16]. This BPMN model presents a process between two *pools* Retail chain management and Construction company. The process starts with the *start event*, which defines the need for warehouse reworked to support delivery. The retail chain management needs to hire a construction company. Once this is done (see *state* Construction company hired), construction company plans the warehouse rework, reworks the warehouse and hands it (see *task* hand over the warehouse) to the retail chain management. Once the warehouse suited for delivery is received at the retail chain management, the process is over.

The creation of the business process is supported by the BMM (see Fig. 2) and *i** (see Fig. 4 and 5) models. In other works the BMM and *i** models provide the rationale for different elements of the BPMN model. This relationship is illustrated in Table 5 and discussed below.

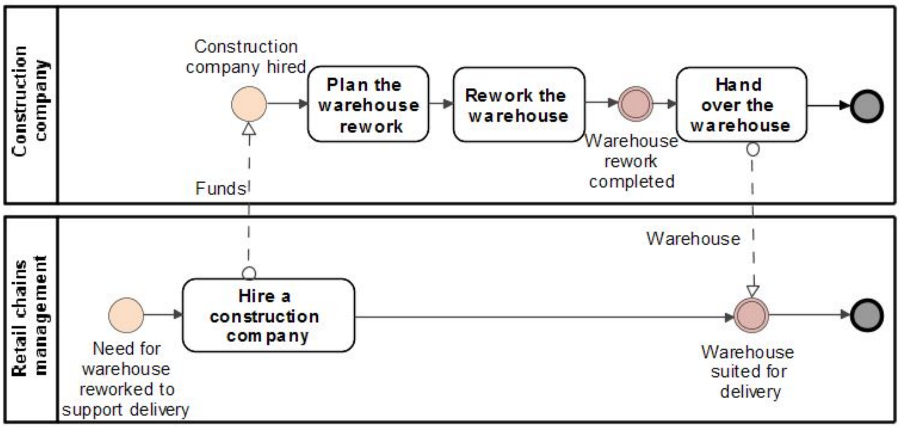

Fig. 6. The BPMN model

Table 5. Mappings among BMM, *i** and BPMN models

	BMM	BPMN	*i**
1	Deriving from *organization unit* [Retail chains management]	*Pool* [Retail chains management]	*Actor* [Retail chains management]
2	Deriving from *tactic* [Hire a construction company]	*Pool* [Construction company]	*Actor* [Construction company]
3	*Influencer* [Warehouse needs rework to support delivery]	*Start event* [Need for warehouse reworked to support delivery]	*Goal* [Warehouse reworked]
4	*Goal* [Warehouse rework completed]	*Intermediate event* [Warehouse rework completed]	*Goal* [Warehouse rework completed]
5	*Vision* [Warehouse suited for delivery]	*Intermediate event* [Warehouse suited for delivery]	*Goal* [Warehouse suited for delivery]
6	*Tactic* [Hire a construction company]	*Task* [Hire a construction company]	*Task* [Hire a construction company]
7	*Tactic* [Plan the warehouse rework]	*Task* [Plan the warehouse rework]	*Task* [Plan the warehouse rework]
8	*Strategy* [Rework the warehouse]	*Task* [Rework the warehouse]	*Task* [Rework the warehouse]
9	*Not defined*	*Message flow* [Funds]	*Resource dependency* [Funds]
10	*Not defined*	*Start event* [Construction company hired]	*Goal* [Construction company hired]
11	*Not defined*	*Task [Hand over the warehouse]*	*Goal dependency* [Reworked warehouse]
12	*Not defined*	*Message flow* [Warehouse]	*Goal dependency* [Reworked warehouse]

BMM and BPMN. The closest similarity between the BMM and BPMN meaning is the mappings between:

- the BMM *ends* (*goal* and *vision*) and BPMN *events*. For instance the BMM *goal* (i.e., Warehouse rework completed) and *vision* (i.e. Warehouse suited for delivery) are mapped to the appropriate BPMN *events*.
- the BMM *means* (*tactics* and *strategy*) and BPMN *tasks*. For instance, the BMM *tactics* (i.e., Hire a construction company and Plan the warehouse) and BMM *strategy* (i.e., Rework the warehouse) are mapped to the appropriate BPMN *tasks*.

The rest of the BPMN model at large depends on the interpretation of the BMM element. For example, both BPMN *pools* do not have the direct mapping to the BMM model, however, they can be captured by interpreting the BMM *organisational unit* (i.e., Retail chains management) and, potentially, other BMM elements (i.e., *tactics* to capture Construction company). It is also important to not that some elements of the BMM model are not captured in the BPMN model; for example, the meaning of *mission* and *assessment* are not expressed in the business process.

***i** and BPMN** models has rather very close meaning as illustrated in Table 5. For instance:

- BPMN *pools* (i.e., Retail chains management and Construction company) could be explained with the *i* actors*;
- BPMN *states* (i.e., Warehouse rework completed and Warehouse suited for delivery) are derived from the *i* goals*;
- BPMN *tasks* (i.e., Hire a construction company, Plan the warehouse and Rework the warehouse) has the correspondence to the *i* tasks*.

Finally the *i** could reason the BPMN *message flows* between the pools. For instance the *i* dependency* relationships (i.e., *resource dependency* Funds and *goal dependency* Warehouse rework) helps to capture and explain why message flows of Funds and Warehouse are introduced in the BPMN model. Basically all the elements from the *i** model are mapped to or used to reason on the elements defined in the BPMN model.

4 Related Work

Literature includes a number of studies reporting the evaluation of the goal modelling languages. Ayala *at al*. have compared three languages (*i**, GRL and Tropos) to develop a reference model to conceptualise the *i**-based oriented languages [3]. Elsewhere, in [11] the experimental comparison of the *i** and KAOS languages indicated weak and strong characteristics of the goal oriented languages. In [6] it was studied how requirements engineering experts understand a language for modelling business goals and enterprise architectures (AMOR).

There exist few studies, which consider the fine-grained quality of goal modelling languages. For instance, in [14] the visual quality of the *i** notations is thoroughly studied according to the principles of language notation design [13]. The approach [1] of the unified enterprise modelling language (UEML) is used to understand and compare the ontological semantics of the GRL and KAOS languages [12]. Similarly, in [18] the ontological semantics of BMM and *i** is analysed using the UEML approach. Although the further model mapping between the BMM, *i**, and other modelling languages (e.g., GRL, KAOS, BPMN and other) could be performed using the UEML approach, this was left as the future study. In our study, however, we focus on the coarse-grained comparison of the BMM and *i** languages.

Koliadis *et al* introduces the way to capture changes between the *i** and BPMN models [8]. Like observed in our study, they introduce the mapping relationships between the *i* actors* and BPMN *pool*, *i* dependencies* and BPMN *message flows*, *i* tasks* and BPMN *tasks*. Differently from our study, the authors argue on the link between the *i* goals* and BPMN *tasks*; in our case it is suggested to link *goals* to the BPMN *states*.

6 Conclusion

This paper presents a coarse-grained evaluation of two modelling languages – BMM and *i** and their models. Next it considered how both languages could be used to reason on the preparation of the BPMN model. The study results in the following conclusion:

- BMM and *i** have similarities regarding the *ends* and *means*. This allows modellers to express phenomenon in the *goal and rule* perspective. However, the BMM *means* (i.e., *tactics, strategy,* potentially *business policy* and *business rule*) and *ends* (i.e., *vision*, *goal* and *objective*) extend the meaning of *i* means* (i.e., *task*) and *ends* (i.e., *goal*). However, in addition the *i** language contains constructs to express organizational actors and relationships (in terms of the actor *dependencies*) between them. This means that modellers can represent phenomena in the *actor and rule* perspective using *i** but not BMM.
- Both models are rather of the equal quality. A slightly better quality could be observed for the *i** model, which uses less colors than the BMM model and contains no syntactic validity errors. However, one should also note that much also depends on the selected modelling tools.
- As observed the *i** framework suggests better means to support decision rationale when constructing the BPMN process model. It covers the complete set of the BPMN constructs, including such mappings as *i* actor* and BPMN *pools*, *i* goals* and BPMN *states*, *i* tasks* and BPMN *tasks*. Also it provides guidelines to map and reason on the BPMN *message flows* through the *i* dependencies*.

The study opens several future research directions. The further research is needed to understand how to use the *best* features of both languages; for instance, the *i** framework could be expanded with the additional constructs for *means* and *ends* taken from the BMM language. This would, also, lead to understanding of the new mappings to reason on the process models. However, these study directions also require the fine-grained investigation of the language quality.

References

1. Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevičius, R., Opdahl, A.L., Panetto, H. & Verdecho, M.J. (2010). The unified enterprise modelling language – overview and further work. Computers in Industry, 61(2), 99–111.
2. Anton, A.I.: Goal-based Requirements Analysis. In: Proc. of the 2nd Int. Conference on Requirements Engineering (ICRE '96). pp. 136 – 144. IEEE Computer Society (1996)
3. Ayala, C.P., Cares, C., Carvallo, J.P., Grau, G., Haya, M., Salazar, G., Franch, X., Mayol, E., Quer, C.: A Comparative Analysis of i*-based Agent-oriented Modeling Languages. In: Proceedings of the international workshop on agent-oriented software development methodology (AOSDM'2005) (2005)
4. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An Agent-oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems 8(3), 203 – 236 (2005)
5. Chung KL, Nixon B, Mylopoulos J, Yu E, (2000) Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Boston.
6. Engelsman W., Wieringa R. Understandability of Goal Concepts by Requirements Engineering Experts, Advances in Conceptual Modeling, 2014, pp. 97-106
7. ITU: Recommendation Z.151 (GRL) – Version 3.0 (September 2003)
8. Koliadis G., Vranesevic A., Bhuiyan M., Krishna A., Ghose A., Combining *i** and BPMN for Business Process Model Lifecycle Management, Business Process Management Workshops, 2006, pp 416-427
9. Krogstie, J.: (2012) Model-Based Development and Evolution of Information Systems. London: Springer-Verlag
10. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley and Sons (2009)
11. Matulevičius R., Heymans P., Comparing Goal Modelling Languages: an Experiment, Requirements Engineering: Foundation for Software Quality, 2007, pp 18-32
12. Matulevičius R., Heymans P., Opdahl A. L., Comparing GRL and KAOS using the UEML Approach, IESA 2007, pp. 77-88
13. Moody DL (2009) The ''Physics'' of notations: towards a scientific basis for constructing visual notations in software engineering. IEEE Transaction of Software Engineering 35(5): 756–777.
14. Moody D. L., Heymans P., Matulevičius R., Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation, Requirements Engineering, 2010, 15, pp 141- 175.
15. OMG, Business Motivation Model (BMM, version 1.3), 2015, URL: http://www.omg.org/spec/BMM/1.3/ (last check: 24.04.2016)
16. OMG, Business Process Model and Notation (BPMN, version 2.0), 2011, URL: http://www.omg.org/spec/BPMN/2.0/ (last visited 24.04.2016)
17. Regev, G.: A Systemic Paradigm for Early IT System Requirements Based on Regulation Principles: The Lightswitch Approach. Ph.D. thesis, Swiss Federal Institute of Technology (EPFL) (2003)
18. Tu, C. (2007). Ontological evaluation of BMM and i* with the UEML approach. Master thesis, University of Namur.
19. Yu, E.: Towards Modelling and Reasoning Support for Early-phase Requirements Engineering. In: Proceedings of the 3rd IEEE international conference on requirements engineering (RE'97) (1996)