

A Technique for Annotating EA Information Models with Goals

Sabine Buckl, Florian Matthes and Christian M. Schweda

Chair for Software Engineering of Business Information Systems (sebis),
Technische Universität München,
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl,matthes,schweda}@in.tum.de
<http://wwwmatthes.in.tum.de>

Abstract. Many of today's enterprises experience the need to establish and conduct management processes to ensure closely alignment between business and IT. Enterprise architecture (EA) management provides a model-based approach to understand and evolve the complex dependencies between the enterprise constituents, as e.g. business processes and business applications. In recent years the understanding of EA management in literature and in practice has converged, but up to this point no commonly accepted standard information model for EA management nor a standard set of goals verifying the overall objective of business/IT-alignment have been devised. Grounded in indications that such models and goals are highly enterprise-specific, this paper presents a method for flexible combining EA-relevant goals and EA information models to optimally support EA management in a using company.

Key words: Enterprise architecture, Enterprise architecture management, Goal modeling, Modeling language, Goal question metric

1 Introduction and motivation

Many of todays large organization, first and foremost the globally acting enterprises, have to face the challenges of managing their business and IT structures as a whole. These structures commonly form highly interwoven systems, which mostly have developed for a long period and become increasingly complex trough mergers, business process re-organizations or technological shifts. Understanding the make-up of systems as well as of their interdependencies as *architecture*, it is sensible to allude to the aforementioned systems of constituents of the *enterprise architecture* (EA). Over the last years, practitioners and researchers have developed a variety of approaches for how to perform EA management and as one central task of this management process on how to document the EA. While in recent days some convergence in the understanding of EA as well as in the terms used to describe EAs can be diagnosed, the field is still far from having a common terminology or description technique. The latter especially applies in the context of *EA management goals*, i.e. the goals that an organization seeks to

attain by performing EA management. Some researchers as Aier et al. (cf. [1]) or Buckl et al. (cf. [2]) even challenge the hypothesis that a common description technique or a "standard" set of EA management goals can be devised. In contrast, they expect both techniques and goals to be highly organization-specific artifacts, accounting for the diversity of organizations as well as of their high level organizational goals and structure.

While one could expect that the aforementioned diversity is well-accounted by the different EA management approaches and modeling techniques, a survey on the current literature shows a slightly different image. Foreclosing the results of our discussion in Section 2 on the state-of-the-art in these areas, we summarize the situations as follows. EA description techniques are either

- developed without explicit reference to the concrete goals they are aiming at, most commonly resorting to abstract and general goals as "providing transparency", or
- tailored via distinct model concepts, as e.g. classes, to a narrow set of concrete goals without giving indications on how to adapt the concepts to an organization-specific utilization context.

Many EA literature does not specifically account for concrete EA management goals, but stays on the abstract and general level, as indicated above. If in contrast concrete goals are described, they are either given textually and without explicit reference to the EA model concepts, that they allude to, or directly incorporated into the corresponding modeling technique, e.g. via properties and KPIs. The second case nevertheless neglects the fact that the modeling technique might have to be adapted to be capable of satisfying the organization-specific information demands. From this, we derive the research gap that this paper aims to address. Summarized as a research question, the gap reads as follows:

How can EA management goals be described in a manner that they can be incorporated into flexible EA modeling techniques?

The rest of the article is dedicated to answering the above question. Preparing our discussions, we revisit the state-of-the-art in EA modeling and EA management goal modeling in Section 2. Thereby, we seek to illustrate that the research gap is an actual one, but also provide a conceptual basis on which flexible EA modeling techniques are grounded. In Section 3, we present a technique that allows to model EA management goals as part of an EA information model. In this section, we further provide accompanying examples for the application of the technique. Concluding Section 4 summarizes critical points of the technique and provides an outlook on future research directions in this context.

2 State-of-the-art in EA modeling and Related Work

We approach the topic of modeling in the context of EAs from two different perspectives. At first, the *Goal Question Metric Approach* of Basili et al. (cf. e.g. [3]) is introduced in Section 2.1 as a conceptual framework for reasoning on

goals as well as the measurement of their achievement. Against that background, we secondly revisit the modeling techniques put forward by selected state-of-the-art EA management approaches in respect to the underlying information models as well as on the means for making EA management goals explicit in Section 2.2. Section 2.3 summarizes the current situation in literature and reifies the research question towards the description of the corresponding research gap.

2.1 The Goal Question Metric Approach

In [3] Basili et al. summarize the basic ideas of the Goal Question Metric (GQM) approach, developed some years earlier by Basili. Central to the approach is the understanding that goals, i.e. what is to be achieved, and metrics, measuring the achievement are closely related. Put in other words, goals are conceptual level entities, that are complemented by metrics on the quantitative level. The measurement of metrics is thereby understood as a mechanism to "aid in answering a variety of questions associated with the enactment of any [software] process". Later in [3] Basili et al. continue with "Measurement [...] helps to assess [the] progress [of a process], to take corrective action based on this assessment, and to evaluate the impact of such action". While the former quotations originate from the context of software development projects and processes, they also sensibly apply to many other processes targeting the creation (development) of a specific artifact. In this sense, we apply the basic notion of the GQM approach to EA management, which in line with e.g. van der Raadt and van Vliet (cf. [4]) can be understood as a management function aiming at the development of the EA.

In the GQM approach (cf. [3]), Basili et al. not only advocate for a strong linkage between goals and metrics but further emphasize that metrics, in order to be both sensible and applicable, must be "defined in a top-down fashion, [i.e.] focused based on goals and models". At this point, we should clarify the understanding of metrics in the GQM approach: a metric is quantitative information on a property of an object derived either via an *objective* measurement procedure or *subjectively*, e.g. by expert assessment. We will see below that according to this broad understanding metrics are widely used in EA management, although they are commonly not explicitly alluded to with the term "metric" but by the term "analysis". This term, while obviously correct, is usually used to mitigate some reservations in respect to the quantitative and objective nature of metrics. As shown later such reservations also exist in the context of EA management. To proactively dispel corresponding concerns, we provide the following working definition for the term "metric" based on the understanding of Basili et al.:

A metric is at-least ordinally scaled information on a property of an object derived via an objective or a subject-dependent measurement procedure.

A side note has to be added to former definition concern the term "at-least ordinally scaled". By explicitly demanding this relatively lax *level of measurement* (for details on levels of measurement see Krantz et al. [5]), we allow e.g. that a

property could be measured by a metric that evaluates to "high", "medium", or "low".

Coming back to the GQM approach of Basili et al. (cf. [3]), a last but highly important concept remains to be introduced: the *question*. Where the goal resides on the conceptual level stating what is to be achieved and the metric provides quantitative information on object properties, the questions comprise the "operational level" mediating between the two aforementioned levels. Put in words of Basili, questions "characterize the object of measurement" but do neither give measurement prescriptions nor indicate an intended achievement. In this sense, the questions establish the link between the goals and the objects that are affected by this goal, i.e. concretize the goal dependencies of the developed artifact. The questions further provide an abstraction from the concrete metrics used to quantify the achievement of goals.

2.2 EA Information and Goal Modeling

The Open Group Architecture Framework (TOGAF) [6] provides since its most recent version 9.0 the "Enterprise Content Metamodel" that describes the core classes, properties and relationships that make up an EA model. The content metamodel thereby puts strong emphasis on holistically covering structural aspects of the EA and is further designed as largely monolithic, i.e. should be used as a whole. A possible exception to latter design principle are the "extensions" that are described as part of TOGAF. These extensions define content metamodel fragments that can be added but do not have to be used. Two of these extensions, the so called "governance extension" and "motivation extension", are concerned with the aspect of goal modeling. Put in more detail, the extension introduce the concepts **Goal**, **Objective** and **Measure**, which are related to each other. These concepts are conversely not equipped with properties other than **name** such that only a statement as "the goal of *increased business continuity* effects (over the measure *availability*) the business service *settle credit card payments*". More detailed modeling of goals is not supported and the goals, more precisely their corresponding measures, cannot be linked to properties of the affected model concept, as **business service**.

The research group of Winter at the university of St. Gallen defines the "core business metamodel" (CBM) as basic model for EA modeling (cf. [7, 8]). The CBM defines classes and relationships useful for describing the structure of an EA. The CBM is further designed as one comprehensive model, although recent publications concede (cf. [1, 9]) that a demand to adapt the model to the specific requirements of a using organization exists. When it comes to the operationalization of goals into the model, three concepts **Strategic goal**, **Success factor**, and **Performance indicator** as well as the complementing relationships are introduced. The performance indicators can further be related to any type of reference object in the EA, but do conversely not provide means to express that one of the reference object's properties relates to the indicator. This may be ascribed to the fact that the CBM does not define properties, but this limit also applies to models that are derived from the CBM via adding properties, which is

according to Kurpjuweit [8] a well-established technique for adapting the CBM. Summarizing, the model allows to relate goals to the underlying success factors as well as to operationalizing performance indicators. In contrast, the goals cannot be linked to the architectural properties in the model.

Niemann gives in [10] information models for describing different architectural layers in the EA, namely the "business", the "application", and the "system layer". On all these layers, core classes and relationships for describing the corresponding architectural concepts are introduced. Properties further describing the classes or the underlying architectural concepts, respectively, are only supplied for a few classes in the information models. Goals, more precisely EA management goals, are not part of the information model, while "business goals" can be described as part of the business layer. Complementing the information model, Niemann provides a plethora of analysis techniques that can be used to measure the attainment of different EA management goals. Most of the analysis techniques are thereby described textually with possible exemplification along graphical models of exemplary EAs. For a few analysis techniques, Niemann provides mathematical equations for deriving performance indicators from properties of the overall EA or the application landscape, respectively. For the topic of cost calculation, Niemann gives a basic economic equation summing up the different types of annual costs associated with an application system and a yearly (linear) deduction of investment costs. These costs are further mirrored as properties in the corresponding information model. When it conversely comes to other performance indicators, Niemann does not supply a link between the equations and the information model.

The architecture method "Quasar Enterprise" described by Engels et al. in [11] gives different information models describing parts of the overall EA. These information models consist of classes and corresponding relationships, but do not provide information on architectural properties that might be of relevance. **Business Goals** are modeled in Quasar Enterprise as part of the the strategy modeling and are interrelated to **Business Services** as part of the logical architecture of an application landscape. Further details on how to measure, to which degree a goal is attained, are neither given in the architecture method nor in the information models. Conversely, different techniques for analyzing the application landscape in respect to general quality measures are briefly sketched in the method. As an example, the quality measure "purity of domains" can be seen. This measure counts to which extent the current **Business Applications** supply **Business Services** for more than one future, i.e. target, **Business Domain**. While a relationship between the information model classes and the corresponding measurement rule is established textually, the information model does not reflect such calculations.

Gringel and Postina, as members of the research group of Appelrath at the OFFIS in Oldenburg, describe in [12] a reification of the architecture quality measures put forward in Quasar Enterprise. Put in more detail, they provide an information model adapted from the model of the Quasar Enterprise method complemented with a set of equations on how to derive values for corresponding

quality measures, as e.g. "purity of domains". The correspondence between the equations on the one hand and the information model concepts on the other hand is established via names, i.e. properties and concepts are named equally in both representations. The work of Gringel and Postina integrates well into other research results of the group of Appelrath as presented by Addicks and Steffens in [13] as well as by Addicks and Gringel in [14]. In the later publication, different "key figures", i.e. indicators, and their defining equations are presented. The equations nevertheless are not complemented with corresponding information models, such a potential user of the indicators might have difficulties to derive the architectural model, that the indicators are built upon.

Lankhorst et al. describe in [15] both a modeling language for EAs and complementing analysis techniques. The information model underlying the modeling language defines the classes and relationships of the architectural elements, but do not specify properties of the corresponding classes. In the analysis techniques, architectural models describing the structure of the EA are augmented with quantitative information, e.g. on "service execution times" or "interarrival times". These augmentations nevertheless take place on instance level, i.e. a concrete service is supplied with such information, while the class **Service** in the information model does not specify corresponding properties. Complementing the augmentations with quantitative information, expressions for deriving certain property values are supplied in mathematical equations.

In [16] Lankes and Schweda describe two "information model patterns" targeting the goal of *business continuity* by analyzing the property of availability. The information model patterns pick up the notion of the EA management pattern as introduced by Buckl et al. in [2] as building blocks for a flexible EA description technique. The complementing analysis proposed by Lankes and Schweda explores how failures of single **Business Applications** propagate through the EA ultimately rendering the execution of one or more **Business Processes** impossible. The information model contains the basic classes and relationships for describing the EA structure on which the analysis is performed. Complementing, the models are augmented with properties, as e.g. **failureProbability**, establishing the link to the corresponding goal. These properties are *derived* properties, for which the information model further supplies derivation rules formulated in the Object Constraint Language (OCL) [17] as well as in mathematical expressions. For the specific goal of business continuity and the selected architectural description language, the information model patterns of Lankes and Schweda [16] achieve a strong linkage, although the relationship between the goal and the underlying performance indicators are only described textually.

Johnson and Ekstedt provide in [18] a collection of information models of which each reflects a specific *viewpoint* on the overall EA. These information models introduce classes and relationships for describing a specific part of the EA, but do not supply properties for further specifying the corresponding instances. A dedicated "goal viewpoint" introduces the concept of the **Goal**, which can participate in a goal-hierarchy and can be linked to **Initiatives** for pur-

suings the goal as well as to **Problems** hindering the goal’s achievement. Further relationships from the goal class are not provided, although the work introduces another modeling language, namely the *influence diagrams* specifically dedicated to goal modeling. An influence diagram is used to relate the central property of a goal to the architectural properties, which define this central property. Exemplifying this, we revisit the influence diagram defining what ”performance” is meant to be composed of. The diagram makes explicit that ”response time”, ”throughput” and ”scalability” are definitorial for ”performance”, which conversely is central to any goal targeting the achievement of specific performance characteristics.

2.3 Summarizing the research gap

Reflecting the plurality of models for EAs as well as EA-relevant goals against the prefabricates of the GQM approach, we can elicit two common ”patterns”, how goals are incorporated into or linked to an EA model. The first pattern (GOAL-TO-ANY-OBJECT) typically contains a variant of the information model fragment shown in Figure 1. Such fragment can be found in the approaches of TOGAF [6], of Winter et al. [7], of Niemann [10], of Engels et al. [11], and of Johnson and Ekstedt [18]. In the understanding of the GQM approach, the pattern operationalizes a goal into the corresponding questions, more precisely into the characterization for the ”objects of measurement” that the questions provide.

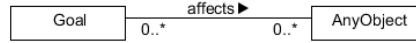


Fig. 1. Pattern GOAL-TO-ANY-OBJECT

The second pattern (METRICS-TO-INDICATORS) describes the fact that architectural properties are interrelated to indicators, more precisely used to define these indicators. This pattern can be found in the EA management approaches of Niemann [10], of Gringel and Postina [12], of Lankes and Schweda [16], and of Johnson and Ekstedt [18]. Against the conceptual framework of the GQM approach, this pattern is slightly more complicated to understand. Many of the architectural properties also are metrics, i.e. are per se at-least ordinally scaled. The properties that do not directly support this level of measurement, e.g. the nominal property used to indicate the *standard vs. custom software* nature of a business application, are in the context of the pattern METRICS-TO-INDICATORS supplied with additional measurement assumptions that allow to interpret them on a higher level of measurement. In the standardization example, such assumption would read as ”standard is better than custom software”, although in reality much more detailed measurement assumptions are employed. Put in other words,

the pattern aggregates different measurable architectural properties (metrics) to a more coarse grained "way of assessment", i.e. a question in the sense of the GQM approach. Different implementations for this pattern and thereby required relationships between architectural properties are given by the approaches:

Mathematical expressions are used by Lankhorst et al. [15], by Niemann [10], and by Gringel and Postina [12]. These expressions are highly expressive, but miss a linkage to the information model. Further, they may be formulated on an abstract level only using a non-further specified function as well as on a concrete and executable level.

OCL expressions are used by Lankes and Schweda [16]. OCL expressions are slightly less expressive than their mathematical counterparts but provide a strong linkage to the information model. Regarding the level of abstractness, only concrete, i.e. executable, OCL expressions can sensibly be formulated.

Influence diagrams are used by Johnson and Ekstedt [18]. Influence diagrams allow a concise and abstract description of relationships between architectural properties and relevant indicators, but are per se of limited expressiveness¹. Additionally, influence diagrams do not provide means to link to the information model.

While each of the aforementioned patterns captures a part of the GQM tri-fecta, both patterns are on their own not able to implement the GQM approach in the context of EA modeling. With this more elaborate understanding of the context at hand, we can concretize the research question from Section 1 to a research gap as follows:

EA management aims to develop and evolve the EA in the direction of EA-relevant goals. To support such development, an EA model should link each goal to the thereby affected objects, should operationalize goals into more concrete EA questions, and should provide measurable metrics for answering the EA questions in a quantitative way. How can we describe EA-relevant goals, questions and metrics in a way to achieve all of the former, while being flexible in respect to both the underlying EA information model and the concrete measurement as well as aggregation prescriptions?

Subsequent Section 3 is dedicated to explaining a technique for annotating EA information models with informations on EA management goals.

3 Introducing a technique for annotating EA information models with goals

Before we introduce our technique for goal modeling in the context of EA modeling, we have to provide some clarifications and definitions that should help us to

¹ In [19] Johnson et al. describe how influence diagrams can be enriched with a more expressive semantics.

avoid ambiguities in the remainder of the section. In line with the understanding of the term "model" as put forward by Stachowiak in [20], we define EA modeling as the activity of creating "purposeful abstractions of the EA" with respect to their "intended usage context" and "using stakeholders". From this, we can emphasize on two central challenges of EA modeling, namely:

- abstracting the **right part** of the overall EA
- to support the **intended usage context** that corresponds to pursuing **intended EA-relevant goals**.

These challenges are closely related, but nevertheless yield two different perspectives from which EA modeling can be approached. Firstly, each EA model commits to a specific EA *concern*², i.e. area-of-interest in the overall architecture of the enterprise. Secondly, each EA model commits to at least one EA-relevant goal, i.e. provides information necessary for (measuring) the achievement of the goal. While proceeding towards our technique for goal modeling, this dichotomous nature of each EA model must be kept in mind.

Taking a concern perspective, each EA model depicts a part of the architectural reality of an enterprise. Any EA model is expressed³ using a distinct description language, the so-called *modeling language*. The necessity to have a modeling language complementing a model at first seems quite obvious, but finds further support when reasoning on *identity*. Van Leeuwen discussed in [22] that identifying a real world individual is only possible, if one associates a type to the individual and if the type supplies a conceptualization of identity. What might sound like an ontological sophistry is well exemplified by Guizzardi in [23] as follows: To know what the real world individual the term "Mick Jagger" refers to, we have to know the type of individual that we can draw our conception of identity from. So the identification of "Mick Jagger" becomes possible by knowing that we look for a **Man** "Mick Jagger". Guizzardi calls types that supply a conception of identity "sortals". Linking back to the concern perspective on an EA model, we can sensibly assume that each individual in the model is instance of a sortal, as e.g. **Business Application** or **Business Process**. These sortals, properties thereof, and the relationships between the sortals form the EA information model, which we – up to this point – have rather intuitively understood as meta-model for an EA model. Figure 2 exemplifies the concern perspective in EA information modeling, depicting the sortals **Business Application** and **Business Process**. To indicate the sortal nature of the corresponding classes, we utilize the UML profile stereotypes put forward by Guizzardi in [23].

From the usage context or goal perspective, each EA model reifies questions related to goals and provides corresponding quantitative answers via metric values. This calls for distinct characteristics in the models underlying modeling

² The term *concern* is used here in line with its definition in the ISO Standard 42010 (cf. [21]).

³ One might argue that this fact only holds for *explicit* models, i.e. models accessible to more than one person. For the purpose of this work, we do not regard this a relevant confinement.

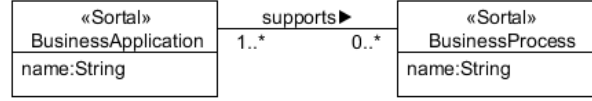


Fig. 2. Exemplary EA information model from concern perspective

language. Put more precisely, a metric can be identified with a property in the EA model’s corresponding information model. Picking up the working definition of metric from Section 2.1, we can further promote the constraint that a property reflecting a metric must be of a datatype supplying at least an ordinal scale. This conceptual modeling for metrics aligns well with the examples for metrics as given e.g. by Lankes and Schweda [16]. When it comes to the representation of a question in the EA model or the EA information model, respectively, the situation becomes a bit more intricate. The twofold nature of the question as

1. designator for objects of measurement, i.e. of objects that are affected by a certain goal, and as
2. aggregator for metrics that measure the achievement of the corresponding goal

aggravates a conceptualization of a question in the EA information model. To devise a suitable modeling construct, we employ a “linguistic trick” and substantiate the question into a conceptual type. For exemplifying this trick, we take the following question that results from the operationalization of a business process optimization goal:

(Q) What is the performance (latency, throughput) of a business process?

In order to answer this question in an EA information model, the sortal **Business Process** has to be “performance-measurable”, i.e. must supply properties reflecting the metrics “latency” and “throughput” which are used to define “performance”. The, admittedly artificial, nomination of “performance-measurable” sheds an interesting light on a possible conceptualization of a question: a question literally adds an *attribute* to a corresponding *noun*, i.e. a sortal. In line with Guizzardi (cf. [23]) such attribution defines a specialized ontological type, the so-called *mixin*⁴. Mixins are *dispersive* types, i.e. specify sets of related (and co-located) properties, of which conversely none provides an identity to the thereby described type. Using a mixin type, we can mirror a question as a concept in the EA information model as shown in Figure 3. In the figure we again utilize the UML profile stereotypes put forward by Guizzardi in [23], here to indicate that a class represents a mixin type. We further employ a slightly adapted form

⁴ In programming languages *mixins* are sometimes referred to as *aspects*. For reasons of clarity we abstain from overloading this term.

of the notation of the *attribute dependency relationship* described by Buckl et al. in [24].

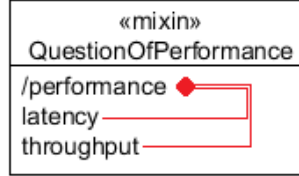


Fig. 3. Exemplary EA information model from goal perspective

Using the notation of the *attribute dependency relationship*, we display that the property "performance" is defined by the metrics "latency" and "throughput", while simultaneously abstaining from needlessly giving details on how to actually derive a performance measurement. This resembles the *definitorial relationship* in the influence diagrams of Johnson and Ekstedt (cf. [18]), but in contrast to the former allows a seamless integration into modeling technique used for EA information modeling. The properties of "performance", "latency" and "throughput" are further not assigned data types, as different measurement procedures for the two metrics will lead to different data types. In aggregating the metrics, more precisely their values, into a single quantitative answer to the question of performance, i.e. to a value for the derived "performance" property, also different options exist. The absence of a data type in the performance mixin accounts for this fact.

In order to consistently model the aforementioned question (**Q**), we have to apply the mixin onto the sortal **Business Process** and have to supply data types for the corresponding metrics, namely for "latency" and "throughput". At this point, an integrated EA information model is created. Using this information model, an enterprise can answer the question of performance in a comparative manner, i.e. can determine for any two business processes A and B, if business process A is more, equally or less performant than business process B, or if no statement can be given. The integrated EA information model clearly indicates that such comparisons are possible, based on component-wise comparison of the two defining metrics "latency" and "throughput", which – according to the definition of the term metric from Section 2.1 – are each comparable. Figure 4 displays the metrics' values for exemplary business processes A to E, whereas complementing Table 5 displays the results of the corresponding performance comparisons. The symbol $\neq$ therein denotes that no comparison is possible.

Having integrated the **Question of Performance** mixin into the sortal **Business Process**, a goal- and concern-specific EA information model is developed. Aforementioned elementary comparison can further be used to provide decision support in evolving an EA, more precisely the business process support

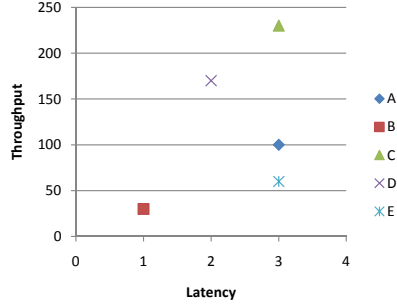


Fig. 4. Latency-throughput diagram

	to	A	B	C	D	E
from						
A		=	≠	<	<	>
B		≠	=	≠	≠	≠
C		>	≠	=	≠	>
D		>	≠	≠	=	>
E		<	≠	<	<	=

Fig. 5. Comparison table

provided by the company's business applications. Nevertheless, this is not necessary the end of a development process for an enterprise-specific information model. A using enterprise may go beyond the basic characterization and select dedicated datatypes for the metrics "latency" and "throughput", for example "milliseconds" and "items per hour", respectively. Based on this information, the definitorial relationships between these metrics and the property "performance" can be reified to a concrete computation prescription. Figure 6 shows the augmented EA information model resulting from the next development step.

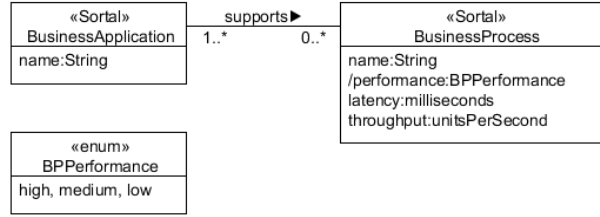


Fig. 6. Integrated information model from a goal and concern perspective

The information model defines a datatype **BPPerformance** distinguishing three categories of business process performance. For determining, which value for the performance property applies, the following derivation rule reifying the *attribute dependency relationship* is supplied:

$$\text{performance} := \begin{cases} \text{low} & \text{if } \text{latency} > 10 \wedge \text{throughput} < 100 \\ \text{high} & \text{if } \text{latency} < 1 \vee \text{throughput} > 1000 . \\ \text{medium} & \text{else} \end{cases}$$

4 Critical reflection and outlook

In this paper we elicited how EA-relevant goals can be operationalized into questions, which are conversely represented as mixin types for annotating EA information models. With the mixins defining architectural properties, we could further show how concrete metrics are linked to the questions and incorporated into an EA information model. Thereby, a consistent modeling of EA-relevant goals, operationalizing questions and quantifying metrics could be achieved. An example building on the prefabrics of Johnson and Ekstedt [18] as well as of Buckl et al. [2] provided first insights into the applicability of the presented technique.

Aside the example, applicability of the technique has not been subjected to further investigation. Especially, an analysis of the usefulness of the technique in a practical setting remains to be undertaken. Such analysis would be especially interesting, as the ability to "mix" questions into arbitrary EA information model sortals has the potential to lead to a plethora of combinations, of which not all might be sensible. Exemplifying this, one could think of the application of the aforementioned **Question of performance** mixin to the sortal of the **Organizational Unit**. While on a fairly abstract level, it might be sensible to assess the performance of an organizational structure, the metrics of **latency** and **throughput** may not be the most appropriate to do so. In the light of such unusual modeling that can result from applying the technique, it remains to be analyzed in a practical setting, if EA information models of that kind actually are created. Put more precisely, a practice cases have to show, if potential users of this modeling technique, can more easily create sensible models or if that kind of abstraction increases the danger to create "absurd", i.e. non-sensible, models.

In the paper also only a single EA information model and a single type of question have been analyzed. While this was sufficient to show, how the method could be applied, a broader analysis in respect to other information models, goals and corresponding questions is yet to be performed. Refraining the modeling of questions by Johnson and Ekstedt in [18], we are confident that the proposed technique can be applied to a broad variety of EA-relevant goals and their operationalizing questions, respectively. When it comes to EA information models, the pattern based method introduced by Buckl et al. in [2] and pertaining to a pattern language as described in [25] makes us confident that the technique devised in this paper can widely be applied in the context of EA modeling.

The pattern language further indicates a direction, into which mixin-based question modeling could evolve: future research could develop a collection or language of "question patterns". Such pattern would describe typical questions as repeatedly used in addressing EA-relevant goals together with best-practice quantifications of the questions, i.e. with practice-proven metrics for answering the questions. This further links to a tool for utilizing such collection of questions in designing an enterprise-specific EA information model. This tool should support the fragment- and mixin-based development and evolution of EA information models for practical usage contexts. With the help of such tool,

the aforementioned danger of creating non-sensible information models could be analyzed in experiments or case studies.

References

1. Aier, S., Kurpjuweit, S., Riege, C., Saat, J.: Stakeholderorientierte dokumentation und analyse der unternehmensarchitektur. In Hegering, H.G., Lehmann, A., Ohlbach, H.J., Scheideler, C., eds.: GI Jahrestagung (2). Volume 134 of LNI., Bonn, Germany, Gesellschaft für Informatik (2008) 559–565
2. Buckl, S., Ernst, A.M., Lankes, J., Schneider, K., Schweda, C.M.: A pattern based approach for constructing enterprise architecture management information models. In: Wirtschaftsinformatik 2007, Karlsruhe, Germany, Universitätsverlag Karlsruhe (2007) 145–162
3. Basili, V.R., Caldiera, G., Rombach, H.D.: The Goal Questin Metric Approach. Wiley, New York (1994)
4. van der Raadt, B., van Vliet, H.: Designing the enterprise architecture function. In: 4th International Conference on the Quality of Software Architectures (QoSA2008), Karlsruhe, Germany (2008) 103–118
5. Krantz, D.H., Luce, D.R., Suppes, P., Tversky, A.: Foundations of Measurement. Volume 1, Additive and Polynomial Representation. Academic Press, New York, NY, USA (1971)
6. The Open Group: TOGAF "Enterprise Edition" Version 9. <http://www.togaf.org> (cited 2010-02-25) (2009)
7. Österle, H., Winter, R., Hoening, F., Kurpjuweit, S., Osl, P.: Der St. Galler Ansatz des Business Engineering: Das Core Business Metamodel. Wisu – Das Wirtschaftsstudium **2**(36) (2007) 191–194
8. Kurpjuweit, S.: Stakeholder-orientierte Modellierung und Analyse der Unternehmensarchitektur. PhD thesis, Universität St.Gallen (2009)
9. Aier, S., Kurpjuweit, S., Schmitz, O., Schulz, J., Thomas, A., Winter, R.: An engineering approach to enterprise architecture design and its application at a financial service provider. In: Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management 27.-28. November 2008 Saarbrücken. (2008) 115–130
10. Niemann, K.D.: From Enterprise Architecture to IT Governance – Elements of Effective IT Management. Vieweg+Teubner, Wiesbaden, Germany (2006)
11. Englebert, V., Heymans, P.: Towards more extensible metacase tools. In Krogstie, J., Opdahl, A.L., Sindre, G., eds.: CAiSE. Volume 4495 of Lecture Notes in Computer Science., Berlin, Heidelberg, Germany, Springer (2007) 454–468
12. Gringel, P., Postina, M.: I-pattern for gap analysis. In: 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010), Paderborn, Germany (2010)
13. Addicks, J.S., Steffens, U.: Supporting landscape dependent evaluation of enterprise applications. In Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P., eds.: Multikonferenz Wirtschaftsinformatik, Berlin, Germany, GITO-Verlag (2008) 1815–1825
14. Addicks, J.S., Gringel, P.: Application landscape metrics: Overview, classification, and practical usage. In: 3rd International Workshop on Enterprise Modelling and Information Systems Architectures, Ulm, Germany (2009) 55–68

15. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin, Heidelberg, Germany (2005)
16. Lankes, J., Schweda, C.M.: Using metrics to evaluate failure propagation and failure impacts in application landscapes. In: Multikonferenz Wirtschaftsinformatik, Berlin, Germany, GITO-Verlag (2008)
17. OMG: Object constraint language (ocl) available specification, version 2.2 (formal/2010-02-01) (2010)
18. Johnson, P., Ekstedt, M.: Enterprise Architecture – Models and Analyses for Information Systems Decision Making. Studentlitteratur, Pozkal, Poland (2007)
19. Johnson, P., Johansson, E., Sommestad, T., Ullberg, J.: A tool for enterprise architecture analysis. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland, USA, Annapolis, Maryland, USA, IEEE Computer Society (2007) 142–156
20. Stachowiak, H.: Allgemeine Modelltheorie. Springer-Verlag, Wien, Austria (1973)
21. International Organization for Standardization: Iso/iec 42010:2007 systems and software engineering – recommended practice for architectural description of software-intensive systems (2007)
22. van Leeuwen, J.: Individuals and sortal concepts: an essay in logical description metaphysics. PhD thesis, University of Amsterdam (1991)
23. Guizzardi, G.: Ontological foundations for structural conceptual models. PhD thesis, CTIT, Centre for Telematics and Information Technology, Enschede, The Netherlands (2005)
24. Buckl, S., Ernst, A.M., Schweda, C.M.: An extension to the essential meta-object facility (emof) for specifying and indicating dependencies between properties. Technical report, Technische Universität München (2008)
25. Chair for Informatics 19 (sebis), Technische Universität München: Eam pattern catalog wiki. <http://eampc-wiki.systemcartography.info> (cited 2010-02-25) (2010)