Association for Information Systems
AIS Electronic Library (AISeL)

2011

Prioritizing Strategic IT Projects with Tropos

Andreas Gehlert

Federal Ministry of the Interior, andreas.gehlert@bmi.bund.de

Follow this and additional works at: http://aisel.aisnet.org/wi2011

Prioritizing Strategic IT Projects with Tropos

Andreas Gehlert
Federal Ministry of the Interior
Alt-Moabit 101D
10559 Berlin
++49 (0) 30 18681 1743

andreas.gehlert@bmi.bund.de

ABSTRACT

One of the daily tasks of an enterprise architect is to prioritize strategic IT projects. To achieve a business-IT alignment, this prioritization needs to be based on business strategies and goals. Therefore, business goals and their traceability to strategic IT projects are relevant for the enterprise architect. However, surprisingly little formalisations and reasoning techniques have been developed in the enterprise architecture domain. In this paper we show that the popular goal modelling technique Tropos together with its formal reasoning techniques can support the enterprise architect when prioritizing strategic IT projects. We prove the feasibility of our work with a tool implementation of the proposed modelling language and its corresponding algorithms; and demonstrate their usefulness with the help of an example taken from the enterprise architecture literature.

Keywords

Enterprise Architecture, Prioritization, Goal Modelling, Tropos, Formal Methods

1. Introduction

Enterprise architecture (EA) is a young discipline and has been recently recognised not only by researchers but also by practitioners. In this paper we are interested in the link between strategic IT projects, initiated by the enterprise architect to strengthen the company's architecture, and the company's business goals (business-IT-alignment [7, p. 77, 12-13, 21, 26]). Since the enterprise architecture should always serve the business [7, pp. 72], the enterprise architect should be able to demonstrate the value of each strategic IT project to the business. This demonstration can be achieved by showing how a strategic IT project contributes to business goals of the company. Linking each strategic IT project with the business goals enables the enterprise architect, additionally, to prioritize strategic IT projects according to the importance of the business goals.

This paper concentrates on the latter aspect and shows how enterprise architects can prioritize their strategic IT projects according to the business goals of their company. We show that current goal modelling techniques used in the EA field are not formal enough to allow for such a prioritization. The research question of this paper is, therefore, to provide a goal modelling technique suitable for the EA domain, which allows prioritizing strategic IT projects

efficiently.

The construction of such a technique is a design science activity and we, therefore, use a design science research method [9, 19]. Such a method starts with requirements for the envisioned artefact. These requirements do not only drive the development of the artefact but are also used to evaluate existing artefacts (cf. Section 2). We use the following requirements for our paper:

Req-1 The goal model must provide concepts to represent goals, relations between goals and relation between goals and strategic IT projects.

Req-2 The goal modelling technique is formal enough to apply formal reasoning techniques.

Req-3 A formal reasoning technique is desirable which calculates the impact of a strategic IT project on all goals of the goal model.

Req-4 The additional information needed to use the formal reasoning technique should be minimal.

Req-1 is derived from the fact that the enterprise architect may need to explain the value of strategic IT projects to the business. Therefore, the goal modelling technique needs to allow modelling goals, strategic IT projects and relations between these elements (see motivation above). Req-2 and Req-3 enables to use the goal modelling technique efficiently by applying formal reasoning techniques. These formal reasoning techniques are a prerequisite to execute parts of the reasoning process on a machine and, therefore, release the enterprise architect from manual work. This is especially interesting for large goal models. Req-4 states that the usage of formal reasoning techniques should not come at the cost of higher modelling efforts for the enterprise architect.

In the sequel we demonstrate how existing goal modelling techniques from the requirements engineering (RE) domain and its reasoning techniques can be applied to the problem of prioritizing strategic IT projects. The paper is structured as follows: Section 2 describes the related work in EA and RE on goal modelling. We show that Tropos – a goal modelling language from the RE domain – is a good candidate, which could be extended to suit our requirements. In section 3 Tropos is briefly described, applied to an example from the EA domain and extended by an algorithm. This algorithm generates a prioritisation of strategic IT projects based on a given goal model. The section also contains a lightweight evaluation of the proposed technique and the algorithm using a tool implementation. We discuss important design decisions and assumptions in Section 4 and summarise our findings in Section 5.

2. Related Work

In this section we review goal modelling techniques found in the EA and RE disciplines. The aim of this section is to identify interesting approaches, whose fragments can be used to construct a goal modelling technique, which fulfils our requirements.

2.1 Goal Modelling in EA

The popular Zachman framework did not include goals from the very beginning [26]. Sowa and Zachman introduced them as the "why" perspective on enterprise architectures five years later [21]. Goals are part of business strategies and are modelled as trees using goal-sub-goal relations. Formal reasoning techniques for analysing the goal tree are not included in the Zachman framework (violations of Req-1 and Req-3).

The same can be said for ARIS [17-18]. In ARIS, goals and relations between goals can be modelled. ARIS' goal trees share the same shortcomings with the goal trees of the Zachman framework (violations of Req-1 and Req-3).

Goals are also mentioned in TOGAF's Architecture Development Method (ADM), Phase A [12]. The so called architecture vision is a set of goals for that architecture. It is used to communicate and to agree upon the future enterprise architecture. This vision should be based on an existing business strategy. Despite the importance of the architecture vision, no methodological support can be found in TOGAF (violations of Req-1, Req-2 and Req-3).

QUASAR Enterprise prescribes the step-wise refinement of goals from business goals to IT-related goals [7, pp. 72]. The relations between goals should serve as traceability links from business to IT. These traceability links can be exploited to explain the value the IT provides to the business. Although the modelling technique is not described in detail, it seems that a tree structure is assumed between the goals. This tree structure shares the same properties with the goal trees of the Zachman Framework and ARIS (violations of Req-1 and Req-3).

In Archimate goals are not seen as core construct of the framework. The authors argue that this concept and its relations can be added to the framework as needed ([3, p. 7]: violations of Req-1, Req-2 and Req-3).

2.2 Goal Modelling in RE

Goal modelling in RE can be divided into two main research lines: research on i* introduced by Yu [25] and research on KAOS introduced by Dardenne [6]. The i* approach was later extended by Tropos [4-5]. Based on these goal modelling approaches, formal reasoning techniques were developed to allow choosing between different designs of a software system.

Mylopoulos introduced was the first who introduced a formal reasoning technique in the goal modelling domain [15]. The author describes qualitative propagation rules, which explain how an evidence for a satisfied (denied) goal can be propagated in a goal graph. The problem with such qualitative reasoning techniques is that they become quickly inconclusive [23, p. 389]. Therefore, the use of qualitative techniques for EA is limited since the reasoning algorithm may not produce valid results (violation of Req-3).

Mylopoulos' algorithm was later extended by Letier and van Lamsweerde [14], van Lamsweerde [23] and Sebastiani et al. [20] to support quantitative reasoning in KAOS and Tropos: However, the approaches rest on the introduction of additional variables for each goal (quality variables in [14]; gauge variables in [23] and costs in [20]). In addition, rules must be assigned to each relation between goals, to propagate these variables along the goal graph structure. These approaches clearly provide higher accuracy and interpretability of the results but at the cost of higher efforts for eliciting the required information. This property hinders the applicability of these approaches in the EA domain (violation of Req-4).

The most promising algorithm was introduced in Tropos by Giorgini et al. [10-11]. The authors describe a formal extension to Tropos, which can propagate qualitative and quantitative satisfaction rates of goals in a goal model. The algorithm computes satisfiability and deniability values of all goals in the goal model by analysing the relationships between these goals. Since the approach supports quantitative reasoning, the inconclusiveness problem does not occur. In addition, the algorithm uses only information, which is already encoded in goal models and, therefore, does not require additional information to use the formal reasoning techniques (fulfilment of Req-2 and Req-4). However, Tropos's algorithm is not capable of computing a prioritization of strategic IT projects (modelled as plans in Tropos) as the algorithm does not compute a quantitative number expressing how one goal influences all other goals in the goal model. The Tropos' algorithm needs to be extended for this purpose (partial fulfilment of Req-3).

2.3 Conclusion for this Paper

As we have seen, goal modelling is an important topic in the EA literature. However, surprisingly little formalisations have been developed in this area. The focus is more on simple and informal goal trees, which hinder the usage of formal reasoning techniques.

In line with previous research by Bleistein et al., who applied RE goal modelling techniques to modelling business strategies, we conclude that using RE goal models and their formal reasoning techniques will be beneficial for the enterprise architect [1-2]. The reason for this conclusion is threefold:

1. Goal models in RE are used to prioritize requirements and therewith to choose between different system designs [24]. The task of making an informed choice between different system designs is very similar to the task of the enterprise architect to choose between different strategic IT projects.
2. Goal modelling in requirements engineering is used to align requirements to initial stakeholder goals and, therefore, allows tracing requirements to business goals [22]. This traceability is very similar to tracing strategic IT projects to business goals.
3. Goal modelling techniques from the RE field are well developed, e. g. they support different types of relations between goals and are not limited to goal trees. In addition, these techniques are formal enough to apply formal reasoning techniques to them (cf. Section 2.2). Since goal modelling techniques in the EA domain are not yet very well developed, it is reasonable to transfer existing knowledge from the RE domain to the EA domain.

Since RE and goal models are used for similar purposes (prioritisation of requirements vs. prioritisation of strategic IT projects; traceability of requirements to business goals vs. traceability of EA goals to business goals) we conclude that RE goal modelling techniques are *applicable* to EA problems as well. Together with their formal focus, which enables formal reasoning and the empirical findings we propose to use these techniques in the EA domain (fulfilment of Req-1).

We chose Tropos from the list of goal modelling techniques discussed in Section 2.2 because it is equipped with formal reasoning techniques, which can be used to construct an algorithm for prioritizing strategic IT projects (Req-2; partial fulfilment of Req-3). Finally, the formal reasoning technique is based on the information in the goal model only; no further elicitation activities are needed (fulfilment of Req-4).
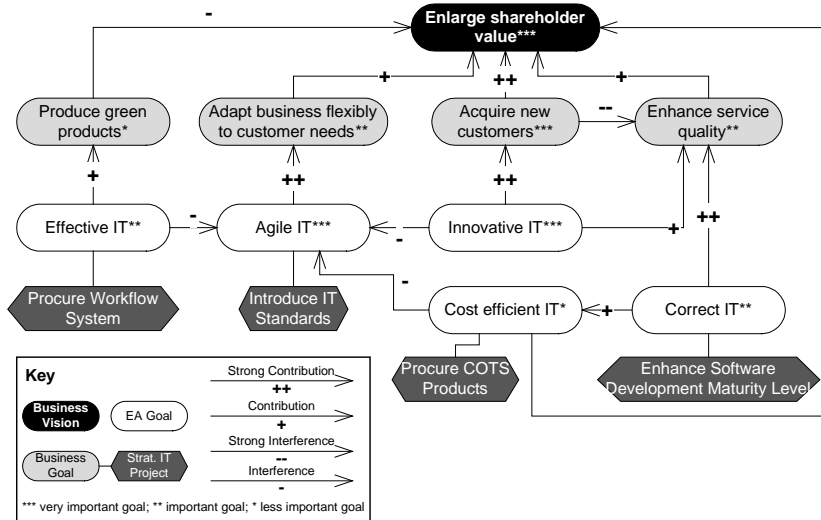
Figure 1: Tropos Example (adapted from [7])

3. Goal Modelling with Tropos

In this section we develop an algorithm, which satisfies requirement Req-3 and provides decision support for prioritizing strategic IT projects. We ground this algorithm in Tropos – an established goal modelling technique. We start our investigation by introducing Tropos (Section 3.1). Then we explain how Tropos can be used to describe and analyse EA problems (Section 3.2). We develop the algorithm in Section 3.3. Finally in Section 3.4, we provide a lightweight evaluation of our technique by applying it to an example taken from the literature. This evaluation also demonstrates how the approach can be used to prioritize strategic IT projects.

3.1 Introduction to Tropos

Tropos is a RE technique, which rests on the agent oriented paradigm and uses goal modelling techniques known from i* for analysing early and late requirements. These early requirements are documented as actor models and goal models [for the following explanations see 4, pp. 206]. *Actor models* include stakeholders of the later system modelled as actors and describe the actors' goals and dependencies. The actor model is complemented by a goal model for each actor. This *goal model* shows the decomposition of the actor's goals into sub-goals.

Sub-goals in goal models can be derived by decomposing super-goals with And/Or decomposition links or by using contribution links. Decomposition links are used to hierarchically decompose a goal into sub-goals. In case of an *Or decomposition*, the super-goal is satisfied if at least one sub-goal is satisfied (modelling alternatives); in case of an *And decomposition* all sub-goals must be satisfied to satisfy the super-goal. *Contribution links* are further described with a strength, which specifies how much a sub-goal contributes to a super-goal. These strengths could also be negative to describe an interference of goals.

Goals are further distinguished in soft-goals and hard-goals. *Hard-goals* have clear cut criteria to decide whether the goal is satisfied. *Soft-goals* do not have such clear cut criteria.

Each goal-model can be complemented with plans. *Plans* describe tasks or activities to be carried out to achieve a certain goal. Plans

are connected to goals with *means-end links*, where the plan represents the "end" and the goal represents the "mean".

Figure 1 demonstrates the elements of Tropos. This example used throughout this paper. The model contains a part of a business strategy of a company and its related EA goals. The EA goals are taken from [7, p. 77]. The top goal of this model is to "Enlarge the shareholder value". It is supported by the goals "Adapt business flexibly to customer needs" and "Enhance service quality"; it is strongly supported by the goal "Acquire new customers" and interfered by the goal "Produce green goods". The EA goals are interpreted analogously. For instance, the goal of an "Agile IT" supports the goal "Adapt business flexibly to customer needs" strongly. Strategic IT projects are modelled in Tropos as plans (hexagon). Linking these strategic IT projects to goals means that the goal is satisfied once the project is realised. For instance, implementing the strategic IT project "Procure Workflow System" satisfies the goal "Efficient IT".

3.2 Applying Tropos to EA Problems

In this subsection we describe important restrictions on using Tropos. These restrictions are expressed as design decisions and assumptions. These design decisions and assumptions allow a) tailoring Tropos to EA problems by choosing a relevant part of the Tropos language; and b) focussing the paper. The consequences of all design decisions and assumptions are discussed in Section 4.

We concentrate on goal models here because we are interested in breaking down business goals to strategic IT projects and not in analysing the stakeholders involved in this activity. Therefore, we deal with goal models only.

> *DD-1   Actor models are disregarded.*

In addition, we restrict ourselves to soft-goals. In RE, hard goals describe (mainly) functional requirements. In this sense it can be decided using clear-cut criteria whether a software system has a certain functionality or not. Soft-goals do not have such clear cut criteria. They are used to describe quality requirements, e. g. usability. Using soft-goals as concept for describing high-level business and EA goals seems reasonable since there are no clear-cut criteria for goals such as agility, efficiency, time-to-market, etc.

DD-2   Hard-goals are disregarded.

It is unlikely that the remaining soft-goals are equally important [cf. 23, p. 390]. Therefore, we introduce the new meta-property importance and assign it to the soft-goal concept. We distinguish between very important, important and less important goals and visually mark them as "***", "**" and "*" respectively.

DD-3   The goal concept is extended by the property importance. The property values are in the range {very important, important, less important}.

Furthermore, we do not use decomposition links here. The reason for this design decision is to reduce the complexity in the later model. For the remaining contribution links we have to decide whether we use a quantitative or qualitative notation for the strengths of the links [10]. Using quantitative reasoning real numbers are assigned to the strength of a contribution link whereas using qualitative reasoning the contribution link is usually annotated as $\{++, +, --, -\}$ meaning strong contribution, contribution, strong interference and interference respectively. We use the qualitative model here to reduce the effort for specifying the contribution link strength.

DD-4   Decomposition links are disregarded.

DD-5   A qualitative notation for contribution link strengths is used.

We assume that strategic IT projects are modelled as plans and are assigned to goals using means-end links. This model fragment means that an EA activity contributes to the satisfaction of the assigned goal. We assume that each plan is assigned to exactly one goal and that the implementation of this plan satisfies this goal completely.

A-1)   Plans represent strategic IT projects and are assigned to exactly one goal. Realising this plan means fully satisfying the assigned goal.

3.3 Impact Analysis in Tropos

Once the goal model is developed and the relevant strategic IT projects are assigned to these goals, the question about the priorities of strategic IT projects arises. When selecting strategic IT projects, the enterprise architect wants to achieve a positive impact on many goals while avoiding a negative impact on goals at the same time [16, p. 95]. Therefore, an impact analysis based on the dependencies between the goals and their importance is a good starting point for this analysis.

Since each plan is assigned to exactly one goal, we can disregard the plans and can concentrate on the goals only (assumption A-1). A goal model is then a directed, weighted graph $DG = (G, C)$ where G represents a set of goals and C represents a set of contribution links with strength ω. The graph is connected (e. g. there are no lose goals), incomplete and may contain cycles as well as multiple edges.

In the following we distinguish between a positive and a negative impact of a goal g ($I^+(g)$ and $I^-(g)$ respectively). Informally the positive impact $I^+(g)$ describes the contribution of g to all connected goals. The negative impact $I^-(g)$ describes the interference of g with all connected goals. In addition, an overall impact $I(g) = I^+(g) - I^-(g)$ is used for an initial prioritization of goals. By connected goals we mean all goals to (with) which g contributes (interferes) including all transitively connected goals. The contribution (interference) to (with) transitively connected goals should degrade the longer the path to the connected goal is.

Let P(g) describe the relative importance of goal g in comparison with all other goals in the goal graph. It describes the importance of g in isolation, e. g. without considering its relations with other goals. This importance value needs to be considered when calculating the impact values and deriving the prioritization of goals.

For the following formalisations of the term "impact" we extend an algorithm proposed by Giorgini et al. [10-11]. The authors introduce two variables $Sat(g)$ and $Den(g)$ for each goal, which describe the evidence that goal g is satisfied or denied. The authors also define rules, which propagate these evidence values along the contribution links in the goal graph. Here, we use the probabilistic model described by the authors and define: Let a contribution link c $\in C = (g_1, g_2)$ with strength ω be represented as $g_1 \xrightarrow{\omega} g_2$. The propagation rules can then be described as:

$g_1 \xrightarrow{\omega} g_2$:

$$\begin{cases} Sat(g_2) \geq Sat(g_1) * \omega; \ Den(g_2) = Den(g_1) * \omega & \text{if } \omega > 0 \\ Sat(g_2) \geq Den(g_1) * |\omega|; \ Den(g_2) = Sat(g_1) * |\omega| & \text{if } \omega < 0 \end{cases}$$

The first line of the formula states that the evidence of a goal satisfiability and deniability is propagated along the contribution link and degraded by ω in case of a (positive) contribution ($\omega > 0$). The second line states that the evidence for satisfiability of g_1 is propagated to the deniability value of g_2 and vice versa and degraded by ω in case of an interference (negative contribution; $\omega < 0$). Please note that according to these definitions, $Sat(g)$ and $Den(g)$ are always positive.

In addition to the propagation rules, Giorgini et al. provide an efficient algorithm to compute $Sat(g)$ and $Den(g)$ for an arbitrary goal model. This algorithm especially considers cycles and multi-edges in the graph. However, the algorithm can only work if $Sat(g)$ and $Den(g)$ is given for a set of goals before running the algorithm. These initial values correspond to an alternative, which should be evaluated.

To calculate the impact of a goal g, we initialise its satisfiability value $Sat(g) = 1$, its deniability value $Den(g) = 0$ and set the satisfiability and deniability values to zero for all remaining goals: $\forall_{g_i \in G \wedge g_i <> g}: Sat(g_i) = Den(g_i) = 0$. Based on these initial values we calculate the satisfiability and deniability values of all goals using Giorgini et al.'s algorithm. The positive (negative) impact $I^+(g)$ ($I^-(g)$) is then the sum of these satisfiability (deniability) values of each goal weighted by the importance of these goals.

For calculating the impact values we can write the following pseudo code:

1) Initialisation: $Sat(g) = 1, Den(g) = 0$ and $\forall_{g_i \in G \wedge g_i <> g}: Sat(g_i) = Den(g_i) = 0$.
2) Apply Giorgini et al.'s algorithm to compute $Sat()$ and $Den()$ for all goals.
3) Compute impact: $I^+(g) = \left(\sum_{g_i \in G} Sat(g_i) * P(g_i) \right)$ - $P(g)$ and $I^-(g) = \sum_{g_i \in G} Den(g_i) * P(g_i)$

The algorithm presented here works with quantitative measures for ω. However, design decision DD-5 prescribes a qualitative model for the contribution link strengths. Therefore, we have to define a mapping between the two systems. We use the following mapping here:

$$\omega_{quantitative} = \begin{cases} 1 \text{ if } \omega_{qualitative} = " + + " \\ 0.5 \text{ if } \omega_{qualitative} = " + " \\ -1 \text{ if } \omega_{qualitative} = "--" \\ -0.5 \text{ if } \omega_{qualitative} = " - " \end{cases}$$

Similarly, we use the following quantification for the importance of goals:

$$P(g)_{quantitative} = \begin{cases} 1 \text{ if } P(g)_{qualitative} = \text{"very important"} \\ 0.5 \text{ if } P(g)_{qualitative} = \text{"important"} \\ 0.25 \text{ if } P(g)_{qualitative} = \text{"less important"} \end{cases}$$

Figure 2 provides an exemplary application of the algorithm to a simple goal model. The impact values for all goals are computed as follows:

- *Goal C* does neither contribute nor interfere with any other goal. Therefore, its positive and negative impacts are $I^+(C) = I^-(C) = 0$.
- *Goals B and E* support (interfere with) goal C. Therefore they have a positive (negative) impact of $I^+(B) = 0.5$ ($I^-(E) = 0.5$) and a negative (positive) impact of $I^-(B) = 0$ ($I^+(E) = 0$).
- *Goal D* contributes to E and has, therefore, a positive impact of $I^+(D) = 0.5$. In addition, it interferes with C transitively via E and has, therefore, a negative impact of $I^-(E) = 0.25$.
- *Goal A* contributes to goals B (strongly) and transitively to goal C (left hand side of Figure 2). The positive impact of this part is 1.5. In addition, goal A strongly interferes with goal D. Since goal D contributes to goal E, A also interferes transitively with goal E (see algorithm). Therefore, the negative impact of goal A from this part is 1.5. In addition, goal E interferes with goal C. Due to the interference of A with D the interference between E and C is calculated as positive impact. Therefore, the positive transitive impact from goal A on goal C is 0.25. Since $Sat(A)$ must be greater than 1.5 the 0.25 value from the sub-graph A-D-E-C is disregarded. Therefore, goal A has a positive and negative impact of $I^+(A) = I^-(A) = 1.5$.
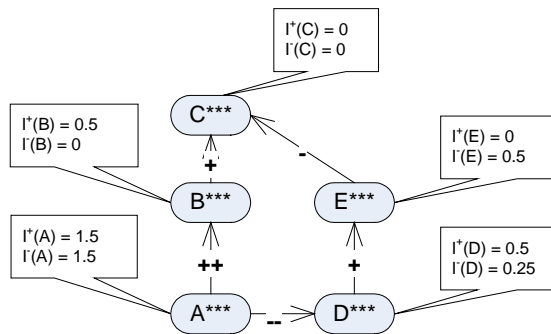

Figure 2: Exemplary Calculation of the Impacts in a Simple Goal Model

3.4 Lightweight Evaluation Using an Example

We have already shown that the technique is suitable for EA problems (fulfilment of Req-1; see Section 3.2) and that an algorithm can be constructed that calculates the impact of one goal on the entire goal model (fulfilment of requirements Req-2 and Req-3; see Section 3.3). Here, we are interested in demonstrating that the proposed technique and its corresponding algorithm are useful

for answering important EA questions (substantiating the fulfilment of Req-1). We analyse this usefulness with the help of the example depicted in Figure 1.

For this proof of concept, we prototypically implemented the algorithm in Microsoft Visio (see Figure 3).
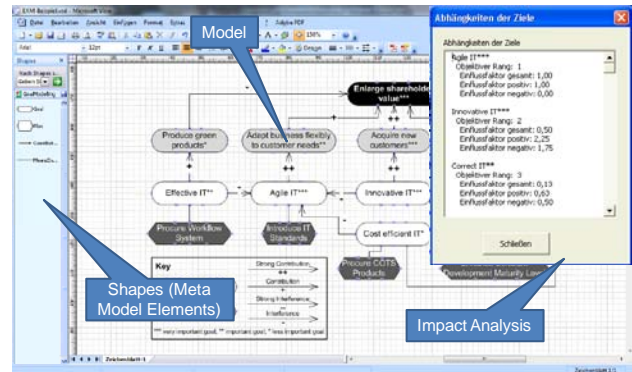

Figure 3: Screenshot of the Prototypical Tool Implementation

We used the introduced algorithm to calculate overall impact values $I(g)$ for all goals in Figure 1. This overall impact value respects positive (wanted) impacts as well as negative (unwanted) impacts respectively. It reflects our previous observation that the enterprise architect should concentrate on goals with high positive but low negative impact values [16, p. 95]. Figure 4 depicts the visualised impact values for all goals in the goal model of Figure 1. We have ordered the goals according to their impacts to improve the readability of the diagram.
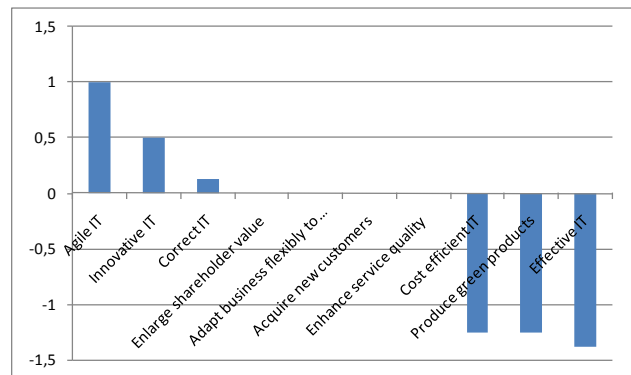

Figure 4: Impacts of all Goals for the Example in Figure 1

The typical structure of the goal model in Figure 1 results in a small number of goals with high impact values and a large number of goals with low impact values. Figure 4 clearly shows that the impact values are degrading rapidly in the example model. This property helps the enterprise architect to concentrate on few important goals rather than on may equally important ones. Having an agile and innovative IT are the two top goals in this example.

Another property of the algorithm can be clearly identified in Figure 4: The algorithm considers the impact values of all (transitively) connected goals. Therefore, it is most likely (depending on the concrete distribution of the contribution link strengths) that fine-grained goals have higher impacts values than coarse-grained ones. Therefore, fine-grained goals will be in the focus of the enterprise architect. Applying this finding to Figure 4, we clearly see that the top goal "Enlarge shareholder value" has no impact on

the goal model and that the business goals are distributed "around" this top-goal. Since these business goals are outside the scope of the enterprise architect, they can be disregarded for the prioritization of strategic IT projects.

Assumption A-1), furthermore, ensures that strategic IT projects (modelled as plans in Tropos) are assigned to exactly one goal. Since concrete strategic IT projects need to be assigned to concrete goals, this leads to the situation that the enterprise architect refines the goal model. Together with the previous observation, the algorithm ensures that fine-grained goals are preferred. Consequently, the enterprise architect gets advice on concrete strategic IT projects, which should be implemented in the future. The same holds true for strategic IT projects, which should be avoided due to the low impact value of their assigned goal.

Given our example the enterprise architect should concentrate on introducing IT standards assigned to the high impact goal "agile IT". In addition, the enterprise architect should not primarily strive for an effective and cost efficient IT and, therefore, should not implement the strategic IT projects "Procure Workflow System" and "Procure COTS products".

The example shows that there is no plan associated to the high impact goal "Innovative IT". The enterprise architect should consider either refining this goal or adding concrete strategic IT projects (plans) to it. In this way new and previously unrecognised activities are considered. Vice versa, our analysis demonstrates that one goal has a significant negative impact and should, therefore, not be supported by strategic IT projects (e. g. efficient IT in our example). The enterprise architect might – after a thorough analysis – decide to remove these activities from the model to avoid them entirely in the future.

We can summarise the findings from our lightweight evaluation of the proposed techniques with three guidelines for the enterprise architect:

1) Goals with high impact values should be achieved. Goals with low impact values should be disregarded.
2) Goals with high impact values and no associate plans should be refined and new plans should be associated to these goals.
3) Removing plans for goals with low impact values associated with these plans should be considered.

4. Discussion

Our approach is based on five design decisions and one assumption. The design decisions basically remove concepts from the Tropos goal modelling vocabulary, while the assumption restricts the ways Tropos is used. The impact of these decisions and assumptions are discussed in the following.

Design decision DD-1 removes actor models from the Tropos vocabulary. These actor models can potentially be used to analyse the enterprise architecture's stakeholders [12, pp. 281]. In combination with a formal reasoning algorithm, the impact of strategic IT projects could be traced back not only to business goals but also to the stakeholders. However, the current Tropos algorithm does not cover actor models and, therefore, the algorithm proposed here cannot be easily extended to actor models. To construct such an algorithm and to proof its usefulness in the EA domain is, therefore, subject to future research.

Design decision DD-2 disregards the hard goal concept. Hard goals are goals, which have clear-cut criteria to decide whether this goal is fulfilled. In terms of our algorithm it means that hard goals should have either a satisfiability value of 1 or a deniability value of 1 (but particularly no values <1). Giorgini et al.'s algorithm explicitly supports hard goals [10]. Therefore, relaxing this limitation does not have any impact on the formal part of the paper. However, it should be carefully investigated whether hard goals are useful in EA since they may increase the complexity of the goal model without having any other positive effect.

Design decision DD-3 introduced the meta-property importance for the goal concept with its possible values "very important", "important" and "less important". Although this extends the Tropos method, this extension was necessary to reflect the fact that not every goal is equally important for the enterprise architect. In addition, the extension has no impact on the existing Tropos algorithm so that existing formalisations can be used without modification.

Design decision DD-4 restricts the goal model to contribution links. Particularly decomposition links are not considered. Again, Giorgini et al.'s algorithm covers decomposition links so that our algorithm will work with decomposition links too [10]. The design decision is, therefore, not a restriction. However, decomposition links only add value in case of an And decomposition since Or decompositions are equivalent to contribution links with a strength of $\omega = 1$. With equivalent we mean that the algorithm treats Or decompositions and strong contribution links in the same way [see 8 for this argument]. From this respect, it does not add much value to the diagram and it should be carefully considered whether the decomposition concept is really needed.

Design decision DD-5 prescribes the use of qualitative strengths for contribution links. Since our algorithm is based on a quantitative calculation, the design decision could be removed without any effect on the algorithm. So, this decision does not restrict the application of the algorithm. However, from the practical view, the qualitative model for contribution link strengths and importance values should be preferred due to the easier elicitation of the contribution strengths and the enhanced readability of the diagram (requirement Req-4).

Assumption A-1) ensures that plans are only related to exactly one goal. This assignment means that the realisation of the plan fully satisfies the goal. There are two different situations, which seem to be impossible to model: 1) a plan might not completely satisfy the goal; 2) more than one plan might be necessary to satisfy the goal. In both cases, the initial goal assignment can be replaced by a goal model: in situation 1) a new goal is introduced and assigned to the plan – this goal contributes to the initial goal (with a contribution link strength "$\omega < ++$"); in situation 2) the initial goal is refined by more than one sub-goal and each sub-goal is assigned to exactly one plan. This assumption is, therefore, not a restriction.

We can conclude that our design decisions and assumptions do not have severe consequences for using Tropos and its formal reasoning techniques. Consequently, Tropos goal models could be used with all its concepts and the algorithms will still produce accurate results.

Furthermore, we decided how to translate qualitative contribution link strengths and goal importance values to quantitative number. The enterprise architect needs to assign appropriate mappings for the qualitative importance and contribution link strengths since there is no general guideline for such an assignment. However, the mapping for the contribution link strengths needs specific consideration since values smaller than one reduces the impact of transi-

tive goals on the overall impact value. Therefore, it seems reasonable to propose a contribution link strength of $\omega = 1$ for strong contribution links so that the impact of goals connected with these links does not degrade transitively.

A property of the proposed algorithm is the distinction between positive and negative impact values for each goal. Although we have only used the overall impact value in Section 3.4, the negative and positive impact values provide additional information for the enterprise architect. Consider for instance two goals with the same overall impact value. The enterprise architect should prefer the goal with the smallest negative impact. In addition, the situation $I^+(g) > 0 \land I^-(g) > 0$ indicates a situation of conflicting goals. These conflicting goals should be carefully analysed before making a prioritization decision. Supporting this analysis is subject to future research.

A criticism, which applies here as well, was brought forward by van Lamsweerde [23, p. 390]: He argues that the *Sat* and *Den* values and, hence, also the impact values derived from these *Sat* and *Den* values, have no clear meaning in terms of the (EA) domain. This problem remains and further research need to show how this problem can be avoided.

5. Conclusions

In this paper we have concentrated on one specific aspect of enterprise architecture: The enterprise architect may be interested in prioritizing strategic IT projects according to the business strategy and business goals. This prioritisation is important to achieve a good alignment between IT and business.

By analysing existing approaches to model business strategies we have found that techniques from the EA domain are not formal enough to apply formal reasoning techniques to the resulting goal models. We have shown that Tropos, a goal modelling technique from the RE discipline, could fill this gap. Its formal reasoning technique allows the enterprise architect to gain a quick overview of the prioritisation of strategic IT projects even if numerous business goals and many relations between them exist.

This prioritisation of strategic IT projects is realised by extending Tropos' algorithm to support the calculation of positive and negative impact values for each goal in the goal model. These impact values encode the relative importance of each goal as well as the different relations between the goals. A high overall impact of a goal signifies a high relevance of this goal to the enterprise architect; and a low overall impact value signifies a low relevance of this goal for the enterprise architect.

Based on the concept of impact value we have identified three rules for the enterprise architect: First, goals with high impact values should be achieved at first while goals with low impact values should be disregarded. Second, goals with high impact values and no associated strategic IT projects should be refined and new strategic IT projects should be assigned to them. Third, strategic IT projects associated with low impact goals should be reconsidered.

However, the outcome of the algorithm comes also with a warning: The algorithm was not yet tested empirically. This means that the algorithm's output should be treated as decision support not as final decision. Thus, the prioritisation of strategic IT projects should be analysed thoroughly before taking any action.

We see further research in three fields: 1) to analyse whether the algorithm can also be applied to actor models to analyse the dependencies between the various stakeholders of an Enterprise Architecture; 2) to extend the approach so that the enterprise architect can interpret the impact values using enterprise architecture phenomena without significantly increasing the effort of eliciting the required information; and 3) to support the enterprise architect when analysing conflicting goals.

6. REFERENCES

[1] Bleistein, S., Cox, K., Verner, J. and Phalp, K. B-SCP: A Requirements Analysis Framework for Validating Strategic Alignment of Organizational IT Based on Strategy, Context, and Process. *Information and Software Technology*, *48*, 9 (2006), 846-868.

[2] Bleistein, S.J., Cox, K. and Verner, J. Validating Strategic Alignment of Organizational IT Requirements Using Goal Modeling and Problem Diagrams. *Journal of Systems and Software*, *79*, 3 (2006), 362-378.

[3] Boer, F.d., Bonsangue, M., Buuren, R.v., Groenewegen, L., Hoppenbrouwers, S., Iacob, M.-E., Jonkers, H., Lankhorst, M., Proper, E., Stam, A., Torre, L.v.d. and Zanten, G.V.v. Concepts for Architectural Description, Centrum voor Wiskunde en Informatica, Radboud Universiteit Nijmegen, Leiden Institute of Advanced Computer Science, Ordina Public B.V., Telematica Instituut, 2007.

[4] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, *8*, 3 (2004), 203-236.

[5] Castro, J., Kolp, M. and Mylopoulos, J. Towards Requirements-Driven Information Systems Engineering: The Tropos Project. *Information Systems*, *27*, 6 (2002), 365-389.

[6] Dardenne, A., van Lamsweerde, A. and Fickas, S. Goal-Directed Requirements Acquisition. *Science of Computer Programming*, *20*, 1-2 (1993), 3-50.

[7] Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.-P., Voss, M. and Willkomm, J. *Quasar Enterprise - Anwendungslandschaften serviceorientiert gestalten*. dpunkt Verlag, Heidelberg, 2008.

[8] Gehlert, A., Bramsiepe, N. and Pohl, K., Proceedings of the 4th International Workshop on Service-Oriented Computing Consequences for Engineering Requirements (SOCCER 2008), September 8, 2008, Barcelona, Spain. in, (2008).

[9] Gehlert, A., Schermann, M., Pohl, K. and Krcmar, H., Towards a Research Method for Theory-Driven Design Research. in *Proceedings der 9. Internationale Tagung Wirtschaftsinformatik Business Services: Konzepte, Technologien, Anwendungen (WI 2009), 25. – 27. Februar 2009, Wien, Österreich*, (2009), Österreichische Computer Gesellschaft, 441-450.

[10] Giorgini, P., Mylopoulos, J., Nicchiarelli, E. and Sebastiani, R. Formal Reasoning Techniques for Goal Models. in *Journal on Data Semantics*, Springer, Berlin, Heidelberg, 2003, 1-20.

[11] Giorgini, P., Mylopoulos, J., Nicchiarelli, E. and Sebastiani, R., Reasoning with Goal Models. in *Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002), October 7--11, 2002, Tampere, Finland*, (Berlin, Heidelberg, 2002), Springer, 167-181.

[12] Group, T.O. *TOGAF Version 9*. Van Haren Publishing, 2009.

[13] Lankhorst, M. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, 2009.

[14] Letier, E. and van Lamsweerde, A. Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering. *SIGSOFT Software Engineering Notes*, *29*, 6 (2004), 53-62.

[15] Mylopoulos, J., Conceptual Modeling and Telos. in *Conceptual Modeling, Databases and CASE: An Integrated View of Information Systems Development*, (New York, 1992), Wiley, 49-68.

[16] Mylopoulos, J., Chung, L., Liao, S., Wang, H. and Yu, E. Exploring Alternatives During Requirements Analysis. *IEEE Software*, *18*, 1 (2001), 92-96.

[17] Scheer, A.-W. *ARIS -- Modellierungsmethoden, Metamodelle, Anwendungen*. Springer, Berlin, Heidelberg, 1998.

[18] Scheer, A.-W. *ARIS -- Vom Geschäftsprozeß zum Anwendungssystem*. Springer, Berlin, Heidelberg, 1998.

[19] Schermann, M., Gehlert, A., Krcmar, H. and Pohl, K., Justifying Design Decisions with Theory-Based Design Principles. in *Proceedings of the 17th European Conference on Information Systems (ECIS 2009), June 8-10, 2009, Verona, Italy*, (2009), 1065-1076.

[20] Sebastiani, R., Giorgini, P. and Mylopoulos, J., Simple and Minimum-Cost Satisfiability for Goal Models. in *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), June 7-11, 2004, Riga, Latvia*, (2004), Springer, 20-35.

[21] Sowa, J.F. and Zachman, J.A. Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*, *31*, 3 (1992), 590-616.

[22] van Lamsweerde, A., Goal-Oriented Requirements Engineering: A Guided Tour. in, (2001), 249-262.

[23] van Lamsweerde, A., Reasoning about Alternative Requirements Options. in *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, (Berlin, Heidelberg, 2009), Springer-Verlag, 380-397.

[24] van Lamsweerde, A., Requirements Engineering in the Year 00: A Research Perspective. in, (2000), 5-19.

[25] Yu, E. An Organisational Modelling Framework for Multiperspective Information System Design, Department of Computer Science, University of Toronto, Toronto, 1993.

[26] Zachman, J.A. A Framework for Information Systems Architecture. *IBM Systems Journal*, *26*, 3 (1987), 276-292.