Towards Goal-Driven Self Optimisation of Service Based Applications

Andreas Gehlert and André Heuer

University of Duisburg-Essen, Schützenbahn 70
45117 Essen
{andreas.gehlert,andre.heuer}@sse-uni-due.de

Abstract. Service based applications are constructed to be easily adaptable to changing environments. This adaptation was primarily investigated with respect to monitoring events, e. g., a service based application is adapted when the execution of a service fails. In this paper we focus on the adaptation of service based applications due to newly available service. In this respect we discuss whether a service of the service based application should be replaced by a service, which becomes available. From the requirements engineering perspective we argue that a service based application may be adapted when the new services contribute better to the goals of the service based application. In addition, we show that it may also be valuable to adapt a service based application when newly available services provide more functionality than the ones previously used. Both analyses are based on model comparison techniques with Tropos goal models and Tropos' reasoning techniques.

Keywords: Adaptation of Service Based Applications, Model Comparison, Goal Models, Requirements Engineering.

1 Introduction

In traditional software engineering three different motivations for software maintenance and corresponding software adaptations are distinguished [1, p. 493]: Corrective maintenance aims to eliminate errors in the software which were not discovered during the testing phase. Adaptive maintenance reacts to changes in the system's context and adapts the software to new requirements. Perfective maintenance aims to improve the current software, e. g. its performance.

Due to an increased number of mergers in the industry [2, pp. 12] and an increased pressure on IT departments to improve their efficiency [3, pp. 5] service based applications (SBA) evolved, which particularly facilitate adaptation. This adaptation is enabled by defining technologies for describing, finding and composing services. By *service* we understand any computational resource offered over a network [4, p. 51].

One current vision is to engineer SBAs to enable self-optimization. Such self-optimising systems "… continually seek ways to improve their operation, identifying and seizing opportunities to make themselves more efficient …." [5, p. 43] Although

P. Mähönen, K. Pohl, and T. Priol (Eds.): ServiceWave 2008, LNCS 5377, pp. 13–24, 2008.

current adaptation and monitoring approaches aim to eliminate errors in the SBA (corrective maintenance) or to improve the performance of the SBA (perfective maintenance), they are usually technology-centred and disregard requirements engineering (RE) aspect of SBAs.

In this paper we complement current adaptation and monitoring approaches with a RE perspective. Our approach strives to achieve self-optimising SBAs (perfective maintenance) by means of fulfilling given requirements. To focus the paper we limit the discussion of adaptation scenarios to the analysis whether existing services in the SBA should be replaced by newly available services. Other adaptation techniques, e. g. the reconfiguration of the workflow or an adaptation of the service infrastructure are not addressed.

If a new service is available, the requirement engineer investigates whether the new service improves the SBA with respect to its requirements. In order to use a new service in a SBA, we raise two requirements: First, the service must "fit" in the existing SBA. Second, the service must contribute to the SBA's requirements better than previously available services. The first requirement ensures that the service provides the functionality needed by the SBA. The second requirement ensures that the new service is superior to existing services.

To satisfy both requirements, we chose a goal-driven approach. The main reason for this choice is the availability of reasoning techniques for goal models, which allows analysing the effect of the satisfaction of a single goal on the whole goal model. This facilitates the analysis of the impact of a single service on the entire SBA. To address requirement one, we compare the goal model of the SBA and the goal model of the new service. The service is only useable if its goal model is identical or similar to the goal model of the SBA. To find out whether the service provides higher satisfaction ratios for the SBA's goal model, we use the goal reasoning mechanisms provided in [6]. We argue, that an adaptation of an initial SBA is advisable if all goals are at least as satisfied as in the initial situation but one goal has a higher satisfaction (pareto principle). In case that some goals of the SBA achieve higher satisfaction rates in the new situation and some goals achieve lower satisfaction rates, the requirements engineer may decide on the adaption of the SBA. Lastly, an adaptation may also be advisable when the new service provides additional functionally.

We chose to use Tropos as goal modelling approach [7, 8]. The rationale for using Tropos is threefold: First, Tropos is a comprehensive approach to develop software systems from early requirements to its implementation. We are particularly interested in the early RE activities, which are well covered by Tropos. Second, Tropos was already applied to the service discipline, e. g., it was already shown that it is applicable to SBAs [e. g. 9, 10-13]. Third, Tropos comes with a formalisation which allows analysing the influence of the satisfaction of one goal on the entire goal model [6].

The paper is organised as follows: In section 2 we introduce Tropos' goal modelling techniques, which are used throughout this paper. In section 3 we show how goal models can be used to decide whether a new service should be used in an existing SBA, e. g. whether an adaption of this SBA is beneficial. We discuss the limitations of our approach along its assumptions in section 4, review the related work in section 5 and provide conclusions in section 6.

2 Goal Modelling in Tropos

Tropos rests on the agent oriented paradigm and uses goal modelling techniques known from i* [14] for analysing early and late requirements. These early requirements are documented as actor and goal models. *Actor models* include actors, their goals and their dependencies. The actor diagram is complemented by a *goal model* for each actor. This goal model shows the decomposition of the actor's goals into sub-goals and plans (tasks in i*; cf. Fig. 1 for an example).

Since we are only interested in the SBA (represented as actor in Tropos) and not its interrelation with other actors, we only use Tropos' s goal models. Its main concepts are actors, goals, resources and plans. These elements are connected with decomposition, contribution and means-end links. An *actor* represents an organisational unit, a position or a role. The actor's strategic interests are represented by *goals*. Goals are further divided into hard-goals and soft-goals. While *hard-goals* have clear cut measurement criteria to specify its fulfilment, *soft-goals* do not have such criteria. In this paper we use hard-goals to model functional and soft-goals to model quality requirements. Goals can be decomposed using And/Or *decomposition* links. A *plan* is an activity which may contribute to a goal. A plan may also *contribute* positively or negatively to soft-goals. *Means-end* links are used to represent which plan or goal (means) is used to fulfil a goal (end, [7, pp. 206]).

In a SBA each plan describes a service, which realises this plan [9, p. 21]. To implement a SBA, it is, therefore, necessary to find services, whose descriptions fit the plan. Consequently, a SBA in the Tropos early RE perspective is a set of services fitting a set of plans. The description of this fitness relation is described in the next section.

3 Using Goal-Models for Adapting SBAs

To define the fitness relation between a plan and a service we need to define what a plan and a service is. A plan in Tropos is defined by its name and its relations to goals and soft-goals via a set of means-end links M and via a set of contribution links C. Since we later use a satisfaction propagation mechanism to describe the dependencies between different goals, it is sufficient to include only directly connected means-end and contribution links in the definition of a plan. Thus, we can define a plan as a tuple of sets of means-end and contribution links: $plan = < M, C >$.

To compare plans and services we assume:

(A1) Services are described by Tropos goal models. These goal models are registered together with the service in a service registry.

(A2) Each service's goal model contains only the name of the service as plan and only goals and soft-goals connected by means-end and contribution links. Consequently, the service goal model is structurally equivalent to the plan's goal model.

Assumption (A1) is critical insofar as service providers need to describe their services with goal models. A detailed discussion of this assumption is postponed to section 4. Assumption (A2) is not critical as we describe below an algorithm, which produces this sub-model from any Tropos goal model.

According to assumption (A2) services and plans are described by the same elements. Consequently a service is also a tuple of sets of means end and contribution links: $service = < M, C >$. A means-end link $m \in M$ is a connection between one Tropos model element e and a goal g. Each contribution link $c \in C$ is a connection between one Tropos model element e and a soft-goal s. It is attributed with a quantitative number to express the strength ω of the contribution to this soft-goal: $c = < e, s, \omega >$.

As we concentrate on requirements engineering for adapting SBAs, we assume that a SBA is already running and that its initial set of requirements are expressed as Tropos goal model. In addition, each service in this SBA fits to one plan in the requirements specification.

After the initial SBA is operating, the service provision is monitored, e. g. by regularly querying service registries. When new services are available, an adaptation cycle is triggered. The first activity in this cycle extracts a goal model for each plan of the SBA's goal model. These goal models are in turn compared to the service's goal model. After this comparison, the goal achievements for all goals in the Tropos goal model are calculated based on formal reasoning techniques described below. These results are than used to decide about the adaptation of the SBA. After this adaption the process starts again. The underlying assumption of this RE process can be formulated as follows:

(A3) New services become available over time.

This assumption is fair because the flexibility of SBAs is only feasible when new services are made available over time.

In the next step we need to describe how the goal model for each plan can be extracted from the entire Tropos model. Extracting a goal model for each plan is necessary since we want to delegate the execution of each plan to a service and since the Tropos model represents the entire SBA and not only one individual service. Intuitively the goal model for each plan contains all its contribution and means-end links and all the connected elements. Both link types are important because we want to know how the service for each plan influences the goal achievement of the entire SBA. In sum, the plan's goal models contains: the plan, all connected goals via contributions links and all connected goals via means-end links.

To illustrate the extraction of the goal model for one plan, Fig. 1 provides an example of a retailer system [partially taken from 9, p. 22]. The main goal of this retailer system is to sell products. A goal analysis revealed the goals "order handling" and "cataloguing" (And decomposition). Tasks and soft-goals were added to the diagram and were connected to these goals with means-end relationships and contribution links respectively. The warehouse service for instance contributes positively with the strength of *+0.3* to the soft-goal "performance". For the initial SBA, each plan is the description of a service. At the beginning of the RE process, the goal model of this service is identical to the extracted goal model of the plan. The goal model for the plan "eShop Service" for instance contains the hard-goal "Cataloguing" and the soft-goals "Performance", "Availability", "Transaction Reliability" as well as the respective means-end and contribution links (Fig. 1, right).

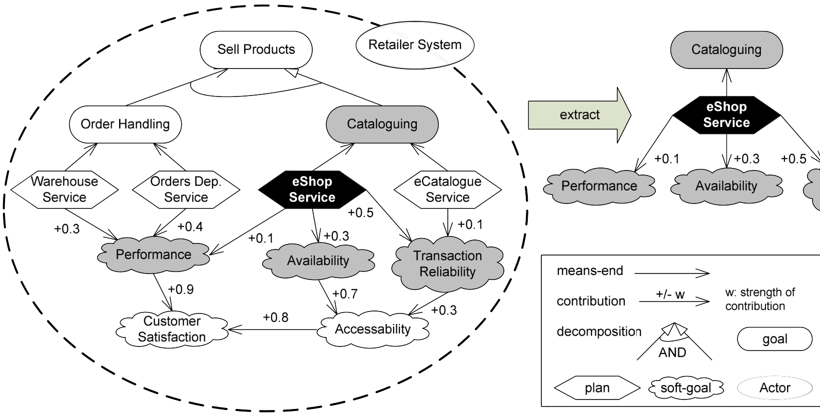

Fig. 1. Extracting the Sub-Model for the Plan “eCatalogue Service” [example partially taken from 9, p. 22]

3.1 Comparing the Service and the Plan Goal Models

After the specification of the initial requirements and the extraction of the goal model for each plan, we need to monitor the service provision. If the service provision changed we want to find out whether the newly available services “fit” the existing plans better than the initial set of services. In this section we define this fitness relation. This fitness relation is based on the comparison of the plan’s goal model and the service’s goal model in accordance with assumptions (A1) and (A2).

The systematic analysis of model comparison conflicts was initially developed in the data modelling discipline. Batini et al. distinguish between naming conflicts, structural conflicts and type conflicts.

- *Naming conflicts* arise due to the different usage of the natural language in models [14, p. 344].
- *Type conflicts* can be traced back to the divergent usage of the modelling language, e. g. to express one and the same real world phenomenon as entity type or as attribute [14, p. 346].
- *Structural conflicts* arise when a real world proportion is differently reconstructed by different modellers, e. g. because of different goals of the modelling project [14, p. 346].

A model comparison technique aims to identify the before-mentioned conflicts. *Naming conflicts* can be resolved by analysing the homonyms and synonyms used in the models and by renaming the respective model elements so that identical names in both models have the same and different names in the models have different meanings. As this problem was already solved previously [for an overview cf. 15, p. 344], we assume here that naming conflicts were already resolved:

(A4) The service's goal model and the plan's goal model use a shared ontology, i.e. two goals with the same name are identical and two goals with different names are different.

Resolving *type conflicts* means to define a similarity relation between equivalent or similar model structures. Our models contain only hard-goals, soft-goals, contribution links and means-end links (cf. assumption (A2)). In addition, hard-goals describe functional requirements and soft-goals represent quality requirements. As functional requirements in the service domain are described by the web service description language (WSDL) and non-functional requirements are described by service level agreements (SLA) it follows that hard-goals and soft-goals are mutually exclusive and cannot be resolved in the type conflict analysis. The remaining elements are means-end and contribution links. Means-end links are used whenever plans and soft-goals provide a means to achieve a goal [7, p. 208]. Consequently, the means fully satisfies the goal or soft-goal, which is identical to a contribution link with a degree of *+1.0*.

To define the fitness relation between a plan and a service we introduce the functions $name()$, which returns the name of a goal model element and $type()$, which returns the type of a model element. Based on this analysis of type conflicts we can now define when a plan matches a service description:

$$p \xrightarrow{fits} s \Leftrightarrow$$

$$\forall m_P \in M_P : \begin{pmatrix} \exists m_S \in M_S : \left(name(g_P) = name(g_S) \wedge (type(g_P) = type(g_S)\right) \vee \\ \exists c_S \in M_S : \left(name(g_P) = name(s_S) \wedge type(g_P) = soft\text{-}goal \wedge \omega_s = 1\right) \end{pmatrix} \wedge$$

$$\forall c_p \in C_P : \begin{pmatrix} \left(\exists c_S \in C_S : name(s_P) = name(s_S)\right) \vee \\ \left(\exists m_S \in M_S : \left(name(s_P) = name(g_S) \wedge type(g_S) = soft\text{-}goal\right)\right) \end{pmatrix}$$

This fitness relation holds when:

1. Each means-end link m_P in the plan's goal model exists also in the service's goal model (m_S) and the connected goals have identical names and types. As a means end link can also be represented as contribution link, it follows: Each means end link m_P of the plan's goal model exists in the service's goal model as contribution link with the strength $\omega_s = 1$ and the connected goals have the same name and the plan's goal is a soft-goal.
2. It must additionally hold that for each contribution link in the plan's goal model c_p there is either a contribution link c_S in the services goal model and the connected goals have identical names. Alternatively, the contribution link c_p may also be represented as means end link m_S in the service's goal model. In this case both connected goals must have the same name and the service's goal must be a soft-goal.

Lastly, *structural conflicts* cannot be resolved and represent the real differences of the models. However, they can for instance be used to analyse whether a goal model 1 includes goal model 2 but has additional hard- and/or soft-goals.

3.2 Decision Support for Adapting a SBA

An adaptation of the SBA is only feasible if the relation $p \xrightarrow{fits} s$ holds – otherwise the service does not provide the required functionality needed by the SBA. When a new service is registered in a registry (assumption (A3)) and the fitness relation holds for this service for one plan, we can distinguish four situations:

Situation 1 *Equal Goal Satisfaction*: The goal model of the new service is identical to the plan's goal model. In this case the new service can be used as substitute of the existing service, e. g. when the existing service fails.

Situation 2 *Different Goal Satisfaction*: The new service may contribute differently to existing soft-goals by assigning different strengths to contribution links. These different strengths are further propagated in the goal model and may lead to different satisfaction ratios of goals and soft-goals. The adaptation decision is based on these new goal satisfaction ratios.

Situation 3 *Goal extension*: The new service may provide additional functionality not used in the initial SBA. This new functionality is expressed as additional hard-goals in the service goal model, which do not correspond to any goal in the plan's goal model (structural conflict). The SBA may be adapted accordingly to exploit the additional functionality of the new service.

Situation 4 *Goal reduction*: The new service may provide less functionality in comparison to the one used in the current SBA. The requirements engineer may decide using this service in combination with another service, which together fulfil the requirements of the SBA better than the services used previously.

In the following we demonstrate the calculation of the goal satisfaction values in accordance to the newly available service. We use the quantitative reasoning techniques in Tropos goal models presented in [6, p. 10]. The algorithm presupposes that each goal has two variables $Sat(G)$ and $Den(G)$ describing the satisfiability and deniability of the goal. These variables are computed according to the strength ω, which is annotated to contribution links. This strength describes the impact of one goal on another goal. Due to space limitations, we restrict ourselves to goal satisfyability.

For each contribution link $G_2 \xrightarrow{\omega+s} G_1$ with the strength ω Giorgini et al. define the following propagation axiom: $G_2 \xrightarrow{\omega+s} G_1 : Sat(G_2) \geq x \rightarrow Sat(G_1) \geq (x \otimes \omega)$ [6, p. 12]. The operator $\otimes$ is defined as $p_1 \otimes p_2 =_{def} p_1 \cdot p_2$. In addition, we assume that means-end links can be treated like contribution links with $\omega = 1$. We can now use the axiom to calculate $Sat(G)$ for each goal of the goal model. This goal propagation assumes the following:

(A5) The strengths ω of all service's goal models are comparable, e. g. they are measured objectively.

This assumption is necessary to actually compare the satisfaction ratios of the soft-goals among different services. We discuss this assumption in section 4.

The result of the label propagation can be presented as bar chart. The y-axis is labelled with goals and soft-goals, the x-axis is labelled with the degree of satisfaction and the bars show the degree of satisfaction of the different goals. The bar chart representation of the goal model in Fig. 1 is depicted in Fig. 2 (black bars).

Assume that new services were registered in a service registry. To use these new services we require, that their goal models are structurally identical to the plan's goal model. Two goal models of new services, which fulfil this requirement are depicted in Fig. 2 (gray bars). Both models conform to the before-mentioned situation 2.

In comparison to the goal model in Fig. 1, the contribution links of service ❶ in Fig. 2 to Transaction Reliability and to Availability have an increased strength (*0.6* instead of *0.5* and *1.0* instead of *0.3*). Using service ❶ means to achieve higher fulfilment rates for all soft-goals. Consequently, using service ❶ is beneficial from the RE perspective and should therefore be used.

Service ❷ in Fig. 2 has a reduced strength for the contribution link to Reliability (*0.3* instead of *0.5*) but an increased strength for the contribution link to Availabilty (*0.*7 instead of 0.*3*). The goals Accessibility, Availability and Customer Satisfaction have now a higher satisfiability. However, the satisfiability of the goal Reliability dropped. A RE expert has to decide about the adaptation of the SBA. In this case this may be valuable because the goal Consumer Satisfaction increased slightly. The RE expert has to balance this advantage with the disadvantage of a lower satisfiability of the goal Reliability.

4 Discussion

The results of this paper are limited to the assumptions (A1) – (A5). In particular it relies on the assumption that new services become available over time (A3). This assumption is fair because changing environments lead to new IT solutions in the past and will most likely lead to new services in the future.

Assumption (A2) is based on assumption (A1) and requires that each provider provides a goal model as description of his/her service (assumption (A1)) and that this goal model consists only of one plan representing the service as well as goals and soft-goals directly connected to the plan with means-end and contribution links. This assumption is also less critical since we explained how such a reduced goal model can be extracted from a larger goal model.

Assumption (A4) requires that service providers and service consumers use a shared vocabulary. Although this assumption is not realistic, it can be eliminated by linguistic approaches, which resolve homonyms and synonyms. For instance, WordNet [16] was successfully used to resolve homonyms and synonyms in the SeCSE approach to requirements engineering [17]. In other words, assumption (A4) was helpful to focus this paper but can be overcome by using existing approaches.

The most critical assumptions are (A1) and (A5). They require that service providers provide a goal model for each service (assumption (A1)) and that the strengths of the contribution links are objectively comparable between different services. Both assumptions seem unrealistic. However, instead of forcing service providers to describe their services with goal models, these goal models may be generated. The central plan element can be generated according to the service's name. The functional requirements of this service are described in a WSDL document. Consequently, the hard-goals are represented by the methods contained in this WSDL document. In

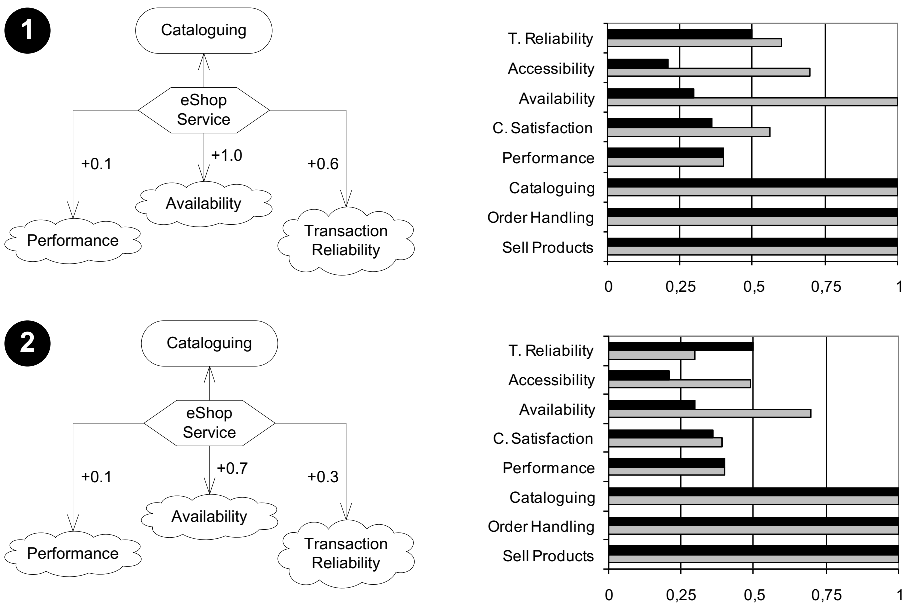

Fig. 2. Quantitative Comparison of Goal Models

addition, a SLA describes the quality requirements for a service and it may be used to generate soft-goals. If the quality characteristics are quantified in the SLA, this quantification can also be used to calculate the strengths of the contribution links. Assume that the requirement for the parameter "response time" is 1s. Service 1 has a response time of *2s* and service 2 a response time of *3s*. Consequently, the strengths of the contribution links are *0.5* for service 1 and *0.33* for service 2. The feasibility of this approach, however, is subject to further research.

5 Related Work

Although Tropos was applied in the service domain, these applications do not explain when to adapt a SBA. Aiello and Giorgini for instance explore quality of service aspects using Tropos actor models [9]. The authors use Tropos' formal reasoning techniques in [6] to calculate the fulfilment of a goal structure according to a given set of services. As the approach by Aiello and Giorgini does not cover the adaptation of a SBA, our approach is an extension to [9]. In another approach Penserini et al. explore how Tropos can be used to develop SBAs. However, the authors do not focus on adaptation. Another application of Tropos was put forward by Pistore et al. in [13]. The authors explain how SBAs can be developed by step-wise refining plans and complementing these plans with a formal workflow definition. Since the focus of Pistore et al. is on deriving service compositions, the authors do not cover adaptation issues.

A similar approach to ours was put forward by Herold et al. in [18]. The authors relate existing components to goal models. This relation is established by so called generic architectural drivers. These drivers enable the selection of existing components, which fit with the goals and soft-goals of the goal model. Herold et al.'s approach focus on finding appropriate components and refining the initial goal model with the help of these components. However, the approach does not address adaptation.

Another RE approach, which is similar to ours, was put forward in the SeCSE project [19]. In SeCSE initial requirements are formulated as goal models [19, pp. 21] or use cases [17, 20-22], which are than translated into services queries [19, p. 31]. These services queries are sent to a registry. The resulting services are used to refine the initial set of requirements. However, in SeCSE the focus was on refining requirements according to the current service provision but not on adapting existing SBAs.

6 Conclusion

In this paper we presented an approach towards self-optimisation of SBAs. In particular we showed how to decide whether a SBA should use newly available services. Informally a SBA should use a new service if this service has the same functionality as the existing service but fulfils the requirements of the SBA better, e. g. has a higher performance than the existing service.

To measure the fulfilment of the requirements we proposed to use Tropos and its formal reasoning techniques. The starting point of our approach is a goal model of the SBA's requirements. Each plan in this goal model is the description of a service. When a new service is available, the Tropos goal model is compared to the goal model of this service. The new service is superior to the previous one, if the goal satisfaction ratios for all goals in the SBA requirements specification are higher compared to the old service. In addition, the requirements engineer may also decide to adapt the SBA, when the new service is superior with respect to some goals but inferior to others or when the new service provides additional functionality expressed as additional goals.

To analyse the impact of one particular service on the whole SBA we use Tropos formal reasoning techniques. These reasoning techniques allow to propagate the local satisfaction ratios of goals of one service to the whole goal model and, thus, to analyse the dependencies of the different services within this SBA.

Our approach clearly shows that the prerequisite of the adaptation of SBAs discussed here requires that service providers and service consumers speak the same language, e. g. that both parties agree on a shared ontology. In addition, the difficulties of measuring quality aspects of services are even more evident since a comparison between two services with respect to their quality attributes requires a shared metrics between service providers and service consumers. One approach, which tackles this problem can be found in [23].

Our approach can be extended in four ways: First, a revised version may also contain the notion of goal deniability introduced in [6], which we left out due to space limitations. Second, we need to show that the proposed approach to generate service goal models from WSDL and SLA specification is feasible. This helps to overcome the restrictive assumption underlying our approach. Third, the combination of the proposed requirements engineering approach with adaptation techniques – in particular with self optimisation techniques – would provide the missing link to the service engineering domain. Forth, the approach should be formally and empirically validated to prove its efficiency.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

References

1. Swanson, E.B.: The Dimensions of Maintenance. In: International Conference on Software Engineering (ICSE 1976), San Francisco, CA, USA, pp. 492–497 (1976)
2. Budzinski, O., Kerber, W.: Megafusionen, Wettbewerb und Globalisierung: Praxis und Perspektiven der Wettbewerbspolitik, vol. 5. Lucius & Lucius, Stuttgart (2003)
3. Mutschler, B.: Die IT und betriebliche Informationssysteme im Spannungsfeld von Innovation und Wirtschaftlichkeit. Ulm (2006)
4. Colombo, M., Nitto, E.D., Penta, M.D., Distante, D., Zuccalà, M.: Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 48–60. Springer, Heidelberg (2005)
5. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. IEEE Computer 36, 41–50 (2003)
6. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R.: Formal Reasoning Techniques for Goal Models. Journal on Data Semantics, 1–20 (2003)
7. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems 8, 203–236 (2004)
8. Castro, J., Kolp, M., Mylopoulos, J.: Towards Requirements-Driven Information Systems Engineering: The Tropos Project. Information Systems 27, 365–389 (2002)
9. Aiello, M., Giorgini, P.: Applying the Tropos Methodology for Analysing Web Services Requirements and Reasoning about Qualities. UPGRADE: The European Journal for the Informatics Professional 5, 20–26 (2004)
10. Lau, D., Mylopoulos, J.: Designing Web Services with Tropos. In: International Conference on Web Services (ICWS 2004), San Diego, CA, USA, pp. 306–313 (2004)
11. Misra, S.C., Misra, S., Woungang, I., Mahanti, P.: Using Tropos to Model Quality of Service for Designing Distributed Systems. In: International Conference on Advanced Communication Technology (ICACT 2006), Phoenix Park, Gangwon-Do, Republic of Korea, pp. 541–546 (2006)
12. Penserini, L., Perini, A., Susi, A., Mylopoulos, J.: From Stakeholder Needs to Service Requirements. In: 2nd International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER 2006), Minneapolis, Minnesota, USA, pp. 8–17 (2006)
13. Pistore, M., Roveri, M., Busetta, P.: Requirements-Driven Verification of Web Services. Electronic Notes in Theoretical Computer Science 105, 95–108 (2004)
14. Yu, E.: An Organisational Modelling Framework for Multiperspective Information System Design. Requirements Engineering 1993 – Selected Papers. Department of Computer Science, University of Toronto, Toronto, pp. 66–86 (1993)
15. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal 10, 334–350 (2001)

16. Miller, G.A.: WordNet - Princeton University Cognitive Science Laboratory (2006), http://wordnet.princeton.edu/
17. Jones, S.V., Maiden, N.A.M., Zachos, K., Zhu, X.: How Serivce-Centric Systems Change the Requirements Process. In: 11th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2005), Porto, Portugal, pp. 105–119 (2005)
18. Herold, S., Metzger, A., Rausch, A., Stallbaum, H.: Towards Bridging the Gap between Goal-Oriented Requirements Engineering and Compositional Architecture Development. In: 2nd Workshop on Sharing and Reusing architectural Knowledge Architecture, Rationale, and Design Intent (SHARK-ADI 2007), Minneapolis, USA (2007)
19. Maiden, N.: Service Centric System Engineering: A2.D5 SeCSE Requirements Process V2.0. City University London (2006)
20. Zachos, K., Maiden, N.: Web Services to Improve Requirements Specifications: Does It Help? In: Paech, B., Rolland, C. (eds.) REFSQ 2008. LNCS, vol. 5025, pp. 168–182. Springer, Heidelberg (2008)
21. Zachos, K., Maiden, N.A.M., Zhu, X., Jones, S.: Discovering Web Services to Specify More Complete System Requirements. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 142–157. Springer, Heidelberg (2007)
22. Zachos, K., Zhu, X., Maiden, N., Jones, S.: Seamlessly Integrating Service Discovery Into UML Requirements Processes. In: Proceedings of the 2006 International Workshop on Service-Oriented Software Engineering (SOSE 2006), Shanghai, China, pp. 60–66 (2006)
23. Zachos, K., Dobson, G., Sawyer, P.: Ontology-aided Translation in the Comparison of Candidate Service Quality. In: Proceedings of the 4th SOCCER Workshop, Barcelona, Spain, September 8 (2008)