Inf Technol Manag
DOI 10.1007/s10799-009-0058-1

Interactive selection of Web services under multiple objectives

Thomas Neubauer · Christian Stummer

Abstract The manual composition of efficient combinations of Web services becomes almost impossible as the number of services increases dramatically. When determining an appropriate set of services, managers must take into consideration given business processes, business strategy and multiple Quality of Service (QoS) objectives while ensuring the cost-efficient usage of limited resources. Because the agility with which new business requirements are adapted has a major influence on business success and poor investment decisions may thus entail corporate failure, decision makers are experiencing growing pressure to prove the value of IT investments—but they often lack appropriate multicriteria decision support tools. This paper introduces a new decision support approach that more properly addresses these challenges. We implemented this approach into a tool and evaluated the performance of two popular methods (i.e., the Analytic Hierarchy Process and the Weighted Scoring Method) by means of a real-life case study in the social security sector. It turns out that the decision support system assists decision makers in identifying investments that more precisely target their company's business needs by allowing them to interactively determine and continually optimize service allocation according to the corporate business processes and multiple (strategic) objectives.

T. Neubauer (✉)
Secure Business Austria, Vienna, Austria
e-mail: neubauer@securityresearch.at

C. Stummer
Department of Business Administration, University of Vienna, Vienna, Austria
e-mail: christian.stummer@univie.ac.at

Keywords Web services selection · Multiple objectives · Interactive decision support

1 Introduction

Companies use business process models to improve their overall performance. With the majority of business processes in today's highly automated world being supported by IT systems, the success of a business strategy depends on the availability of appropriate IT systems. Agile and dynamically adaptable processes are requisite for realizing competitive advantages, but they can only be realized efficiently if IT systems are optimally aligned for providing the needed degree of flexibility [6, 40]. Lately, Service-Oriented Architectures (SOAs) and Web services have become synonyms for methods that make a flexible and agile use of IT possible [1, 50]. SOAs aim at building independent, loosely coupled applications (services) that are created according to business process requirements. Every service provides a well-defined set of functionality that is derived from its origin and underlying implementation-related issues [29]. As soon as business requirements and, thus, business processes change, the underlying services are modified accordingly to reflect these changes. If new services are required, they can be built upon existing ones. The selection of services is important for aligning IT to business, just as it is the case with all IT components. Achieving an effective combination of available technologies brings with it huge potentials for optimization, particularly in the context of loosely coupled and autonomous services.

However, the number of services available over the Internet has increased dramatically during the past few years, making an efficient manual composition of these

Published online: 16 July 2009

services impossible [42]. There are many different service providers selling services with very little if any difference in their functionality. As a result, decision makers selecting the most appropriate set (i.e., portfolio) of services must increasingly focus on additional objectives such as (i) an optimal alignment between business processes and services in order to concentrate resources on value-generating and supplementary business processes, (ii) the cost-efficient use of available resources, or (iii) interdependencies between the services. Moreover, investments must precisely target a company's specific business needs, as competitive advantages can only be accrued by aligning services according to the corporate business processes and strategic objectives. Nevertheless, companies are frequently unaware of the level of their capital expenditure and/or—even more importantly—whether these investments are effective [26]. As a result, decision makers more often than not buy, implement and maintain systems that do not contribute to the corporate value. Morgan Stanley estimates that between 2000 and 2002, US companies spent $130 billion especially for software they either did not need or could not use properly [25]; according to Gartner, companies waste as much as 20% of the annual (worldwide) technology investments of $2.7 trillion [13]. This has led to a need for better measures for aligning IT expenditure to actual business needs. In a 2002 survey among 400+ top IT executives, Verhoef [72] found that 60% felt increased pressure to prove ROI on IT investments, while 70% believed that their metrics did not fully capture the value of IT. Nearly half of the survey participants had no confidence in their ability to accurately calculate ROI on IT investments.

Thus, just spending more is obviously not a winning strategy [32, 38, 43]. Accordingly, a variety of approaches have been introduced that aim to support decision makers in identifying the "right" investment candidates. Those valuation methods that focus solely on financial measures are often considered to be ill-suited for IT investments, because they (unsurprisingly) failed to properly take into consideration the many important non-financial criteria [26, 54]. An analogous reasoning holds for approaches that primarily focus on technical issues such as the adaptation of services to changing environmental conditions [23, 53] (or the challenge of building easily-adaptable services). Thus, multicriteria approaches that are capable to properly considering financial, technical and/or further types of objectives come into play. However, the two approaches that are most commonly used in practice, namely the Weighted Scoring Method (WSM) and the Analytic Hierarchy Process (AHP), have a number of serious drawbacks as will be detailed later in this paper. In particular, decision makers demand to have full control over the decision making process, i.e., they typically are not willing to just provide a priori weights for the criteria under consideration and leave all the rest to some system. Instead they want to learn about alternative solutions and then be supported in making their own decisions in an interactive and intuitive way (cf. [2] and others).

In order to address the reservations and demands outlined above, we have developed a new (two-phase) decision support approach named ATANA (which is derived from "AlTernative ANAlysis" and refers to the Greek goddess of wisdom). In its first phase, ATANA determines solution alternatives (i.e., portfolios of Web services) that are both feasible with respect to given constraints and Pareto-efficient with respect to a number of objectives that have been identified as the most relevant ones for a given decision setting. Note that Pareto-efficiency implies that for each of these solutions there is no other (feasible) solution with values that are at least as good or better in all the objectives, i.e., Pareto-efficient solutions are "good" ones from an objective point of view. In its second phase, ATANA supports decision makers in interactively exploring the determined solution space until they find their individually "best" solution, i.e., the particular combination of Web services that promises a mix of objective values that best fits the decision maker's (private) preferences. In order to illustrate this new approach, this paper describes a real-life case study carried out in the social security sector in Austria.

The remainder of this paper is organized as follows: Sect. 2 provides a background on the composition of Web services. Next, Sect. 3 describes the two above-mentioned common multicriteria approaches for Web services selection, while Sect. 4 introduces the new ATANA approach. Section 5 then illustrates the performances of these three approaches by means of a real-world case study. Finally, Sect. 6 presents our conclusions as well as an outlook on further research.

2 Web services composition

The selection of Web services typically starts with four phases [7], namely:

– *Criteria definition* In this phase, a set of criteria is defined with regard to an organization's infrastructure, application architecture and design, service requirements, objectives and constraints, or the availability of services. Establishing the criteria for service selection is a critical task and one that must be performed individually for each decision instance. Literature particularly suggests the use of Quality of Service (QoS) objectives for the composition of Web services (cf. [22, 28, 69, 70]).

- *Service identification* This step aims at having the user identify services. Identification can be carried out—e.g., in the scope of a tender procedure—by providing a portal where a potential provider can register and provide information about their company and, in particular, its services. Alternatively, the user may use open UDDI-directories for identifying appropriate services.
- *Evaluation* In this phase, the decision maker evaluates the services identified in the second step with regard to the criteria defined in the first step. This holds especially true for qualitative criteria that require decision makers to apply values that are often based on experience. The goal of this step is to provide a common basis for services from different providers so that these can be used as input for the selection algorithm.
- *Selection* The objective of this phase is to filter services based on the evaluation carried out in the preceding step.

In the succeeding phases, these services are integrated into the internal directory and the selection and execution at runtime are carried out. Finally, accounting, as well as the potential deactivation of the services, must be taken care of. While this paper focuses on interactive decision support in the selection phase, a considerable body of literature deals with related issues. Particularly, the automated composition of Web services is widely acknowledged as a promising field of research [33]. Corresponding papers can be roughly divided into two families of approaches, namely composition by planning and business process optimization [59]. Artificial intelligence (AI) planning for the composition of Web services falls in the first group where, e.g., ontologies are used for defining an initial state, an explicit goal representation, and a set of possible state transitions. Such an approach is considered to be highly flexible, as a high level specification of the required functionality is used for automatically or semi-automatically composing a service process. Typical research questions address the discovery of proper services and, to this end, commonly use semantic methods that make it possible to describe the capabilities and processes of Web services [33]. Note that in our paper, we assume that potential services and their functionality are already known (e.g., by applying tools based on semantic languages [18, 74]). Approaches from business process optimization typically rely on BPEL processes that are composed of abstract services. The actual combination of "real" services is determined at runtime by solving an optimization problem. Here, local approaches that consider only local QoS constraints (cf. [4, 44, 77]) can be distinguished from global approaches that consider the constraints for the whole application (cf. [9, 12, 27]). For an example refer to Zeng et al. [77], who proposed solving the problem of Web service composition by measuring each constraint in the QoS model. For this purpose, they introduced a linear mathematical program that took such factors as availability, successful execution rate, response time, execution cost and reputation into consideration. Next, Ardagna and Pernici [4] developed a mixed integer linear approach with both local and global constraints, while Yu et al. [75] came up with a multi-dimensional 0–1 knapsack formulation that takes into considering the practice of different quality levels. Further approaches applied integer programming (cf. [21]) or multi-objective genetic algorithm (cf. [10, 12, 36]). Related literature also refers to Ran [51], who incorporated the QoS model into UDDI registries in order to collect all QoS attributes for each service in a centralized place, or to Sreenath and Singh [58], who described a mutual evaluation process between agents that selects the best services based on rates provided by the agents. Lately, Ardagna and Pernici [5] focused on problems with large processes and severe QoS constraints. For this purpose, they designed a mixed integer linear programming approach and recommend supplementing it with additional negotiation techniques if necessary. For a more detailed literature review on current developments regarding the Web services composition problem (see [5, 12]).

3 Traditional multicriteria approaches

While it is relatively straightforward to calculate the total cost of ownership (TCO) by summing initial investments and succeeding maintenance costs, assessing efficiency gains or additional returns from a potential—yet, hypothetical—IT solution is a difficult and highly approximative task. Moreover, several non-financial (e.g., QoS) aspects must also be taken into consideration in Web services composition (for a discussion refer to [2, 39, 45, 46, 73] and others).

As a consequence, a single solution that is the "best" one from an objective point of view typically does not exist; instead, many reasonable solution alternatives usually exist. Accordingly, methods from multicriteria decision making come into play (also cf. [3, 17, 49, 52]). These can be divided into two complementary areas, namely (i) mathematics-based multiple objective programming (MOP) followed by an interactive decision support phase (e.g., the ATANA approach introduced in this paper) and (ii) multiple criteria decision analysis (MCDA) that encompasses decision makers' judgments and preferences in order to derive formal utility functions representing their preferences. This chapter describes two common approaches of the latter type that have been adapted to IT investment selection so far.

Weighted Scoring Method and Analytic Hierarchy Process are well established and frequently used approaches for the evaluation of IT-investments (cf. [56, 66, 71]). They have been extensively applied to the selection of components-off-the-shelf (COTS; cf. [2, 30, 39, 45, 46, 73]) and many companies are still using AHP and WSM for their IT-investment decisions, including Web service in terms of interfaces to software components, legacy systems or other IT-systems. It has to be considered that the Web service is just the interface to the system or software component that provides the benefits and requires the resources. There is no doubt that companies could use systems for the automatic composition of Web services at runtime, but a vast majority of companies are either (i) still far away from using SOA and Web services or (ii) want to have influence on the investment decision. Therefore, AHP and WSM were chosen for the case study in order to provide decision makers with information on the benefits of using ATANA for their investment decisions.

3.1 Weighted Scoring Method

The WSM is the most commonly used approach in literature (cf. [20]). This method tackles multicriteria problems by weighting each of the objectives numerically and combining the weighted criteria into a single numerical utility indicator value for each candidate. An example is provided in Table 1, in which the contributions of each candidate were rated on a scale ranging from 1 (i.e., "poor") to 9 (i.e., "excellent"); apparently "System B" should be chosen in this example. Procedurally, the steps for the WSM are as follows:

1. Identify all alternative choices.
2. Identify all relevant factors.
3. Identify, judgmentally derive, or compute factor weights for each factor.
4. Construct a WSM table with individual columns for each alternative and one additional column labeled "Factor Weights", as well as rows for each factor and a final row labeled "Total Score".
5. Rate each alternative using a scale of choice (e.g., 1–9) in which the lower value on the scale represents a less preferred value and the higher value represents a more preferred value for each factor.
6. Place the ratings by row and column in each of the cells that make up the table.
7. Place the factor weights in the "Factor Weight" column.
8. Multiply the factor weights by each of the ratings across each row and place these values in each of the table's rating cells.
9. Sum these computed ratings by column (i.e., each alternative) to generate a total score and place these values in the "Total Score"-row at the bottom of the table.
10. Select the alternative with the largest total score.

Table 1 Example of the WSM

Criterion	Weight	System A	System B
Flexibility	0.1	$0.1 \times 3 = 0.3$	$0.1 \times 1 = 0.3$
Brand name	0.1	$0.1 \times 5 = 0.5$	$0.1 \times 2 = 0.2$
Price	0.3	$0.3 \times 2 = 0.6$	$0.3 \times 1 = 0.3$
Security	0.5	$0.5 \times 7 = 3.5$	$0.5 \times 9 = 4.5$
Total score	1.0	4.9	5.3

The WSM is quite straightforward, which may explain its widespread popularity in practice. While it is fast to perform, WSM comes at the price of significant drawbacks. First and foremost among these is the fact that its results depend highly on the availability of perfect (or nearly perfect) information on (i) the form of the decision maker's preference function and (ii) criteria weights. Note that linear preference functions are assumed by default, although the phenomenon of decreasing marginal benefits is well known and accepted in economic and psychological theory. Moreover, experts are often not willing and/or, in particular, not capable of providing a priori precise weights for several criteria. As these weights directly affect the results, even small deviations may lead to a recommendation that is derived quickly but may prove to be poor. Using the WSM as the sole approach for something as critical as Web services composition may therefore turn out to be a convenient, but ultimately expensive decision.

3.2 Analytic Hierarchy Process

The AHP was introduced by Saaty in 1980 [55]. Because it provides a means for a more objective determination of criteria weights, the AHP became quite common as well (cf. [16, 30, 32, 76]). Typically, it is performed in three stages:

– *Decomposition* At the beginning of the process, the overall decision problem is decomposed in a hierarchic structure in which the top represents overall objectives and lower levels represent criteria, sub-criteria and alternatives (see Fig. 1).
– *Comparative judgments* In the next step, participants set up a matrix that compares pairs of criteria or sub-criteria at each hierarchy level. Weights are employed to provide a numerical expression of the user's preference for one of

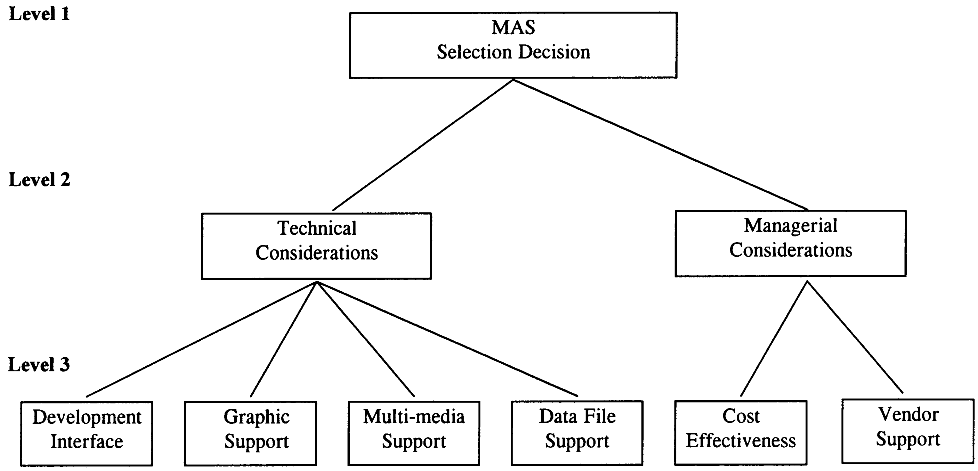

Fig. 1 Example of an AHP decision network following [32]

the two compared criteria: a rating of 9 indicates "total preference for criterion A over criterion B", $\frac{1}{9}$ indicates "total preference for criterion B over criterion A", and 1 indicates "ambivalence between criterion A and criterion B" (see Table 2).

– *Synthesis of priorities* Finally, the composite weights of all alternatives are calculated based on preferences derived from the matrix.

The AHP method's strength lies in its ability to structure the criteria of complex problems hierarchically [76]. However, it also exhibits major drawbacks. While the form of the preference function must still be determined in some manner (and, typically, is simply—and wrongly—assumed to be linear), the AHP's main drawback lies in the high number of pairwise comparisons between (sub-) criteria that it makes necessary. In addition, there are several unsolved methodological reservations (such as, why should a nine-point-scale be used and not a seven- or an eleven-point-scale, or something completely different?) that may considerably affect results.

4 The ATANA approach

The ATANA approach consists of two phases. In the first one, all portfolios (i.e., combinations of Web services) that are both feasible (i.e., fulfill all constraints) and Pareto-efficient with respect to the given objectives are determined. During the second phase, the identified solutions are subjected to an interactive screening method. By taking this approach, ATANA substitutes the AHP approach (as well as the WSM) and provides decision makers with "true" multiobjective decision support. A graphical illustration of how ATANA may be embedded in Web services selection as well as on corresponding data flows is provided in Fig. 2. Note that a previous version of ATANA has already been applied to a similar problem, namely the COTS selection [48]. This paper not only applies this approach to the selection of Web services, it also includes business processes as the basis for the valuation of Web services and provides a case study from the social security sector.

Table 2 Example for an AHP matrix [32]

Criterion	A	B	C	D	E	F
A. Device interface	**1**	1/3	1/4	1/3	5	7
B. Graphics support	3	**1**	1/3	1/4	5	6
C. Multi-media support	4	3	**1**	2	7	8
D. Data file support	3	4	1/2	**1**	6	8
E. Cost effectiveness	1/5	1/5	1/7	1/6	**1**	5
F. Vendor support	1/7	1/6	1/8	1/8	1/5	**1**

4.1 Determining efficient solutions

The first task in the ATANA approach lies in determining efficient service solution alternatives—a task that technically constitutes a multiobjective combinatorial optimization (MOCO; for a survey cf. [14]) problem. Solving this problem involves identifying Pareto-efficient combinations of services in which the binary variables $x_i \in \{0, 1\}$ indicate whether or not a service i is selected ($x_i = 1$ if so, and $x_i = 0$ otherwise). A solution can be represented as vector $x = (x_1, \ldots, x_N)$, with N being the number of proposed Web service alternatives or necessary choices between services, respectively. The MOCO problem lies in maximizing K objectives (such as functionality, usability or supplier capability). The objective functions can be written as follows:

$$u_k(x) = \sum_{i=1}^{I} b_{i,k} x_i + \sum_{j=1}^{J} g_j(x) c_{j,k} + \sum_{j=1}^{\hat{J}} h_j(x) d_{j,k} \quad (1)$$
$$\text{for } k = 1, \ldots, K,$$

where $u_k(x)$ stands for the overall benefits of service portfolio x with respect to category k. Therein, the first

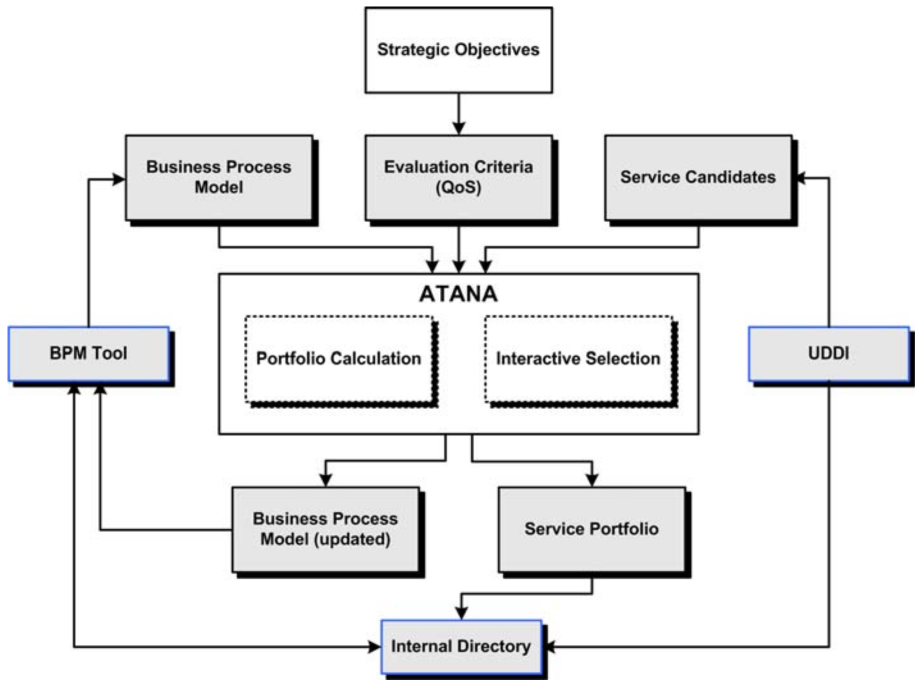

Fig. 2 Overview of ATANA's data flows

sum gathers the individual benefits $b_{i,k}$ of services i. The second sum concerns interrelation effects: additional benefits $c_{j,k}$ arise if portfolio x contains at least a number m_j of services that are elements of some interrelation subset A_j of all services. If this is the case, then $g_j(x) = 1$, otherwise $g_j(x) = 0$ (for a formal description of how function values are determined see below). The third sum gathers benefits $d_{j,k}$ that occur if at most a given number of services of some interdependency subset $\hat{A}_j$ is contained in x, where $h_j(x)$ is defined in a manner analogous to $g_j(x)$. Following prior work by [62], ATANA determines $g_j(x)$ by means of an auxiliary function

$$f_j(x) = \sum_{i=1}^{I} a_{i,j} x_i \quad \text{for } j = 1, \ldots, J. \tag{2}$$

This function counts the number of services in portfolio x that are affected by an interdependency j (with $a_{i,j} = 1$ if i belongs to A_j and $a_{i,j} = 0$ otherwise). Given m_j as the necessary minimum number of services of some type, the relations

$$f_j(x) - m_j + 1 \leq I g_j(x) \leq f_j(x) - m_j + I \quad \text{and} \tag{3}$$

$$g_j(x) \in \{0, 1\} \quad \text{for } j = 1, \ldots, J \tag{4}$$

ensure: if $f_j(x) \geq m_j$ then $g_j(x) = 1$, and if $f_j(x) \leq m_j$ then $g_j(x) = 0$, as required. With $h_j(x)$ instead of $g_j(x)$ and a change of signs in equation (3) before both $f_j(x)$ and m_j we define interdependencies that do not allow for more than a maximum number of services.

Observe that objective functions referring to criteria that should naturally be minimized (e.g., costs) can easily be transformed by simply multiplying the underlying objective values with (-1). Further note that functions $u_k(x)$ may take any form (linear, non-linear, etc.) as long as they are defined for all (feasible) alternatives x. Finally, it should be recognized that finding proper functions for criteria such as the expected availability of a given combination of Web services may prove tricky: however, this difficulty also holds true to the same degree for all other decision support approaches. Any procedure applied in this phase is meant to identify (or, at least, provide an approximation of) the set of all solutions that are Pareto-efficient. Of course, all solutions taken into consideration must be feasible with respect to two sets of constraints. The first set relates to limited resources (e.g., initial costs or running costs). These constraints may be formulated simply as

$$\sum_i r_{iq} x_i \leq R_q \quad \text{for } q = 1, \ldots, Q, \tag{5}$$

where r_{iq} represents the amount of resources of type q required by service i and R_q stands for the maximum available amount of resources. The second set ensures that at most a maximum—or at least a minimum—number of Web services from given subsets is included in feasible solutions. For instance, a constraint may require that at least two defined services (referring to the corresponding services having assigned indices 1–6) but not more than four services must be selected and, thus, takes the form $2 \leq \sum_{i=1}^{6} x_i \leq 4$. Accordingly, decision makers can assure that certain Web services should only be selected in combination with each other (e.g., two services from the same vendor might be preferred) and/or they can take into consideration that their combination yields synergy effects (e.g., the use of the two services from the same vendor might result in reduced costs). Other services are mutually exclusive (e.g., services that provide exactly the same functionality) or cause cannibalism effects (e.g., the use of a service fulfilling only part of the needed functionality might demand the use of a second service and thus would result in higher costs or reduced performance).

4.2 Interactive exploration of solution space

In ATANA's second phase, the decision maker requires support in making a final determination of the solution that best fits his notions out of the possibly hundreds (or even thousands) of Pareto-efficient alternatives identified in the first phase. Literature provides several methodologies for this task, namely (i) filtering methods, (ii) clustering methods, and (iii) search based procedures. Filtering methods reduce the efficient set by discarding the most "redundant" portfolios while retaining solutions which are most dissimilar. An application of such an approach in the field of IT risk management is provided by Strauss and Stummer [60], who used a k-tree data structure (cf. [65]). Clustering can be described as the process of forming groups of similar portfolios. Once a manageable number of clusters has been identified, a representative point from each cluster is presented to the decision maker, who can then choose the most preferred of these solutions and, in a second step, examine a neighborhood around this point. For a recent application in locating and sizing medical departments, confer a paper by Stummer et al. [64], who implemented the commonly used (non-hierarchical) k-means clustering for this purpose. Search-based procedures, finally, start from an efficient portfolio and allow the decision maker to iteratively "move" in solution space towards more attractive alternatives until no "better" portfolio can be found (cf. an application by Focke and Stummer [19], who provide an example from strategic technology planning in hospital management).

The ATANA approach is of the latter type. During the design phase, various alternative techniques for visualizing multidimensional data came into consideration (for a brief survey see [24]). Some methods represent solution alternatives as dissimilar objects—such as faces [11] or houses [31]—whose individual attributes correspond with the respective criteria values. However, it turned out to be difficult to find user-friendly mechanisms that allow for the interactive manipulation of aspiration levels within these visualizations. Moreover, interactive decision maps [37] and spider diagrams such as the "navigation star" [67] have been discussed as well. While the first-mentioned theoretically can deal with six or even more objectives, this comes at the price of a considerably increased cognitive burden imposed on the decision maker. Note that in our sample application six criteria have to be considered where two criteria would be represented on axes, two as cells in a matrix, one by color shading and one through animation and it would be quite difficult to determine which criteria to handle in which way given that this may affect the results of the interactive portfolio selection process. The interactive navigation star, however, constituted an appealing alternative and only after having received feedback from practitioners we decided to use bars instead because managers are much more familiar with bar diagrams and therefore seem to prefer them.

ATANA is based on interactive modifications of lower and upper bounds for one or more objectives. To this end, K subwindows are displayed (cf. Fig. 3).

Each subwindow (see Fig. 4) focuses on a single objective and provides the following types of information:

- Values that may be achieved by at least one of the efficient solutions identified in the first phase: These values are represented by small horizontal marks (referred to as [I] in Fig. 4) that remain fixed throughout the interactive procedure since no change in the set of efficient solutions takes place in this second phase. Observe that the marks may visually grow together to vertical blocks when several solutions provide similar values in this objective.
- The range of objective values being achievable by solutions that have remained after the decision maker has set lower and/or upper bounds for some objectives (this will be illustrated by means of a small example in

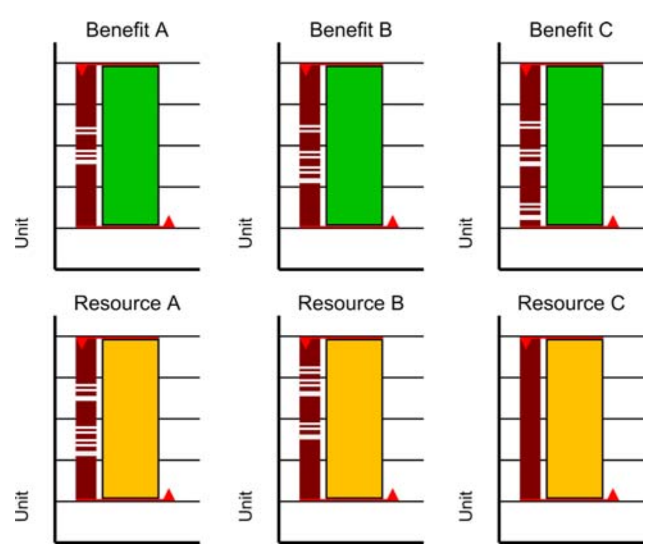

Fig. 3 Status of the DSS at the beginning

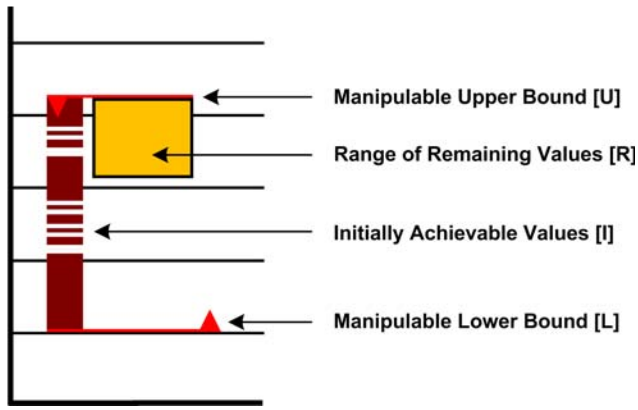

Fig. 4 Subwindow details

the remainder of this section): This range is represented by "flying" bars (see marker [R] in Fig. 4). Once the decision maker tightens or loosens some bounds (see below), this range will be automatically adapted with respect to the new settings.

– Lower and upper bounds (i.e., aspired minimum/maximum benefits): These bounds are represented by two manipulable (moveable) horizontal lines with small arrows at one side (see [L] and [U] in Fig. 4). They can be used to restrict the set of remaining solutions in a step-by-step manner (e.g., by raising the minimum bound in one of the objectives) or for expanding it (e.g., by once again relaxing some other bounds) according to the decision maker's preferences. Such a manipulation immediately affects the "flying" bars [R]. If, for example, the decision maker raises the aspiration level in one objective, the number of remaining efficient solutions (i.e., the number of solutions that provide higher benefits than the new aspiration level) will be reduced. Thus, the ranges of available objective values in this and other objectives will become narrower as well.

The system provides immediate feedback about the consequences of such choices in terms of the remaining alternatives. Let us illustrate this by reducing the maximum allowance for resource A (cf. Fig. 5).

While this setting has eliminated those solutions that come with a relatively high value in "Resource Category A" (and, on average, a somewhat higher need for resource C), the options in the other objectives have been reduced as well and the position and size of the flying bars (i.e. remaining solutions) have changed accordingly. Raising the minimum value for Benefit A (e.g., functionality) narrows the set of remaining alternatives even further, since many alternatives with low resource values (e.g., price) drop out (cf. Fig. 6).

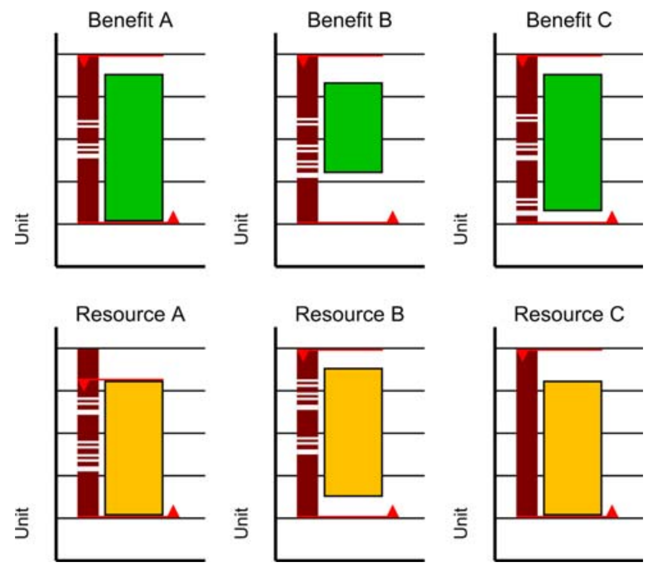

Fig. 5 Status of the DSS after the first setting

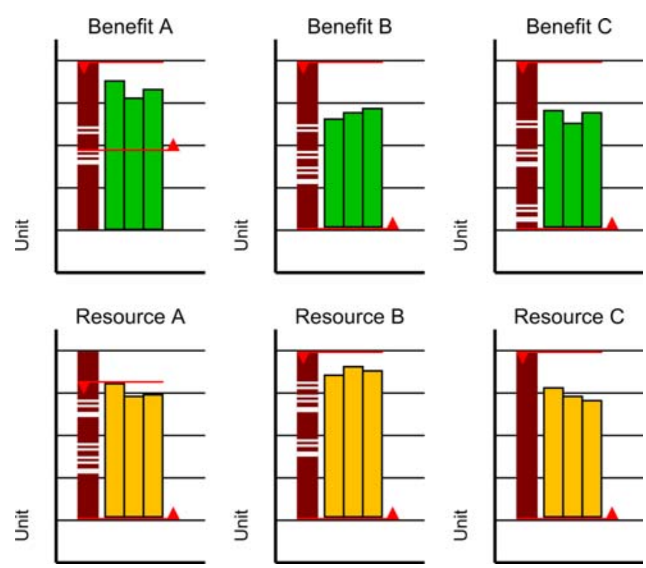

Fig. 6 Status of the DSS after two settings

In further iterations, the decision maker continues playing with minimum and maximum bounds and by doing so can learn about the consequences of his decisions and, thus, gain a much better "feeling" for the problem in terms of what can be achieved in some objectives at what "price" in terms of opportunity costs in other objectives. After several cycles of restricting and once again expanding the opportunity set, the decision maker will finally end up with a solution alternative that offers an individually satisfying compromise between the relevant objectives. Note that he does not need to explicitly specify weights for objectives nor to specify the form of his preference function or to state how much one solution is better than another during any stage of the whole procedure. Instead, ample information on the specific selection problem is provided to him and the system ensures that the final solution will be an optimal (i.e., Pareto-efficient) one, with no other feasible solution available that is "better" from an objective point of view.

4.3 Excursion: determining business process coverage

Business process coverage plays a major role when setting up a portfolio of Web services. Note that while this holds true irrespective of which decision support approach (e.g., WSM or AHP) is used, ATANA also provides procedural support in this respect. Also note that it may prove to be problematic dealing with information such as that a given service supports a specific business process activity or business process, e.g., "by 70 percent", because there are no general rules for determining the actual business process coverage of a portfolio of Web services holding two such services. Obviously, there's no coverage of 140%, but total coverage may even be as low as 70% (in the event that both services cover the exactly same function requirements) or somewhat in between up to 100%. That is the reason why ATANA supports the modeling of business processes in nearly arbitrary granularity and, consequently, other than that allows only binary n-to-n relations between services and business processes (i.e., services do or do not support a certain business process activity or business process). The mapping between services and processes (cf. Fig. 7 for a (small) illustrative example) can be represented in a binary $N \times M$ matrix where N represents the number of service candidates and M the number of processes and/or activities that needs to be covered. Each entry c_{ij} (for $i = 1, \ldots, N$ and $j = 1, \ldots, M$) holds the information whether a service i covers a process j (in this case $c_{ij} = 1$) or not (if $c_{ij} = 0$). The coverage of a business process j by a given portfolio x therefore can be determined as

$$c_j(x) = \min\left\{ \left(\sum_{i=1}^{N} c_{ij}\right), 1 \right\}. \quad (6)$$

In its simplest form, the total business process coverage results in

$$c(x) = \frac{\sum_{j=1}^{M} c_j(x)}{M}. \quad (7)$$

As a matter of course, this approach can be further refined by taking into consideration the business contribution ("value") of a process (e.g., measured in number of requests or another indicator reflecting the process' relevance) or by setting up a hierarchy (e.g., coverage of a specific process requires coverage of other processes on a lower or higher level, respectively).

5 Case study

The case study was carried out in the social security sector in Austria. The goal of the organization was the consolidation of the existing system architecture based on a given set of business processes that serve up to 500,000 customers. Therefore, a set of existing core business processes was selected and used as the basis for defining the requirements the services had to fulfill. Further, the decision makers provided a list of potential services, as well as a set of evaluation criteria. With this data serving as input, we analyzed the capabilities of the three approaches under consideration.

5.1 Selection of evaluation criteria

The criteria set includes typical QoS objectives taken from literature (cf. [34, 57, 68, 77, 78]), as well as financial

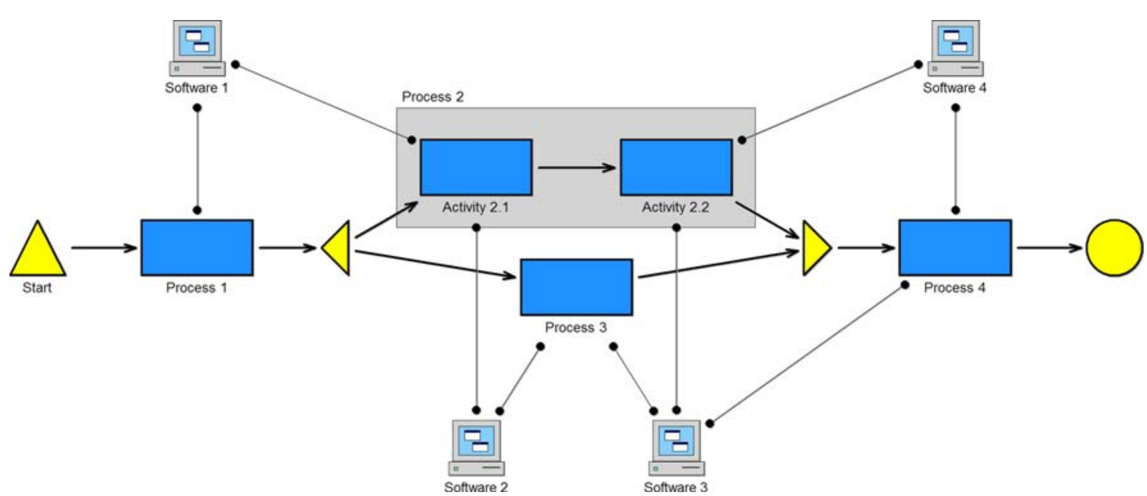

Fig. 7 Mapping between business process activities and services

criteria. Note that these criteria are just a representative selection and should be adapted by the decision makers to the specific decision scenario (e.g., in order to allow the consideration of business partnerships, or legal agreements). Further note that this section only regards non-functional criteria because functional criteria are considered by using the mapping of the services to business processes (cf. Sect. 5.2). The tool guarantees that only portfolios are selected that completely fulfill the given business requirements.

– Availability of a service i is defined as the probability $q_{av}(i)$ that a service is available measured as the total time service i is available within a time period T and dividing by T where T is a constant that is chosen by the administrator. For setting up an objective function we also may opt to take into account the expected number of service requests θ_i within time period T that can be derived from a benchmark analysis. Thus, an objective function takes the form $u_1(x) = \sum_i (q_{av}(i) \cdot \theta_i \cdot x_i)$. Essentially, this function counts the number of requests that have been fulfilled by any of the services included in the portfolio within a time period T. Alternatively, we may focus on the number of incidents for which requests have not been granted

$$\hat{u}_1(x) = (-1) \cdot \sum_i ((1 - q_{av}(i)) \cdot \theta_i \cdot x_i) \tag{8}$$

where the multiplier (-1) just ensures that this variant of the objective function is to be maximized as well. It may further be noteworthy that we have neglected any interdependency between $q_{av}(i)$ and θ_i (although a higher number of requests may increase the threat for a denial of service) in the above objective functions. Further, one can distinguish between several types of operations with differing values for the business (e.g., probably there are some requests that are considerably more harmful if they are not executed than others). However, both extensions may be easily incorporated in the model.

– The performance $q_{per}(i)$ describes how fast a request can be finished by calculating the lag between the moment of the request and the moment of the response. This value can be calculated using $q_{per}(i) = T_{process}(i) + T_{delay}(i)$, where $T_{process}(i)$ is the period of time that an instance of a service takes while being processed and is usually defined by the provider, and $T_{delay}(i)$ represents non-value-added time such as transmission time or queuing time needed in order for an instance of a service to be processed. This results in a second objective function

$$u_2(x) = \sum_i (q_{per}(i) \cdot \theta_i \cdot x_i). \tag{9}$$

Obviously, this function may be extended in an analogous way as described for the first objective function (e.g., there are usually critical core processes for which performance aspects are considerably more important than for others).

– The reputation $q_{rep}(i)$ of a service i is defined as the average rank given to the service by end users, e.g. by calculating $q_{rep}(i) = \sum_j \frac{rep_{i,j}}{\tau_i}$ where $rep_{i,j}$ represents the end users grade for a service's reputation and τ_i is the number of times that the service has been graded $(j = 1, \ldots, \tau_i)$; alternatively, one may also opt for the mode of the users' votes. The objective value of a portfolio then results in

$$u_3(x) = \sum_i (q_{rep}(i) \cdot \theta_i \cdot x_i). \tag{10}$$

Once again, this may be refined by distinguishing between more and less relevant services.

– Revenue $q_{rev}(i)$ refers to the amount of money the provider of a service i receives for its execution. Note that this may be subject to differing gains depending on the underlying operation and/or may vary depending on the times of day and/or the number of users. In its basic form the corresponding objective function can be written as

$$u_4(x) = \sum_i (q_{rev}(i) \cdot \theta_i \cdot x_i). \tag{11}$$

– The term running costs $q_{rc}(i)$ should be self-explanatory. They either depend on the number of requests, or on data volume, or a flat rate (cf. [22]). For simplicity reasons we assumed

$$u_5(x) = \sum_i (q_{rc}(i) \cdot \theta_i \cdot x_i). \tag{12}$$

– Finally, the initial costs $q_{ic}(i)$ represent the amount of money an enterprise has to invest in order to integrate a service i into its corporate environment:

$$u_6(x) = \sum_i (q_{ic}(i) \cdot x_i). \tag{13}$$

5.2 Definition of business processes and investment candidates

The existing corporate business process model that determines the whole workflow for the reimbursement of medical services, starting with the arrival of a request, continuing with request approval and proceeding to informing respective applicants, is used as basis. Nearly all activities involved in the business process require the permanent support of underlying IT systems. For this reason, software selection must be performed with a heavy focus on functional activity coverage. A set of feasible candidates was pre-selected prior to evaluating candidates using structured approaches. This selection was conducted by performing a rough selection of potential candidates and comparing their main characteristics to the decision situation's baseline parameters (i.e., knock-out criteria, such as the coverage of needed functionality), as well as approximate available monetary or performance parameters. The number of candidates to include in individual evaluation strongly depends on several factors, including application domain and the availability of suitable software—in this specific case, 20 candidates were selected. According to these preconditions and the requirements of the given business process, the services chosen for further evaluation were denoted with the letters A to Y and divided into five groups: accounting (T, U, V), approval (I, J), archiving (W, X, Y), data access (A, B, C), document management (N, O, P, R), paper digitizing (D, E, F), and text processing (L, M). In order to visualize both the functionality and business process coverage of each candidate, the business process model was updated to include the mapping to potential services to business process activities (cf. Table 3 for a listing of the activities of the business process under consideration, as well as the mapping between these business process activities and corresponding investment candidates).

As many portfolio alternatives required taking into consideration interdependencies between services, we set the following constraints:

- Service A, the existing scan software, cannot be used together with the potential new scan software, component D.
- The two DMS applications component K and component L cannot be used together.
- Component E is an extension of component D. If component D is used without component E, total usability of the respective portfolio decreases by 40 points.
- Component J is a plug-in for component I. If component I is used without component J, reliability decreases by 20 points.

Table 3 Mapping between business process activities and Web service candidates

Activity	Service candidates
Register Request	A B C N O P R
Check request for eligibility	A B C N O P R T U V
Return request to applicant	A B C L M T U V
Scan request	A B C D E F N O P R
Check for identical request	A B C N O P R T U V W X Y
Open request	A B C N O P R W X Y
Open document	A B C N O P R W X Y
Edit document	A B C L M N O P R W X Y
Mark for approval	A B C N O P R
Open marked request documents	A B C I J N O P R
Approve request	A B C I J N O P R
Check correctness	A B C N O P R
Inform applicant	A B C L M T U V W X Y

- Component P is an extension for component K. If component K is used without component P, access control decreases by 45 points.
- Initial costs must not be higher than 120,000.
- Maintenance costs must not be higher than 15,000/year.

Candidates are rated based on data taken from specifications, empirical evaluations or estimations (cf. Table 4 for the rating of all candidates). Although the applicability of our approach depends on the availability and quality of appropriate input data, this difficulty is not unique for ATANA, but rather holds for all other approaches as well. We tackle this problem by using a workshop-based process where decision makers collaboratively decide on the input values (cf. [8, 47] for an example). Note that the ranges of the ratings differ depending on whether values naturally can be measured quantitatively (e.g., monetary units, time units or resource consumption). If so, candidates are directly assigned their absolute values for this criterion. Otherwise, an abstract scale of points is applied.

5.3 Applying the three approaches

5.3.1 Weighted Scoring Method

The WSM essentially aggregates the criteria values to a single total value by using criteria weights. To this end, all values must be on the same scale in order for objective

Table 4 Ratings of the Web service candidates

Candidates	AV	Per	Rep	Rev	IC	RC
A	4	4	2	9	7	51
B	3	8	7	0	9	3
C	5	9	5	5	10	34
D	2	1	3	1	9	22
E	7	6	1	7	7	11
F	3	9	2	7	9	30
G	3	4	1	0	7	7
H	3	6	9	8	9	44
I	1	6	6	0	7	12
J	1	1	2	0	4	11
K	10	4	5	3	5	12
L	1	7	4	2	2	15
M	1	1	5	11	8	59
N	8	4	2	0	2	9
O	1	1	1	0	7	23
P	5	5	7	0	7	18
Q	3	4	4	3	8	80
R	6	2	7	0	5	14
S	4	3	3	0	8	2
T	6	4	1	5	6	23

weighting to be effective and for the final output to have any meaning. For example, when aggregating initial costs and usability, directly weighting both objectives is not effective, as the range of initial costs usually far exceeds the range of usability (which, in our decision situation, is limited to a maximum of 10). Therefore, all objective values are normalized to a range of [0,10]. Further, the WSM requires an absolute and declarative a priori weighting of all objectives. In this case study, the weights are set as follows: availability (15%), security (10%), performance (15%), reputation (10%), revenue (20%), running costs (10%), initial costs (20%). Finally, as the WSM—in analogy to the AHP—does not produce portfolios, but only candidate rankings, a knapsack algorithm is applied for portfolio composition [41].

5.3.2 *Analytic Hierarchy Process*

The AHP requires all decision criteria to be ordered in a hierarchy showing a hierarchical representation. All objectives are assigned with indices (such as "L2.1" for the first criterion on the second level) to facilitate readability. In the first step, the AHP requires pairwise comparisons of (sub-) objectives on their respective levels with preferences expressed numerically using a scale from 9 ("much more important") to 1 ("of equal importance"). Two compared objectives are always assigned reciprocal values, meaning that if objective A is rated with 4, the rating for candidate B is implied as $\frac{1}{4}$. After this, eigenvalues are calculated and the overall objective weight is derived from the average of its row sum. This process is repeated for all objectives until each of them is assigned a total weight. In order to calculate overall ratings for candidates, all of the candidates' L3 objective values must be correlated (multiplied) with all of their parent objectives up to the root node and summed up. The result is a list in which candidates are ordered by descending overall value but that does not yet point out reasonable combinations of candidates. Thus, once again a knapsack-like algorithm is applied for testing each portfolio alternative with regard to its business process coverage and its compatibility to all given constraints.

5.3.3 *ATANA approach*

The process starts by entering objectives together with their respective limits, while potential Web services and business processes, together with their mappings, are imported from the business process management tool Adonis (for an example cf. Table 3). In addition, constraints such as interactions and dependencies are defined. Depending on the number of objectives, constraints and business processes, ATANA is capable of evaluating about 40 Web service candidates per decision situation. In our case study (which includes six objectives plus business coverage, 13 business processes or business process activities and 20 software candidates) the underlying MOCO problem can be solved on an average workstation in less than 20 seconds. Thus, 222 non-dominated (i.e., Pareto-efficient) feasible portfolios are identified. These solution alternatives are further subjected to ATANA's interactive decision support module.

Figure 8 shows the initial screen of the analysis tool. By moving the red upper and lower rulers, aspiration levels are set (for minimum or maximum values in a given objective category) and, thus, the number of remaining solutions can be reduced in a straightforward manner. In our example, this is performed as follows: at first, the maximum initial costs are reduced to a value of 120k, which reduces the number of portfolios from 222 to 145 (cf. Fig. 9).

After this, the minimum requirement for availability is set to a value of 30 points, while the corresponding values for performance and for reputation are set to 35 and 30, respectively. Furthermore, the maximum value for running costs is set to a value of 47k. Afterwards, the remaining five portfolios are visualized side by side (cf. Fig. 10).

Note that the second portfolio provides not only the highest values for revenue and performance, but also the highest initial and running costs—whereas the fourth portfolio promises the lowest values in all resource categories, but also comes with the lowest benefit values. The other portfolios provide benefits on an average level and are associated with average resource consumptions (for a numerical description of the remaining portfolios cf. Table 5). Depending on whether the slightly higher costs of portfolio two appears problematic, it can either be selected or the evaluation process can be continued by picking other portfolios and/or (re-)setting the aspiration levels.

5.4 Comparison of the results

In this case study, two traditional multicriteria methods that are commonly used for selecting *single* IT investments (AHP, WSM) were compared to the new ATANA approach. The solutions constructed with WSM and AHP are both contained in the solution space generated with ATANA, which implies that these portfolios are Pareto efficient (which is not surprising, as they were optimized by means of a linear mathematical programming solver tool once criteria weights or weights for the Web service alternatives, respectively, have been determined). Note that this will probably not be the case if the WSM or AHP are used in the classical way, namely, by sequentially selecting the highest ranked alternatives. Further, the used ILP-solver made it possible to also take into account interdependencies between Web service alternatives that may remain unconsidered in a standard WSM or AHP approach.

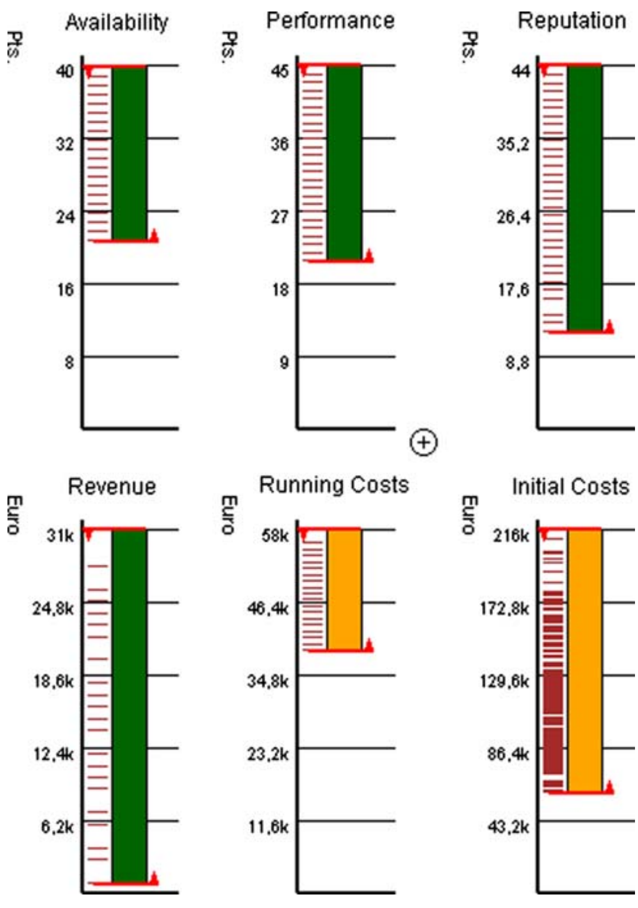

Fig. 8 Initial mask of the ATANA analysis tool

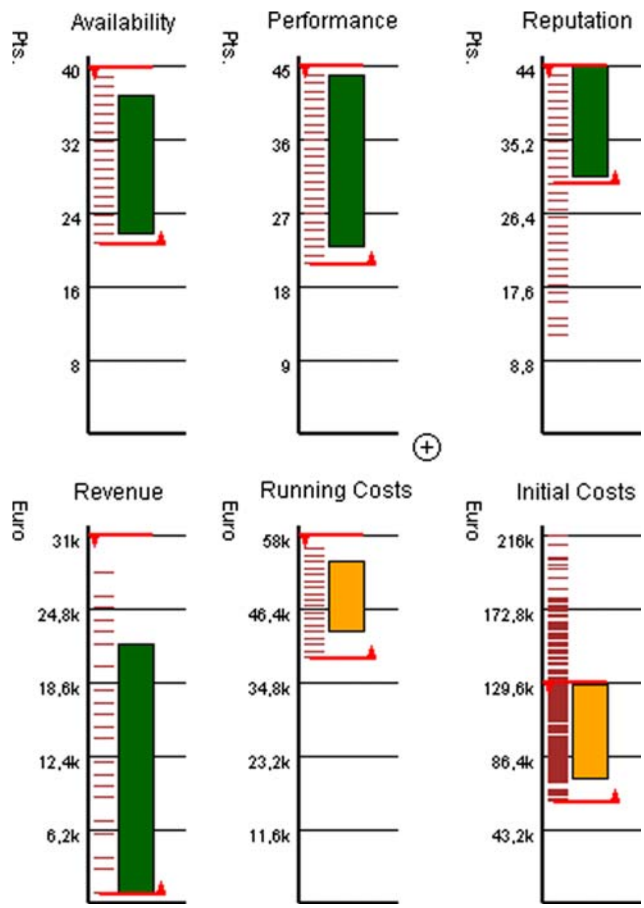

Fig. 9 Mask after the user's first setting

Apart from the fact that the quality of the results achieved is comparable if supplementing the two traditional approaches with additional Operations Research tools (otherwise it is very likely that their results are inferior), the ATANA approach provides decision makers with some major advantages that can prove to be even more important than the numerical result:

- Analytic Hierarchy Process and Weighted Scoring Method are limited to simple ranking Web service alternatives, whereas ATANA supports the construction of portfolios and, thus, also the consideration of interdependencies between Web service candidates.
- Moreover, ATANA explicitly considers given corporate business processes and therefore assures that only combinations of services are proposed that properly support the corporate business processes with regard to technical and economic requirements.
- Both traditional techniques assume linear utility functions. In addition, they require the assignment of weights to each criterion. ATANA, on the other hand, does not force decision makers to provide a priori preference information such as criteria weights or the form of their (private) utility functions. Instead, it offers a stepwise and repeatable process that provides decision makers with rich information on the composition problem at hand.
- With regard to the amount of time to be spent by the decision makers, WSM comes for the lowest costs. However, it bears significant drawbacks, such as its need for extensive a priori preference information. WSM requires decision makers to (i) have criteria weights that perfectly represent their preferences at their fingertips, without having seen any alternative solutions, and (ii) to be willing to freely provide these criteria weights to the system in order to calculate utility values for each service. In contrast to WSM, ATANA permits the stepwise definition of preferences with interactive decision support and prevents decision makers from overhasty defining constraints that could result in the loss of feasible solutions. The ATANA approach provides decision makers with the opportunity to thoroughly explore the set of Pareto-efficient solution alternatives until they find the individually most attractive portfolio of Web services, while at the same time guaranteeing that only efficient solutions are taken into consideration.

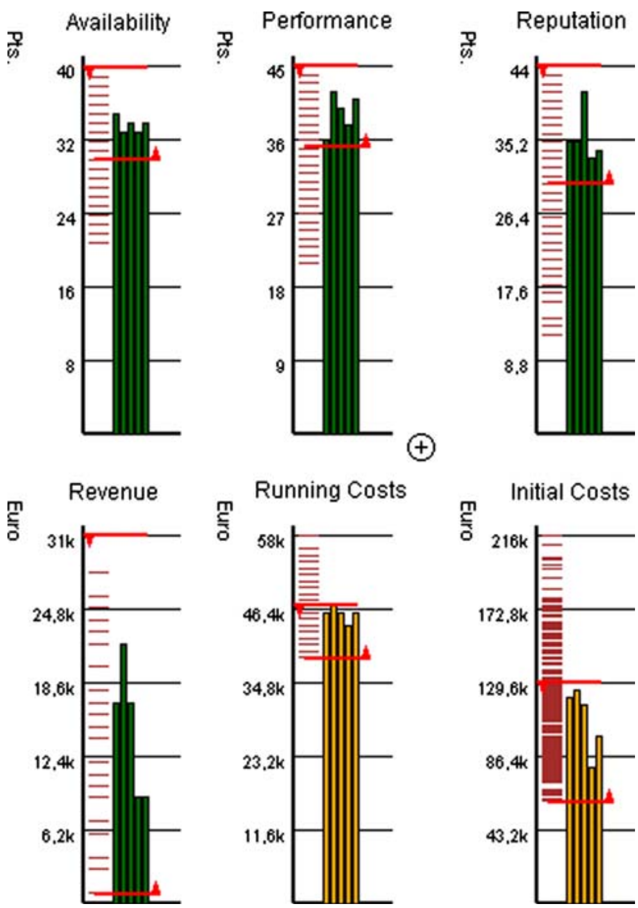

Fig. 10 Mask after further settings

Table 5 Numerical description of the resulting portfolios

Category	Sol. 1	Sol. 2	Sol. 3	Sol. 4	Sol. 5
Availability	35	35	33	38	41
Performance	83	83	83	83	83
Reputation	81	100	75	87,5	75
Revenue	365	450	400	435	405
Running costs	54	60	58	64	62
Initial costs	910	920	850	840	890

– While the results obtained by using AHP are typically more consistent and reliable than those obtained through WSM, applying the AHP method can easily result in numerous pairwise comparisons that have to be performed by the decision makers (cf. [39] for an illustrative case study). A further drawback of AHP concerns dynamic decision situations (i.e., in cases where new candidates or criteria are added or existing ones are withdrawn). In such situations, it becomes necessary to restart the whole procedure (including most comparisons) from scratch. Compared with the AHP, ATANA requires significantly less effort from the decision makers and provides additional benefits (as already outlined).

– And finally, ATANA is designed for supporting decision makers in learning more about the specific decision problem at hand; by “playing” with the system they may even learn about their own preferences.

We used the following criteria for the tool evaluation (e.g. [35]): (i) The improvement in the ability of decision makers to find their favorite solution, (ii) capturing inter-dependencies, (iii) usability (ease of use without knowledge), (iv) complexity/cognitive burden, (v) ease of exploring the solution space, (vi) efficiency (repeatable process, no a priori preference information necessary), and (vii) business process consideration.

The case study compared the decision making process of ATANA to the WSM and the AHP. In Table 6, we measured the chosen criteria using a three point Likert scale in the case the decision maker gave his expertise about the process (criteria 1, 3, 5, 6) and yes/no regarding criteria that measured the existence of certain characteristics (criteria 2, 4, 7).

6 Conclusions and further research

The majority of current corporate business processes are supported by IT systems. As a result, corporate success is directly affected by the composition of these systems where Web services play an increasingly important role. Decision makers are particularly challenged by having to consider multiple objectives, the optimal alignment between business processes and services as well as the cost-efficient use of available resources and interdependencies between the systems. In addressing the obvious shortcomings of existing approaches, we developed a two-phase interactive decision support approach named ATANA. In its first phase, ATANA determines those portfolios of Web services that are both feasible with respect to given constraints and Pareto-efficient with respect to the given objectives. Then, ATANA supports decision makers in interactively exploring the determined

Table 6 Summary of the approaches’ properties

Requirement	WSM	AHP	ATANA
Ability	Low	Medium	High
Dependencies	No	No	Yes
Usability	Medium	Low	High
Complexity	Low	High	Medium
Exploring	Low	Low	High
Efficiency	Low	Medium	High
BP coverage	No	No	Yes

solution space in the second phase. To the end of illustrating this new approach, a case study was carried out in the social security sector in Austria. It turned out that ATANA outperforms traditional methods such as AHP or WSM in many respects.

Evidently, the applicability of our approach is limited by the availability and quality of input data (e.g., it may prove to be difficult to find exact data for the changes in an enterprise's exposure when implementing a certain combination of Web services). However, this difficulty is not unique for ATANA, but rather holds for all other approaches as well; nevertheless it confines the applicability of formal decision support approaches to cases with the necessary information available. Another problem may arise if the set of considered services becomes too large (e.g., comprises hundreds of candidates). In this event, the exact determination of all Pareto-efficient portfolios in the first phase of ATANA cannot be performed in reasonable time. Although meta-heuristic solution procedures (e.g., [15, 63]) have the potential to considerable raise this limit, ATANA remains more or less limited to decision problems with some dozen Web service candidates. If necessary, the required reduction in the number of service candidates may be achieved by introducing an additional screening phase at the very beginning, during which those candidates are filtered that either have excellent values in at least one of the criteria or provide a promising mix of objective values (e.g., measured by applying the WSM in this screening phase).

Deriving "better" data in terms of both quantity and quality may constitute a first field for further research (e.g., accomplished by means of a workshop-based process, for an example cf. [47]). Next, ideas from group decision making and negotiation analysis might be integrated in order to elicit preferences from multiple stakeholders and to propose proper (e.g., fair) compromises (cf. [61]). These information may also be used in automated service composition approaches (for an interesting recent approach cf. [5]). And finally, we will continue conducting experiments in order to more thoroughly test alternative user-interface designs with respect to their acceptance in real-life decision environments. This includes the extension of ATANA to the field of information security in order to allow a more thorough evaluation of security investments.

Acknowledgements This work was performed at Secure Business Austria, a competence center that is funded by the Austrian Federal Ministry of Economics and Labor (BMWA) as well as by the provincial government of Vienna.

References

1. Alonso G, Casati F, Kuno H, Machiraju V (2004) Web services: concepts, architectures and applications. Springer
2. Alves C, Finkelstein A (2003) Investigating conflicts in COTS decision-making. Int J Softw Eng Knowl Eng 13(5):473–495
3. Andresen JL (2001) A framework for selecting an IT evaluation method in the context of construction. Technical University of Denmark, Technical Report
4. Ardagna D, Pernici B (2005) Global and local QoS constraints guarantee in Web service selection. In: Proceedings of the IEEE international conference on Web services, pp 805–806
5. Ardagna D, Pernici B (2007) Adaptive service composition in flexible processes. IEEE Trans Softw Eng 33(6):369–384
6. Becker J, Kahn D (2003) The process in focus. In: Becker J, Kahn D, Rosemann M (eds) Process management: a guide for the design of business processes. Springer, pp 1–12
7. Berbner R, Heckmann O, Mauthe A, Steinmetz R (2005) Eine dienstgüte unterstützende Webservice-architektur für flexible Geschäftsprozesse. Wirtschaftsinformatik 47(4):268–277 (in German)
8. Boehm B, Gruenbacher P, Briggs RO (2001) EasyWinWin a groupware-supported methodology for requirements negotiation. In: Proceedings of the international conference on software engineering, pp 720–721
9. Canfora G, Penta MD, Esposito R, Villani ML (2005) QoS-aware replanning of composite Web services. In: Proceedings of the international conference on Web services, pp 121–129
10. Cao L, Li M, Cao J (2005) Cost-driven Web service selection Using genetic algorithm. Workshop on Internet and network economics. LNCS, vol 3828. Springer, pp 906–915
11. Chernoff H (1973) The use of faces to represent points in K-dimensional space graphically. J Am Stat Assoc 68(342):361–368
12. Claro DB, Albers P, Hao J-K (2005) Selecting Web services for optimal composition. In: Proceedings of the 2nd international workshop on semantic and dynamic Web processes, pp 32–45
13. Cobb CG (2003) From quality to business excellence: a systems approach to management. ASQ Quality Press
14. Ehrgott M, Gandibleux X (2000) A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spectr 22(4):425–460
15. Ehrgott M, Gandibleux X (2004) Approximative solution methods for multiobjective combinatorial optimization. Top 12(1):1–63
16. Elzinga DJ, Horak T, Lee C-Y, Bruner C (1995) Business process management: survey and methodology. IEEE Trans Knowl Data Eng 42(2):119–128
17. Farbey B, Land F, Targett D (1993) How to assess your IT investment: a study of methods and practice. Butterworth-Heinemann
18. Fensel D, Bussler C (2002) The Web service modeling framework WSMF. Electr Commer Res Appl 1(2):113–137
19. Focke A, Stummer C (2003) Strategic technology planning in hospital management. OR Spectr 25(2):161–182
20. Freitas AA (2004) A critical review of multi-objective optimization in data mining: a position paper. ACM SIGKDD Explor Newsl 6(2):77–86
21. Gao A, Yang D, Tang S, Zhang M (2006) QoS-driven Web service composition with inter service conflicts. In: Proceedings of the 8th Asia-Pacific Web conference. LNCS, vol 3841. Springer, pp 121–132
22. Gouscos D, Kalikakis M, Georgiadis P (2003) An approach to modeling Webservice QoS and provision Price. In: Proceedings of the 4th international conference on Web information systems engineering workshops, pp 121–130
23. Hirschfeld R, Kawamura K (2006) Dynamic service adaptation. Softw Pract Exp 36(11):1115–1131
24. Hodgkin J, Belton V, Koulouri A (2005) Supporting the intelligent MCDA user: a case study in multi-person multi-criteria decision support. Eur J Oper Res 160(1):172–189

25. Hopkins J, Kessler M (2002) Companies squander billions on tech. USA Today 4(351):1A
26. Ittner CD, Larcker DF (2003) Coming up short on nonfinancial performance measurement. Havard Bus Rev 81(11):88–95
27. Jaeger MC, Ladner H (2005) Improving the QoS of WS compositions based on redundant services. In: Proceedings of the international conference on next generation Web services practices, pp 189–195
28. Kalepu S, Krishnaswamy S, Loke SW (2003) A QoS metric for selecting Web services and providers. In: Proceedings of the 4th international conference on Web information systems engineering, pp 131–139
29. Koch C (2005) A new blueprint for the enterprise. CIO Magazine 5(4):1–8
30. Kontio J, (1995) OTSO: a systematic process for reusable software component selection. University of Maryland, Technical report, December
31. Korhonen P (1991) Using harmonious houses for visual pairwise comparison of multiple criteria alternatives. Decis Support Syst 7(1):47–54
32. Lai C, Gong L, Koved L, Nadalin A, Schemers R (1999) User authentication and authorization in the java platform. In: Proceedings of the 15th annual computer security applications conference, pp 285–290
33. Lécué F, Léger A (2006) A formal model for semantic Web service composition. In: Proceedings of the 5th international semantic Web conference. LNCS, vol 4273. Springer, pp 385–398
34. Lee K, Jeon J, Lee W, Jeong S-H, Park S-W (2003) QoS for Web services: requirements and possible approaches. W3C Working Group Note 25, November
35. Little JDC (2004) Models and managers: the concept of a decision calculus. Manag Sci 50(12):1841–1853
36. Liu F, Dai K, Wang Z (2004) Applying multiple criteria decision making to improve security architecture development. In: Proceedings of the 3rd international conference on information security, pp 244–246
37. Lotov AV, Bushenkov VA, Kamenev GK (2004) Interactive decision maps: approximation and visualization of Pareto Frontier. Kluwer
38. Luftman JN, Papp R, Brier T (1999) Enablers and inhibitors of business-IT alignment. Commun AIS 1(art 11):1–33
39. Maiden NA, Ncube C (1998) Acquiring COTS software selection requirements. IEEE Softw 15(2):46–56
40. Malone TW, Crowston K, Lee J, Pentland B, Dellarocas C, Wyner G, Quimby J, Osborn CS, Bernstein A, Herman G, Klein M, O'Donnell E (1999) Tools for inventing organizations: toward a handbook of organizational processes. Manag Sci 45(3):425–443
41. Martello S, Toth P (1990) Knapsack problems: algorithms and computer implementations. Wiley
42. Matskin M, Rao J (2002) Value-added Web services composition using automatic program synthesis. In: Bussler et al (eds) Proceedings of the international workshop on Web services, E-Business, and the semantic Web. LNCS, vol 2512. Springer, pp 213–224
43. McAfee A (2004) Do you have too much IT? MIT Sloan Manag Rev 45(3):18–22
44. Menascé D (2002) QoS issues in Web services. IEEE Internet Comput 6(6):72–75
45. Navarrete F, Botella P, Franch X (2005) How agile COTS selection methods are (and can be)? In: Proceedings of the 31st EUROMICRO conference on software engineering and advanced applications, pp 160–167
46. Ncube C, Dean JC (2002) The limitations of current decision-making techniques in the procurement of COTS software components. In: Proceedings of the 1st international conference on COTS-based software systems. LNCS, vol 2255. Springer, pp 176–187
47. Neubauer T (2006) Multiobjective decision support for the business process driven allocation of service oriented architectures. Vienna University of Technology, Technical Report
48. Neubauer T, Stummer C (2007) Interactive decision support for multiobjective COTS selection. In: Proceedings of the 40th Hawaii international conference on system sciences, pp 1–7
49. Powell PL (1999) Evaluation of information technology investments: business as usual? In: Willcocks LP, Lester S (eds) Beyond the IT productivity paradox. Wiley, pp 151–182
50. Preuner G, Schrefl M (2002) Integration of Web services into workflows through a multi-level chema architecture. In: Proceedings of the 4th IEEE international workshop on advanced issues of E-commerce and Web-based information systems, pp 51–60
51. Ran S (2003) A model for Web services discovery with QoS. ACM SIGecom Exch 4(1):1–10
52. Renkema TJW, Berghout EW (1997) Methodologies for information systems investment evaluation at the proposal stage: a comparative review. Inf Softw Technol 39(1):1–13
53. Rossi P, Tari Z (2006) Software adaptation for service-oriented systems. In: Proceedings of the 1st workshop on middleware for service oriented computing, pp 12–17
54. Ryan SD, Gates MS (2004) Inclusion of social sub-system issues in IT-investment secisions: an empiricial assessment. Inf Resour Manag J 17(1):1–18
55. Saaty T (1980) The analytic hierarchy process. McGraw Hill
56. Schniederjans MJ, Hamaker JL, Schniederjans AM (2004) Information technology investment: decision making methodology. World Scientific
57. Shuping R (2003) A model for Web services discovery with QoS. ACM SIGecom Exch 4(1):1–10
58. Sreenath RM, Singh MP (2004) Agent-based service selection. J Web Semant 1(3):261–279
59. Srivastava B, Koehler J (2003) Web service composition: current solutions and open problems. In: Proceedings of the international conference on automated planning and scheduling, pp 28–25
60. Strauss C, Stummer C (2002) Multiobjective decision support in IT-risk management. Int J Inf Technol Decis Mak 1(2):251–268
61. Stummer C (2001) Fair group decisions in investment planning. OR Spectr 23(4):431–443 (in German)
62. Stummer C, Heidenberger K (2003) Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives. IEEE Trans Eng Manag 50(2):175–183
63. Stummer C, Sun M (2005) New multiobjective metaheuristic solution procedures for capital investment planning. J Heuristics, 11(3):183–199
64. Stummer C, Doerner KF, Focke A, Heidenberger K (2004) Determining location and size of medical departments in a hospital network: a multiobjective decision support approach. Health Care Manag Sci 7(1):63–71
65. Sun M, Steuer R (1996) InterQuad: an interactive quad tree based procedure for solving the discrete alternative multiple criteria problem. Eur J Oper Res 89(3):462–472
66. Teltumbde A (2000) A framework for evaluating ERP projects. Int J Prod Res 38(17):4507–4520
67. Thieke C, Küfer K-H, Monz M, Scherrer A, Alonso F, Oelfke U, Huber PE, Debus J, Bortfeld T (2007) A new concept for interactive radiotherapy planning with multicriteria optimization: first clinical evaluation. Radiother Oncol 85(2):292–298
68. Thio N, Karunasekera S (2005) Automatic measurement of a QoS metric for Web service recommendation. In: Proceedings of the Australian conference on software engineering, pp 202–211

69. Tian M, Gramm A, Naumowicz T, Ritter H, Schiller J (2003) A concept for QoS integration in Web services. In: Proceedings of the 4th international conference on Web information systems engineering workshops, pp 149–155
70. Tian M, Gramm A, Ritter H, Schiller J (2004) Efficient selection and monitoring of QoS-aware Web services with the WS-QoS framework. In: Proceedings of the IEEE/WIC/ACM international conference on Web intelligence, pp 152–158
71. Triantaphyllou E, Sanchez A (1997) A sensitivity analysis approach for some deterministic multi-criteria decision-making methods. Decis Sci 28(1):151–194
72. Verhoef C (2002) Quantitative IT portfolio management. Sci Comput Progr 45(1):1–96
73. Wanyama T, Far BH (2005) Towards providing decision support for COTS selection. In: Proceedings of the 18th IEEE Canadian conference on electrical and computer engineering, pp 908–911
74. World Wide Web Consortium (2007) OWL-S: semantic markup for Web services. http://www.w3.org/Submission/OWL-S
75. Yu P, Ma X, Lu J (2005) Dynamic software architecture oriented service composition and evolution. In: Proceedings of the 5th international conference on computer and information technology, pp 1123–1129
76. Yusuff RM, Yee KP, Hashmi MSJ (2001) A preliminary study on the potential use of the Analytical Hierarchical Process (AHP) to predict Advanced Manufacturing Technology (AMT) implementation. Robotics Comput Integr Manuf 17(5):421–427
77. Zeng L, Benatallah B, Dumas M, Kalagnanam J, Sheng QZ (2003) Quality driven Web services composition. In: Proceedings of the 12th international conference on World Wide Web, pp 411–421
78. Zeng L, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H (2004) QoS-aware middleware for Web services composition. IEEE Trans Softw Eng 30(5):311–327