International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. XXX-XXX

Introducing a Generic Concept for an Online IT-Benchmarking System

Pujan Ziaie[1], Markus Ziller[2], Jan Wollersheim[3] and Helmut Krcmar[4]

[1] Technische Universität München,
Arcisstrasse 21, 80333 München, Germany
ziaie@cs.tum.edu

[2] Technische Universität München,
Boltzmannstr 3, 85748 Garching, Germany
ziller@in.tum.de

[3] Fortiss, An-Institut der Technischen Universität München,
Guerickestr. 25, 80805 München, Germany
Wollersheim@fortiss.org

[4] Technische Universität München,
Boltzmannstr 3, 85748 Garching, Germany
Krcmar@in.tum.de

***Abstract*: While IT benchmarking has grown considerably in the last few years, conventional benchmarking tools have not been able to adequately respond to the rapid changes in technology and paradigm shifts in IT-related domains. This paper aims to review benchmarking methods and leverage design science methodology to present design elements for a novel software solution in the field of IT benchmarking. The solution, which introduces a concept for generic (service-independent) indicators is based on and refined by a comprehensive case study that involved large enterprises and has been evolved in the last three years. In this paper, we first review the theoretical background in the literature and highlight challenges within benchmarking processes in general and benchmarking of IT services in particular. We then explain how the initial design elements were extracted based on these preliminary findings as well as a comprehensive case study. The case study was conducted with a group of 15 large enterprises that were actively performing off-line IT benchmarking to enhance their organizational processes. The case study together with interviews with the supporting consulting firm helped us find out what kind of an online software solution can address the existing complexities and how. The proposed solution practically enabled the target organizations to support, ease, improve, and evaluate their IT-benchmarking process.**

Keywords: IT benchmarking, IT operations, management support, generic indicators, design science, case study**.**

I. Introduction

One of the responsibilities of IT managers is to focus on process optimization (for enhancement), harmonization (for consolidation), and standardization (for integration). This mission appears to be even more important in difficult economic times, which explains why IT benchmarking has recently attracted a lot of interest. IT benchmarking allows comparison of products, services and practices to external (or internal) reference points in order to generate information and provide insight on one's own IT performance [1]. The recent trend of IT commoditization [2] supports this development, as IT services tend to become less individual in organizations and can therefore be better compared externally, meaning with other organizations.

In order to improve one's own practices, not only finding others with similar problems and issues, but also knowing how to learn from them and their operations, is of great importance. To properly glean information for the sake of comparison, performance indicators are required which are the cornerstone of benchmarking processes. These indicators are defined based on their own data and data of other organizations. In the past 20 years benchmarking has become an accepted management practice and the topic of a rising body of literature "ranging across the academic to practitioner-oriented spectrum" as Francis and Holloway [3] described it.

Despite the widespread use, benchmarking as a management instrument is often regarded as unsophisticated and costly, particularly in small and medium size enterprises [4]. Frameworks and software solutions to facilitate benchmarking are indeed available for a wide range of topics, however, merely a few of them such as the work of Ebner et al. [5] support exclusively IT processes and operations [6, 7]. External IT benchmarking has certain characteristics and requires patience and determination in gathering the required data and expertise for evaluating and analyzing the results. Therefore, the process is often supported merely by specialized consultancies [8]. In fact, benchmarking processes not only encourage and assist companies in enhancing their operations, but also require active involvement of a knowledgeable consulting party in the process. More importantly, a tool that facilitates such a process can be used by both sides (companies and consultancies). Because scientific research covers only one part of benchmarking as an approach to generate information on IT performance [6], a comprehensive case study within a design science approach [9] was conducted in order to gather and evaluate the needs of all actors together with the requirements and prerequisites involved in a sample benchmarking process. The raison d'être of this paper is to establish a research foundation and also provide design elements of a software tool (solution) in order to support the IT benchmarking process and remove the barriers that scare IT managers away from applying benchmarking methods.

This paper is structured as follows: in section II, we review the salient theoretical aspects and concepts together with the studied challenges and provided solutions in the field of benchmarking in the literature. We then present our methodological approach in section III, which is deriving design elements based on a case study with 15 large organizations that actively benchmark their IT operations. In section IV, based on the gathered information from the case study, we explain how software design elements were defined and later refined to introduce a generic online solution. Finally, we discuss benefits, potential limitations, extension possibilities and research contributions.

II. Related Work

A. *Theoretical aspects of benchmarking*

From the outset, benchmarking has been a method for identifying issues of an organization's activity that could be more efficient and/or effective by comparing it with other similar activities [3]. Benchmarking can be classified according to its purpose into qualitative and quantitative benchmarking [10]. Camp [11] derived a formal definition for the term benchmarking from his experience at Xerox Corporation in the US and referred to benchmarking as "...the search for industry best practices that lead to superior performance". Watson [12] refined and extended Camp's definition by considering benchmarking a continuous search for, and application of, significantly better practices that lead to superior competitive performance. In a nutshell, benchmarking can be seen as a performance management and monitoring instrument. In this sense it can be classified as decision support system, with an underlying architecture as shown by Al-Qaheri and Al-Mejren [13].

There is still no consistent theoretical determination of benchmarking as of today [14], but rather different classifications of benchmarking approaches that have evolved over time. Considering benchmarking partners as the focus point [11] benchmarking can be classified into four types as listed in *Table 1*.

Type	**Description**
Internal benchmarking	Compares similar operations within one organization
Competitive benchmarking	Compares with direct competitor's performance. Common triggers of such re-evaluation include observable customer-facing factors such as defect rates.
Functional benchmarking	Compares some common elements of business practice performed by non-competitive organizations for a particular practice. Common elements such as the use of information technology, administrative or logistical processes allow cooperation between organizations.
Generic benchmarking	Compares business practices of one organization with those of other organizations that are more performant. Comparisons are conducted irrespective of the type of industry or location.

Table 1 Four major types of benchmarking [11]

A similar picture becomes apparent when looking at a benchmarking process and its fundamental characteristics. As shown by Drew [15] and Spendolini [16] most authors treat benchmarking as a process of change management and therefore, their models are generally comprised of the following five basic steps, which together form a benchmarking process: 1) determining what to benchmark, 2) forming a benchmarking team, 3) identifying benchmarking partners, 4) information collection and analysis, and 5) actions that involve the transfer and integration of best practices.

Based on benchmarking experience with over 4000 cases, Yarrow pointed out that comprehensive business practices are always needed in order to acquire high levels of performance and facilitate the possible new technologies [17].

Several challenges such as hidden costs [18] have been pinpointed in the literature when it comes to benchmarking. One of the most important issues is that benchmarking carries serious strategic risks, such as exposing sensitive organizational data to competitors [19]. A similar point was raised by Francis & Holloway [3] when they questioned how benchmarking is affected by legislation and corporate policies on information management. This is an important issue to take into account when designing a software solution. May et al. [20] describe some similar challenges in the context of online learning.

The mathematical delimitation and comparability of indicators is another important and critical success factor. Particularly in external benchmarking, it is required that all organizations' actions be coordinated in order to obtain meaningful and comparable results. Choosing operationalized indicators and structuring them into a consistent and comprehensible indicator set presents another challenging task [21].

B. Practical aspects of benchmarking

When applied to IT operations, benchmarking similarly enables organizations to systematically observe objects such as experiences, practices and knowledge of other organizations regarding IT processes for their own interests and concerns [22]. In this paper, we refer to benchmarking objects as *services*. As in any other form of benchmarking, defining the subject of study is the first and the most crucial step in benchmarking initiation. Since operations are generally managed from a functional perspective, mapping business processes is indeed a fundamental procedure for understanding the flow of information and resources through business processes. This further contributes to the assessment of performance of operational and supporting processes [23].

IT services (objects of IT benchmarking) aim at describing the performance of IT deliverables to all involved actors, including customers. To do so, each service should encompass certain sets of deliverables and infrastructure components [8]. Moreover, these services can be structured in a way that would reflect both the perspective of service suppliers and that of the customer [24]. This modularization allows clear-cut IT services, service modules and service elements which are coherent to each other and at the same time are consistently linked to services on other layers. **Figure 1** illustrates the resulting structure and relations.

As each layer can be implemented independently, this structure can be used to externally benchmark certain IT operations such as Backup or Email services.

Another hurdle in IT benchmarking is the inconsistency of data. Each organization has its own way of collecting data, let alone different units and measurement tools/standards. Leveraging an online tool with sophisticated techniques of defining indicators provides the participating organizations to have a framework to collect inter-consistent data and enables them to receive more reliable and normalized results. Many attributes that are proposed in Chapter IV are based on putting our prototype into practice and receiving feedback from end-users based on their experience with the tool.

Finally, same indicators may be leveraged by different services. In our original concept [25], we defined indicators for and within each service. In practice, this leads to data redundancy und unnecessary administrative burden, for the same value had to be entered for each service individually. Based on the feedback of users (organizations and the consulting firm), a concept for generic indicators was introduced. In this concept, indicators are defined separately or "service-independent" (*generic indicators*), and will be assigned to each service upon request (*service indicators*). In Section VI, the concept is explained in more detail.

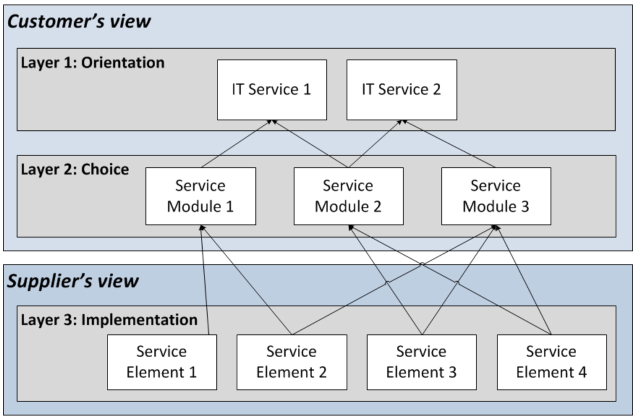

Figure 1. The layer structure for IT services [24]

III. Methodology

Design science methodology [9] was used to define the requirements and design the proper software elements and architecture. We created the artifact and evaluated it based on experts' knowledge to solve two major problems in IT benchmarking which are high organizational costs and low quality and/or inconsistent data. In our approach, we went through all possible infrastructures and determined their utility and constraints based on end-users' feedback. The results can be deemed as a search for satisfactory solutions [26] and can be qualified as credential knowledge [27], for it was not possible in practice to explicitly specify all possible solutions. With respect to the five different types of IS theory proposed by Gregor [28], our research can be categorized as theory for design and action. It provides a decision-supporting tool to a community of users (IT decision makers) with persuasive results qualified by a prestigious group of experts, namely the CIOs of fifteen large organizations.

The information was collected based on literature review and a case study with the aforementioned companies, each having more than a billion dollars in revenue. All of these companies regularly benchmark their IT operations externally with the guidance of a consulting firm. The consulting firm frequently organizes workshops for the benchmarking companies and overtakes organizational tasks such as qualifying the data, checking that the indicator descriptions are up-to-date, and ensuring that the benchmarking results are edited and presented properly. The participating companies not only compare their indicators, but also learn from each other's practices underlying those data. We investigated the benchmarking process not merely from the consulting firm's perspective but also from their clients' to glean the necessary requirements and concerns for a supporting tool.

While conducting our empirical work, we used existing guidelines outlined by Dubé and Paré [29] to ensure consistency and allow reproducibility and the ability to generalize when using only one single case. Data was collected systematically via observations, content analysis and interviews as sources of evidence in a one-year period to allow data triangulation [30].

In the first three months, a prototype was developed based on the gleaned information. Afterwards, in the following nine months, the first version of the software was refined based on users' (companies') feedbacks and confronted issues [25]. Interviews were mainly conducted with stakeholders of the consulting company to obtain detailed requirements concerning administrative needs to support the complete benchmarking process. The second version of software was developed in the next phase, where generic indicators were introduced and leveraged. In each phase, to negotiate and validate the collected requirements, we followed the approaches outlined by Kotonya & Sommerville [31] for assuring consistency, completeness and accuracy through requirement reviews and prototyping. Notes regarding each observation together with interview summaries were jointly analyzed with the results obtained from content analysis to fine-grain the findings.

IV. Case Study Findings

The first phase of the study started in January 2009. It should be noted that all organizations where asked to divide their IT operations into several benchmarking services (IT services, service modules, service elements), and each of them should go through the same benchmarking process, so that the finding are consistent and comparable.

A. Benchmarking Process

The process follows the steps outlined by Spendolini [16] mentioned in the first section and are as follows. Please note that the results are summarized in the tables at the end of the paper:

1) Determine what to benchmark

The benchmarking process is applied to every IT service separately, so that each of them can focus on different aspects of the pertinent IT operations, and at the same time, they may follow the layered structure developed and described by Rudolph et al. [24] (*Table 1*).

Indicators are mapped to those levels not only to form a structure, but also to include different levels of aggregation and views. Additionally, most services consist of quantitative indicators and their corresponding details and structure, together with qualitative indicators (such as a list of possible predefined values to choose from). The indicators and their descriptions are defined by the group of organizations, whereas the fine-tuning and harmonization is performed by the consulting firm.

2) Form a benchmarking team and identify partners

Organizations interested in benchmarking any one of the defined services come together in a group. Within these groups, organizations agree on sharing their data openly and protect it from organizations outside of the group to assure protection of confidential organizational data.

3) Information collection, analysis and action

a) Information collection

After a service and its pertaining indicators are defined, a workshop takes place to discuss and clarify open aspects within the group. The data collection will then start within each participating organization. At this stage, it is common that some organizations are not able to gather certain indicators within their organization and therefore slight adjustments might be necessary to one or more indicators' definition. The coordination process in this case is done by the consulting firm. It should also be clear, whether the defined indicators already exist for other services, or are completely new. If some of the indicators are already defined for previous services, then there is no need for collecting the value, for the value already exists in another service for the same interval. In this case, the indicator should be assigned to the service from the generic indicator pool and the value will be transferred automatically (for more info see Section VI).

b) Analysis

Upon completion of the data collection, this data is quality assured by the consulting firm if all the quality criteria are met. The quality assured data is then available (released) for benchmarking-operations. The accumulated information together with the benchmarking results are then discussed and analyzed within a workshop and also passed on to senior management of the participating organizations.

c) Action

In the final phase of the benchmarking process, organizations that have achieved acceptable benchmarking results will present their underlying practices that have possibly lead to their success. This enables others to learn and acquire the insights for further improvements.

During such workshops that take place at different stages of the benchmarking process, we took part as observers to gather the requirements for software. This accumulated information was later used to support the process of introducing the design elements.

B. Collecting and analyzing the requirements

Following the outlined process for requirement engineering [31], the following steps were taken to glean and analyze the requirement:

1) Organizing service indicators into sub-segments and sub-groups

We found that the first thing to do after defining a service and its belonging indicators is to organize the identified service indicators into different segments and groups. The sub-segments encompass indicators with the same overall topic and the sub-groups within a segment divide indicators into different categories. For example, there can be three segments, namely general info, human resources, and costs, and under costs we can have hardware, software, or specific services. This segmentation not only eases the data collection process within an organization, but also provides the necessary infrastructure for a more appealing and structured presentation of data.

2) Indicators' characteristic and value attributes

For all the services, there exists a pool of *generic indicator*s. Each service is assigned specific indicators from the pool, depending on its definition. These assigned indicators can be characterized as *service indicator*s. Each generic and service indicator has certain particular characteristic (*Table 2* and *Table 3*) and value attributes (*Table 4*). The value attributes determine different values of an indicator for each organization and time period (*dataset*), while the characteristics values are fixed for every indicator and pertain to the nature of each and every one of them.

All these attributes were derived from interviewing end-users before designing and implementing the prototype as well as receiving their feedback after putting the prototype into real practice in both phases.

a) Characteristic attributes

Characteristic attributes are inherent features of indicators and can be either service-independent (generic) or service-specific. Generic characteristic attributes are describing information such as name and description, type of the value of the indicator (qualitative or qualitative, fixed or variable), activity dates (e.g. creation or modification date), general validation criteria (maximum and minimum permitted values), the status (active or inactive), and so on. For example, whether an indicator is monotonically increasing or decreasing is a generic characteristic attribute for the sake of possible automation and clustering of the existing indicator values. *Table 2* provides a list of common generic characteristic attributes.

Generic indicators

Characteristic attributes	**Description**
Name	The title/name of the indicator (e.g. number of printers)
Description	The description of the indicator
Value type	Indicators may be quantitative or qualitatitive. The value can also be variable or fixed (should be selected from a predefined list of values). Moreover, the value of indicators can either be entered manually by the users (input indicators) or be calculated automatically based on a defined mathematic formula (output indicators) (see Section VI.B for more information on output indicators).
Unit	For quantitative indicators, unit determines the assigned unit (e.g. person per month or Euro).
Dates	Dates of certain actions such as creation, modification, or deactivation (for logging purposes)
Validation boundaries	Maximum and minimum permitted value for automatic validation (e.g. for the reveneue, min value is 0 and max value can be set depending on the organizations)
Monotonicity	Whether an increase in the value is positive or negative
Status	Active or inactive (if inactive, it cannot be assigned to services anymore and the value of the corresponding service indicators is not considered for benchmarking)
File reference	The corresponding file (e.g. informational PDF file)

Table 2. Characteristic attributes of generic indicators

The value of service-specific characteristic attributes as implied by the name depends on the definition and features of the service. These values should be provided when an indicator is assigned to a service. For example, position of the indicator within the assigned sub-group of the service or whether users have to provide a value when performing data entry are two of the attributes that depend on what service an indicator is being assigned to. The "weight" attribute determines the weight of the value of an indicator in the benchmarking process. This value can be leveraged when certain formula are developed to assess the ranking of organizations within a service. *Table 3* provides a list of common service-specific characteristic attributes.

Service indicators	
Characteristic attributes	**Description**
Corresponding service-subgroup	The corresponding sub-group for this indicator within a service (indirectly determines the corresponding service)
Label	Label of this indicator within the corresponsding service
Description	The service-specific description of the indicator
Dates	Dates of certain actions such as assignment or deactivation (for logging purposes)
Position	Position of the indicator within the specified sub-group (for presentation purposes)
Required	Whether a value must be entered by users or not
Weight	Weight of an indicator determines its value compared to other indicators in the service
Validation boundaries	Service-specific maximum and minimum permitted value (for validation purposes)
Status	Active or inactive within the corresponding service (if inactive, no values can be entered and the odl value will not be considered for benchmarking)
File reference	The corresponding service-specific file (e.g. informational PDF file)

Table 3. Characteristic attributes of service indicators

b) Value attributes

For every indicator within a service, its value attributes are set for each dataset. This means that the given value depends on the organization, the time span, and the corresponding service. *Table 4* provides a list of value attributes. This list has been refined during both phases of software development and is based on the practical experience and received feedback. For example, because multiple employees of one organization work within the benchmarking team, the possibility of internal comments, which can only be viewed by employees of the same organization, had to be provided in addition to external comments that can be viewed by everyone. Moreover, organizations are also usually interested in knowing how certain projects would have impacted (or would impact) their outcome and perhaps their ranking in a certain benchmarking process. Therefore, it is of great importance for organizations to be able to enter simulation values (desired values) for each indicator and see the outcome[1].

The last attribute, value status, represents the state of the entered value with regard to the quality assurance procedure. Each indicator value may have four status: not released (entered, but not released for the verification), released (released for the verification, but not verified), accepted (verified and accepted), and rejected (verified and rejected). What status is required for a value to be used in the benchmarking process may vary and depends on the definition of a service.

Indicator value attributes	**Description**
Real value	The value of an indicator within a dataset
Desired value	Description
Internal comment	A comment on the value that can only be seen by the employees of the owner of the data
External comment	A comment on the value that can be seen by all other parties (organizations)
Dates	Dates of certain actions on values such as insertion, status update, or value update
Value status	Indicates whether the value is released by the organization to be verfied and or whether it has been accepted or rejected by the admins

Table 4. Value dimensions of service indicators

3) Different levels of access

Four levels of access (four actions) within each account were identified according to our experience: *Modifying Data, Viewing aggregated benchmarking results* (such as median value or position of an organization for an indicator), *Viewing detailed benchmarking results* (viewing the exact value of an indicator for each organization), and *Viewing other organizations' identity* (in all other actions, the organizations remain anonymous).

We also found out that organizations prefer different levels of access for their employees within their own level. This implies that a role pyramid should be defined and the role of each employee shall be inherited from the role if his or her organization.

4) Data export functions and Diagrams

The benchmarking data is available for every organization in two major forms. One of them is to have a structured Excel

[1] For further information on the use and benefits of simulation values and their application in the domain of IT Benchmarking please refer to the work of Al-Qaheri and Hasan [32] H. Al-Qaheri and M. K. Hasan, "An End-User Decision Support System for Portfolio Selection: A Goal Programming Approach with an Application to Kuwait Stock Exchange (KSE)", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 2, pp. 0-10, 2010.

file that contains all indicator values of all organizations (if they have the right to view them, of course) and another way is to generate representing diagrams. The diagrams can either represent the raw data (indicator values) in datasets, or demonstrate the results of a certain benchmarking practice.

V. The Resulting Design Elements

In the last step toward defining design elements for an IT benchmarking tool, the requirements gathered from the literature, workshops and interviews as source of evidence were mapped to apt design elements. These preliminary results without implementing a concept for generic indicators were introduced in [25] based on the first phase of development. These results were then extended in the second phase in order to address the need for generic indicators. We structured the findings according the proposed benchmarking process by Spendolini [16]. The proposed process was refined by changing the *validation of data* to *data quality*, and by adding *data entry* as one important part of the process pertaining to data quality assurance. This new dimension was added to support the interrelation of data and to address the need for generic indicators in order to increase usability and reduce data redundancy. The results are as follows.

A. Definition and team building

Our findings regarding the first step (determine what to benchmark) and the second step (finding partners and forming teams) are listed in tables III and IV, respectively.

B. Information Collection and Analysis

For the subsequent step of information collection and analysis, multiple software design elements could be found to support the benchmarking process. Our findings show that data collection and analysis are not always conducted by the same stakeholder within organizations and therefore, we consider collection and analysis to be two separate parts. The requirements together with the corresponding design patterns are shown in table V. The step of information analysis within the benchmarking process can also be supported by software, as shown in table VI. To further support the collaborative work environment the concept of wiki pages will be introduced, to harness the benefits researched by Yang [33].

C. Data Quality Assurance

The quality of data is of utmost importance in benchmarking systems, for without high-quality data the results of the process may be misleading. We considered two approaches for assuring the quality of data: on the one hand, we should make sure that the system provides necessary means to automatically prevent users from entering false or redundant data, and on the other hand, the data should go through the qualification phase, where it will be checked and verified by experts. As discussed in the previous section, expert verification is facilitated by adding an attribute to the indicator values to represent its status. This status can be changed by an expert. Regarding system elements, we built in two mechanisms in order to assure an acceptable level of quality by the system. First, the database architecture was designed in a way to support the concept of *generic indicators* which will be explained more thoroughly in the next section, and second, validation methods were implemented at the point of data entry.

1) Database architecture to prevent redundancy

In order to prevent redundancy of data, we considered the indicators to be above the services and not defined within them. This means that if two or more services contain the same indicator, the value of this indicator (for a company in a specific period) should be inserted only once. This value will then be used for other services that use the same indicator. If the definition of an indicator is defined within a service, then the value of the same indicator should be entered separately for another service, which not only puts unnecessary burden on the users, but also may result in redundancy of data. The redundancy of data causes problems such as inconsistency, when the value is edited for one service, and the old value remains for another one.

2) Validation of data at the point of entry

Another essential feature that was brought up after presenting the prototype to the customers was the necessity of data validation mechanisms for both organizations and the consulting firm. For this purpose, value-boundaries (setting minimum and maximum values as indicators' attribute) and data mining (statistical analysis of existing data) to identify possible outliers were added to the design elements.

Moreover, it became clear that some indicators are not input variables, but output functions. In this paper we call them output indicators (e.g. the total sum of a group of cost-indicators). To further reduce the amount of false data in the benchmarking process, the consulting firm suggested that these indicators should be calculated automatically and the values should be shown during data entry. By showing the calculated results (values of output indicators), the possible error in input values would become apparent to the users. In the next section, the concept of generic indicators, along with its relation with output indicators will be elaborated upon.

VI. The Concept of Generic Indicators

As mentioned in the Section IV.B.2), indicators are not defined per service, but assigned to them from a generic indicator pool. Moreover, the value of indicators can either be entered manually by users (input indicators) or be automatically calculated based on a mathematic formula (see Value type in *Table 2*). In this last section, the concept of generic indicators, its relation with output indicators, and the necessary design elements are discussed in detail.

A. *Generic Indicators*

Generic indicators were introduced as a proper solution to increase the quality of data on the one hand, and the need to reduce administrative costs on the other. Without generic indicators, the value of each indicator within every service had to be inserted and updated separately. This was not practical, since not all the indicators are specific to one and only one service and more services often use a certain number of similar indicators. If indicators are not defined generically, not only the same value should be entered for every service, but also change in the value should be applied for each service separately. This increases the possibility of human error and therefore, the potential redundancy of data.

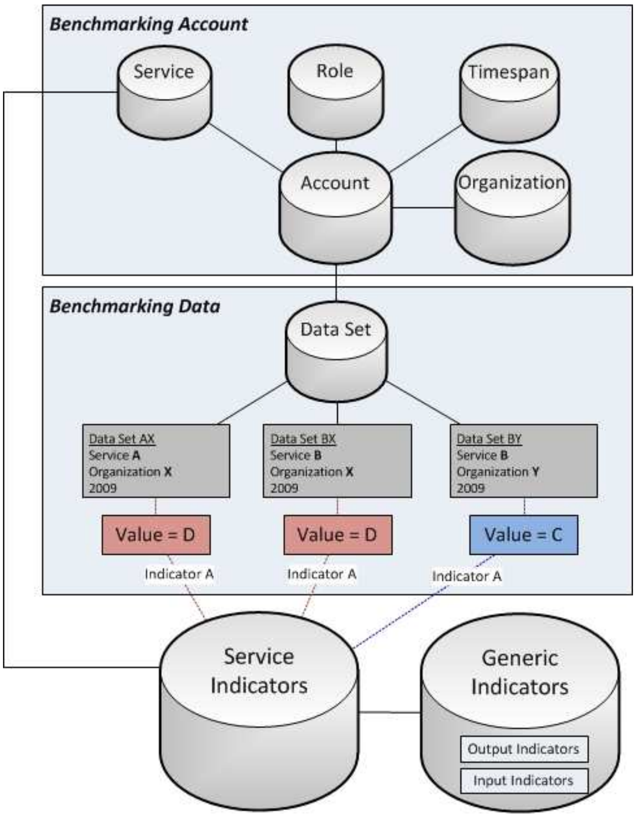

Figure 2. The concept of generic indicators

In order to keep the values of indicators independent from the using services and to ensure a high quality of content [34], a pool of generic indicator was defined. This way, whenever a service requires a certain indicator, it is first searched in the pool. If an indicator with the desired specifications already exists, then it is selected and assigned to the service. If the required indicator for a service has not yet been defined, it will be first created and added to the pool, and then assigned to the service. Following this procedure, the value of an indicator for a certain timespan and year remains the same for all services that are facilitating this particular indicator. The value should be entered for the first service to which the indicator is assigned, and this value is automatically set for the following services that leverage the same indicator. Moreover, when the value of this indicator is changed in the dataset of one service, the value for other services is updated for the same time span. The process of indicator assignment is demonstrated in **Figure 2**.

Considering the data model, generic indicators and service-specific indicators are stored in two separate entities. Each generic indicator is then related to service-specific indicators via a 1-to-N relation. Figure 5 depicts the proposed database schema for the implementation of generic indicator concept. According to this relational model, each organization has a set of accounts, where each account represents the role of an organization for a service during a certain time span (e.g. year). Each account has one or more dataset, and each dataset contains the values of the consisting indicators.

B. *Output Indicators*

The value of indicators, as mentioned in *Table 2*, can be either provided manually by the user (input indicators) or calculated by the machine (output indicators). Indicator may also have a certain unit (dimension) or may be dimensionless. The value of output indicators is calculated by giving the constructing indicators to a mathematical equation and calculating the outcome. **Figure 3** briefly illustrates this concept.

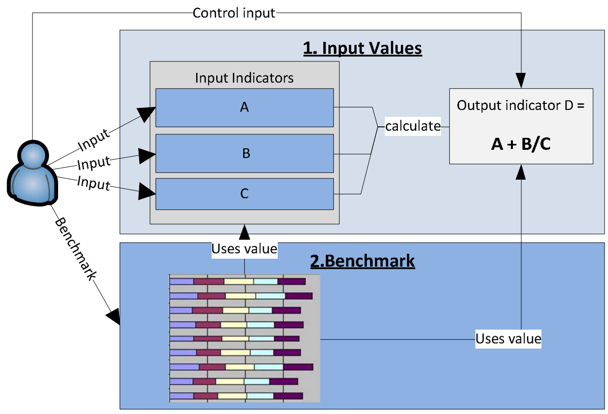

Figure 3. Interrelations of input and output indicators

Output indicators can have various purposes. For example, they can be used as a control function that would help detecting possible wrong entries by users (see **Figure 3**). This way, when, for example, a digit is skipped by a user, the provided output function would show a strange value, indicating that an error has happened during the data entry process. Although in some cases the data has to be qualified by the admins (e.g. the consulting firm in our case) before being available for benchmarking operations, this system provides a very convenient mechanism to detect faulty data in the early stages and, therefore, prevent consecutive faults. Moreover, output indicators may represent functions such as average or total costs. Entering these values manually instead of deriving them from the existing values increases the operation costs and the risk of any inconsistency when one of the entered values does not correspond to the desired output.

The second objective is to enable the users to combine different indicators to form a new indicator without any limitation other than mathematical restrictions. The values of these indicators will then be used within the benchmarking process. To address this requirement, an approach was chosen that was similarly used by Zanibbi, Blostein and Cordy [35] to convert handwritten mathematical formulas into digital data structures. This approach is to represent the function by nesting expressions in a so called tree and to calculate the output by means of basic arithmetic functions that connect elements of the tree. **Figure 4** illustrates this concept und shows how an output indicator can be composed of a series of other input and output indicators.

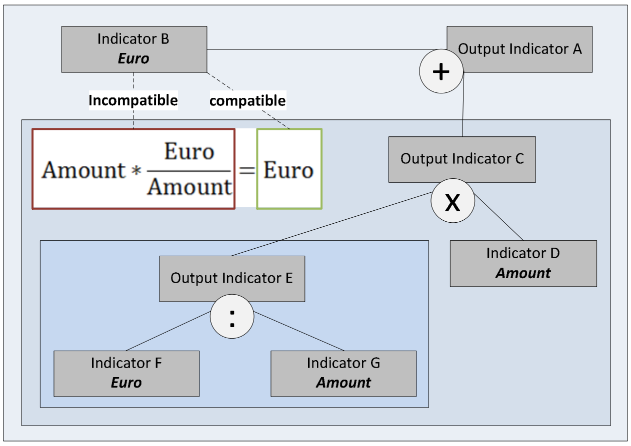

Figure 4. Tree structure representing an output indicator

With each output indicator being a subclass of an indicator and fully compatible to input indicators, this system allows the creation of tree-like structures that can provide a very comprehensive and flexible way of generating new indicators based on individual requirements. **Figure 3** shows an example of how an output indicator is created by joining different input indicators into a function. When the value of D is required, it will be derived from dividing the value of B by the value of C and adding the value of input indicator A to the result. This is a rather simple example, but when we take into account that D may serve as a part of a new function itself (as shown in **Figure 4**), it becomes clear, that this mechanism allows the creation of any complex structure of indicators.

A central point of this feature is the ability of the system to treat the output indicator as a mathematical formula and the other individual indicators (input as well as the value of other output indicators) as variables in this formula. After an output indicator has been stored, it is available for serving as a function to calculate the concrete value for given values as input. In addition, it is now also available for reuse in further output indicators as illustrated before.

To make this mechanism work properly, it is necessary to make sure that each output indicator is canceled out to the smallest possible form with respect to its corresponding formula before being stored. Taking the mathematical rules regarding cancelation, addition and subtraction of fractions and the fact that determining compatible indicators is solely based on the indicator's unit into account, the following example based on **Figure 4** clarifies why such a mechanism is needed.

1. $\text{output indicator E} = \frac{\text{Euro}}{\text{Amount}}$
2. $\text{indicator D} = \text{Amount}$
3. $\text{output indicator C} = \text{E} * \text{D}$
4. $\text{indicator B} = \text{Euro}$
5. $\text{output indicator A} = \text{C} + \text{B}$

When looking at step (3), the necessity of this mechanism becomes clear. The system has to recognize, that the product of E and D is an indicator whose unit indicates the currency Euro (Amount * Euro / Amount = Euro). If the system isn't able to recognize this, the verification of the compatibility between C and B in step (5) will fail and, as a result, the system will prohibit the creation of A, although it would be a valid new output indicator.

Without such a mechanism, the system could falsely inhibit the creation of a new output indicator and also decrease usability. The decision to use the abovementioned structure to design output indicators results in a clearer algorithm, and yet it provides flexibility for users. On the other hand, it also requires the users to have an understanding of the indicators, because it doesn't prevent users from creating output indicators that wouldn't make much sense on a logical level. The relational model of the proposed structure and its relation to generic indicators can be seen in Figure 5.

VII. Conclusion and Discussions

This paper provides a short but meticulous requirement analysis and review of the IT benchmarking domain by using design science methodology. The possibility to generalize the solution with the design elements mentioned in this paper might be limited, because we only considered one general case study, and the number of organizations involved in the case was small. However, due to studious and long-term data collection methods based on the literature review, observation and interviews, we were able to identify specific features and needs voiced by professional actors. These actors provided us with valuable insight and best practices to design a generic concept. The evolution of the proposed concept took place during two major phases. The first phase took nine months of testing and refining and the second about one year. This shows that the requirements and the provided software design elements are promising to an acceptable degree and offer a sophisticated solution to overcome many current challenges and hurdles of organizations such as data redundancy in the field of IT benchmarking.

This work also contributes to the research on IT benchmarking by providing a profound set of software design elements for four steps of the benchmarking process outlined by Spendolini [16]. This helps researchers to understand the benchmarking challenges of all the involved actors. The comprehensive list of requirements can also be used as a starting point for future research within other benchmarking groups.

Finally, the findings are a step forward to address challenges and hurdles within the IT benchmarking process between organizations, and a solution that reduces the need for consulting support since organizations are able to derive and process much of the required information by themselves. Furthermore, the classifications that have been used to develop and evaluate the design elements my not be absolutely accurate. Therefore, it might be helpful to apply the machine learning algorithms used by Eminagaoglu and Eren [36] in the domain of IT benchmarking to examine weather similar classification results can be achieved.

As for future improvements, two additional features can be scrutinized and enhanced. One is embedding the specification of graphical presentation into the relational model, and the other is the generalization of time span. For now, the time span is fixed (to one year), however, the received feedback indicates that there is a need for a more robust structure in which the time span can be dynamically configured and assigned.

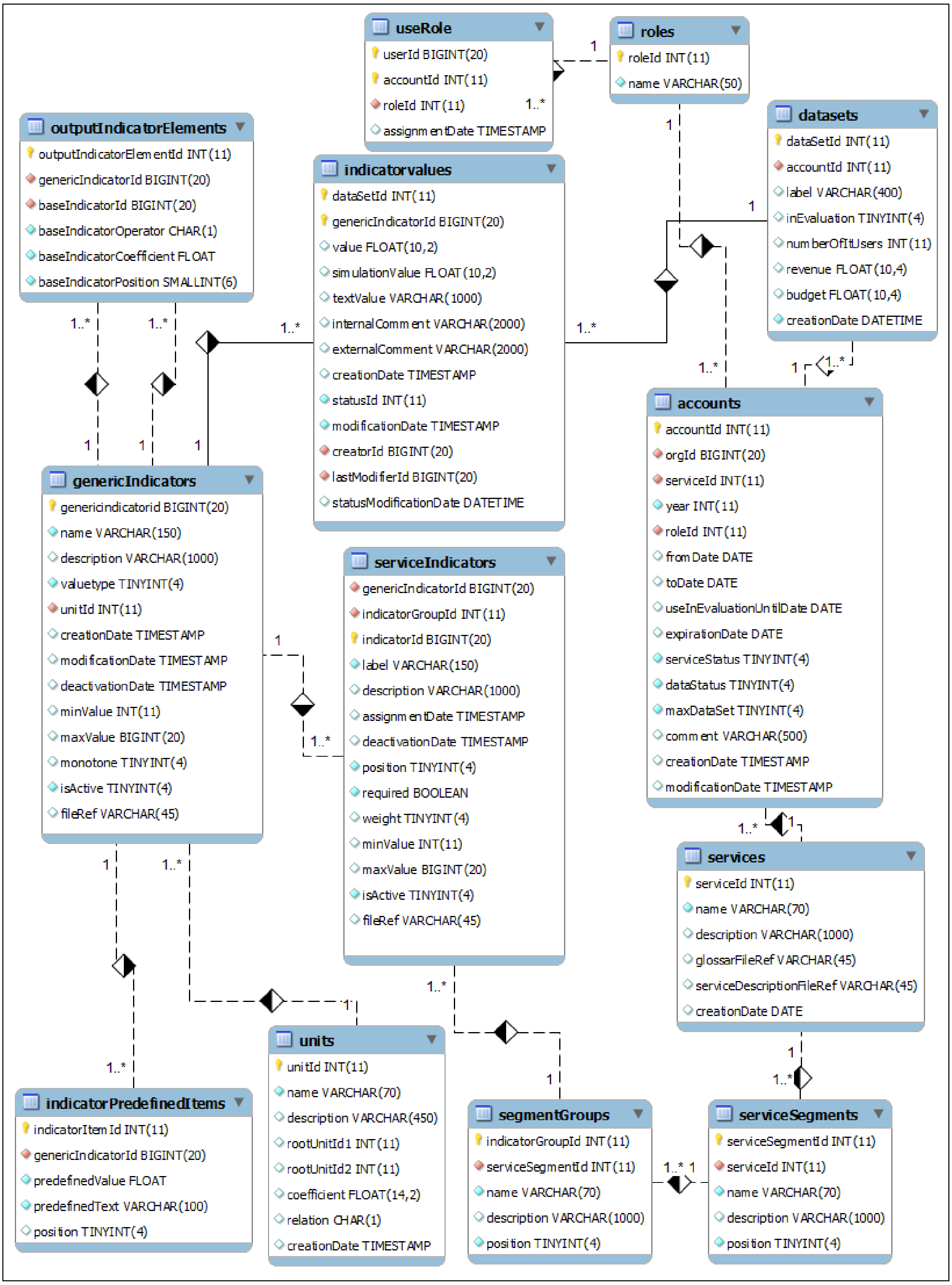

Figure 5: Relational model of a benchmarking system based on generic indicators

Findings	Design Element	Reason
Indicators need to be structured for different viewpoints and aggregation levels [24, 37]	Providing a tree-like structure for benchmarking objects	To represent layered structure of IT benchmarking objects
Data collection mechanisms within companies should be eased	Structural element of indicator groups to be added within layers	To divide indicators into groups that can be delegated to different departments
Benchmarking object and its indicators require administration	Administrative access & role based concept of user rights	So that a hierarchy is provided for accessing the data and functionalities
Two types of quantitative and qualitative indicators are used	Indicator attribute: type (text, quantitative values as well as lists of predefined values)	To categorize indicators into different types based on their attributes

Table 5. Findings: determining what to benchmark

Findings	Design Element	Reason
Organizations and their users need to be grouped into benchmarking objects	An entity called service that contains indicators, organizations, and their roles	So that users (belonging to an organization) can be assigned access to a particular service
Users require a subset of permissions of their organization	The role of a user is inherited from that of the corresponding organization	So that a user would not have a higher access level compared to that bestowed to its organization
Roles require varying permissions for data	Defining roles as a group of rights (access levels)	So that a group of rights can be labeled as a role and assigned to a user/ organization
Administration of roles per organization and user are required [3]	A graphical user interface for the administration	To provide a user interface to manage users, organizations, roles and services
Synchronization of events such as deadlines are required [7]	An announcement mechanism as well as time restriction on accessing data	To inform participants about the deadlines and restrict access after the deadlines
General organizational data needs to be specified [18]	Quantitative and descriptive criteria per dataset	So that each dataset of an organization can be specified via descriptive criteria
Information of timeframe is required per corresponding indicator value	Considering variable attributes for indicators	To provide more information on the value of an indicator

Table 6. Findings: forming a benchmarking team and identifying partners

Findings	Design Element	Reason
Enable indicators to be sorted ascending or descending	Indicator attribute: monotonicity	To know the characters of indicators and cluster them later based on their value
Enable recently changed indicators to be highlighted	Indicator attribute: last-modification time of indicator and indicator description	To distinguish the recently modified indicators from others
Descriptive comments to indicator values for other organizations should be provided	Indicator attribute: external comments per indicator, visible within benchmarking group	To provide a way to explain a value for others (externally)
Descriptive comments to indicator values for internal colleagues should be provided	Indicator attribute: internal comments per indicator, visible by colleagues only	To provide a way to explain a value for own organization (internally)
Prevent obviously out of bound entries	Indicator attribute: minimum and maximum value per indicator	To compare the values with their max-min threshold and prevent errors and enhance the quality of data
Make calculated indicators immediately visible during data collection	Additional indicator type: output function	To calculate the output indicators immediately for users
Enable user to flag data as ready for quality assurance	Indicator attribute: status, with the possible values controlled vs. not controlled	To inform administrator about the status of an indicator so that it can be processed (qualified)

Table 7. Findings: information collection

Findings	Design Element	Reason
Analysis needs to be based on reliable data only	Provide quality assurance steps	To only include qualified data in benchmarking process and outcomes
Overview of benchmarking data as well as one's own value required	Functionality to calculate aggregated benchmarking results	So that organizations get a sense of their position compared to others
Detailed benchmarking data (single values) required	Functionality to search and show detailed benchmarking results	So that organizations can compare the exact indicator values with each other
Rankings per indicator required	Indicator attribute: monotonicity	So that the position of an organization is shown based on its effectiveness
Condensed overview of results for managerial users required [6]	Report functionality for specific roles	To provide a summary of results for the top manager and prevent others from accessing them
Standardized graphical overviews per bench-marking object needed	Provide a graphical export option for the benchmarking results	To provide users with a standard, friendly and comprehensible output
Individual, graph based analysis required	Provide graphic templates for individual graph-based analysis	To provide users with an exclusive and comprehensible output
Exported data is required to analyze specific aspects in detail	Provide options to export benchmarking data	Not every operation can be performed by the tool

Table 8. Findings: information analysis

References

[1] B. Müller, *et al.*, "Understand the success of strategic IT/IS benchmarking - Results from a multiple-case study", presented at the 18th European Conference on Information Systems, Pretoria, 2010.

[2] Gartner. (2008, 10 August 2010). *Gartner Says IT Leaders must prepare for the industrialization of IT*. Available: http://www.gartner.com/it/page.jsp?id=641909

[3] G. Francis and J. Holloway, "What have we learned? Themes from the literature on best-practice benchmarking", *International Journal of Management Reviews,* vol. 9, pp. 171-189, 2007.

[4] R. Dattakumar and R. Jagadeesh, "A review of literature on benchmarking", *Benchmarking: An International Journal,* vol. 10, pp. 176-209, 2003.

[5] K. Ebner, *et al.*, "Making Strategic IT/IS Management Comparable: Designing an Instrument for Strategic IT/IS Benchmarking", presented at the Multikonferenz Wirtschaftsinformatik (MKWI), Braunschweig, 2012.

[6] P. Cragg, "Benchmarking information technology practices in small firms", *European Journal of Information Systems,* vol. 11, pp. 267-282, 2002.

[7] M. Björklund, "Benchmarking tool for improved corporate social responsibility in purchasing", *Benchmarking: An International Journal,* vol. 17, pp. 340-362, 2010.

[8] H. Krcmar, *Informationsmanagement*, 5 ed. Berlin: Springer, 2010.

[9] A. R. Hevner, *et al.*, "Design Science in Information Systems Research", *MIS Quarterly,* vol. 28, pp. 75-105, 2004.

[10] C. Legner, *Benchmarking informationssystemgestützter Geschäftsprozesse*. Wiesbaden: Deutscher Universitätsverlag, 1999.

[11] R. C. Camp, *Benchmarking: The search for industry best practices that lead to superior performance*. Milwaukee: Quality Press, 1989.

[12] G. Watson, *Strategic benchmarking: how to rate your company's performance against the world's best*. New York: Wiley Chichester, 1993.

[13] H. Al-Qaheri and A. Al-Mejren, "A Decision Support System for Plot Pricing of BO T Projects in the State of Kuwait", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 1, pp. 329-340, 2009.

[14] J. Moriarty and C. Smallman, "En route to a theory of benchmarking", *Benchmarking: An International Journal,* vol. 16, pp. 484-503, 2009.

[15] S. Drew, "From knowledge to action: the impact of benchmarking on organizational performance", *Long Range Planning,* vol. 30, pp. 427-441, 1997.

[16] M. J. Spendolini, *The benchmarking book*. New York: American Management Association, 1992.

[17] D. Yarrow, *et al.*, "Made in the 21st Century - How far have we come on the journey to excellence?", *Total Quality Management & Business Excellence,* vol. 15, pp. 829-839, 2004.

[18] D. Elnathan, *et al.*, "Benchmarking and management accounting: a framework for research", *Journal of Management Accounting Research,* vol. 8, pp. 37-54, 1996.

[19] A. Cox and I. Thompson, "On the Appropriateness of Benchmarking", *Journal of General Management,* vol. 23, pp. 1-20, Spring98 1998.

[20] M. May, *et al.*, "TrAVis to Enhance Students' Self-monitoring in Online Learning Supported by Computer-Mediated Communication Tools", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 3, pp. 623-634, 2011.

[21] J. Rehäuser, *Prozeßorientiertes Benchmarking im Informationsmanagement*. Wiesbaden: Deutscher Universitätsverlag, 1999.

[22] M. Kütz, "Ein Konzept für das IV-Benchmarking", in *IV-Controlling aktuell: Leistungsprozesse, Wirtschaftlichkeit*, L. v. Dobschütz, *et al.*, Eds., ed Wiesbaden: Gabler, 1999, pp. 23-52.

[23] L. Carpinetti and A. De Melo, "What to Benchmark? A Systematic Approach and Cases.", *Benchmarking: An International Journal,* vol. 9, pp. 244-255, 2002.

[24] S. Rudolph, *et al.*, "Struktur von IT-Servicekatalogen - Ein praxisorientierter Gestaltungsvorschlag für die Dokumentation des IT-Leistungsangebots", in *Multikonferenz Wirtschaftsinformatik*, München, 2008, pp. 145-146.

[25] P. Ziaie, *et al.*, "Introducing software design elements for IT-benchmarking purposes", in *Next Generation Web Services Practices (NWeSP)*, Salamanca, Spain, 2011, pp. 404-409.

[26] H. Simon, *The sciences of the artificial (3rd ed.)*: The MIT Press, 1996.

[27] P. E. Meehl, "What social scientists don't understand", *Metatheory in social science,* pp. 315-338, 1986.

[28] S. Gregor, "The nature of theory in information systems", *Management Information Systems Quarterly,* vol. 30, p. 611, 2006.

[29] L. Dubé and G. Paré, "Rigor in information systems positivist case research - current practices, trends and recommendations", *MIS Quarterly,* vol. 27, pp. 597-635, 2003.

[30] T. Jick, "Mixing qualitative and quantitative methods - triangulation in action", *Administrative science quarterly,* vol. 24, pp. 602-611, 1979.

[31] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*. New York: Wiley, 1998.

[32] H. Al-Qaheri and M. K. Hasan, "An End-User Decision Support System for Portfolio Selection: A Goal Programming Approach with an Application to Kuwait Stock Exchange (KSE)", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 2, pp. 0-10, 2010.

[33] C.-H. Yang, *et al.*, "The Implementation of Wiki-based Knowledge Management Systems for Small Research Groups", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 1, pp. 68-75, 2009.

[34] O. Arazy and R. Kopak, "On the measurability of information quality", *Journal of the American*

Society for Information Science and Technology, 2011.

[35] R. Zanibbi, *et al.*, "Recognizing mathematical expressions using tree transformation", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 1455-1467, 2002.

[36] M. Eminagaoglu and S. Eren, "Implementation and Comparison of Machine Learning Classifiers for Information Security Risk Analysis of a Human Resources Department", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),* vol. 3, pp. 391-398, 2011.

[37] T. Böhmann, *Modularisierung von IT-Dienstleistungen - Eine Methode für das Service-Engineering*. Wiesbaden: Deutscher Universitätsverlag, 2004.

Author Biographies

Pujan Ziaie is a PhD candidate at the Technical University of Munich (TUM) in the field of information systems since 2009. He completed his Bachelor's in Computer Engineering at Sharif University of Technology in Iran, and his Master's in Computer Science (Artificial Intelligence and Robotics) at the TUM. He is also a member of the Center for Digital Technology and Management (CDTM) in Munich. His research interests include online production communities, collective intelligence, knowledge management, IT in developing countries, and IT benchmarking.

Markus Ziller earned his bachelor of science degree in information systems at the Technische Universität München (Germany) in 2011 and is currently, besides working for a big german insurance, doing his master's in information systems at the Technische Universität München (Germany). In addition, he is working as esearch assistant at fortiss GmbH.

Jan Wollersheim earned his bachelor of science degree in information systems at the Universität Stuttgart (Germany) in 2005 and completed his Master of science degree at the Technische Universität München (Germany) in 2007. After working for a consultancy he joined the research institute fortiss as research assiciate and started his PhD at the chair for information systems at the Technische Universität München (TUM).

Helmut Krcmar is Full Professor of Information Systems and holds the Chair for Information Systems at the Computer Science Department at Technische Universität München since 2002. He worked as a Postdoctoral Fellow at the IBM Los Angeles Scientific Center, as Assistant Professor of Information Systems at the Leonard Stern School of Business, New York University, and at Baruch College, City University of New York. The fifth edition of his book, Information Management, was published in 2009.