**SEVENTH FRAMEWORK PROGRAMME**

**THEME [ICT-2009.8.10]**

**Identifying new research topics, Assessing emerging global S&T trends in ICT for future FET Proactive initiatives**

# ITSy

# Deliverable D2.2.3: Recommendation Document

Project acronym: **ITSy**
Project full title: **IT[t] Simply Works**
Contract no.: 258058

| | | |
|---|---|---|
| | **ITSY – RECOMMENDATION DOCUMENT**<br>**Deliverable D2.2.3** |  |
| **ICT-2009-258058** | **Version: 2.8** | **28/10/2011** |

**Deliverable type:** Report

**Classification:** Pub.

**Work packages and tasks:** **WP2, Task 2.4, D2.2.3**

**Responsibility:** **Universität Potsdam**


**Executive summary:**

The present document covers the final recommendation based on the results achieved in the ITSy project.

It covers deliverable D2.2.3 of Task 2.4.

## Table of Contents

# 1. Introduction

The ITSy project is a European Union funded 7th Framework Programme that is focused on generating a set of innovative research topics on the concept of **simplicity as a driving paradigm in ICT development, maintenance and use**. The research we did included a literature review and a direct interaction with experts in the ICT area and beyond. The findings, especially in the focus groups and individual interviews, give evidence that the community of researchers, adopters, and users of ICT technologies and products believe that the philosophy of simplicity is strategically important, yet still insufficiently understood. This seems to be the case especially concerning the societal and technological consequences of its use, for example as a strategic directive in IT development and uptake, and thus it is rarely systematically applied.

Instead, design principles attempt to focus on increased functionality within thinly disguised complexity, often at the expense of life cycle costs and total cost of ownership issues (e.g., training, system malfunctions, system upgrades). Often designers are unaware of the tradeoffs and impacts.

Researchers in fields such as Computer Science center their work on efficiencies rather than on issues that make the application of their work easier to adapt, deploy and integrate into effective use.

With the increased use of ICT in socially critical application areas such as communications, healthcare, and administration, society can no longer afford systems that do not perform as specified, either functionally, or because they are too difficult to use or to maintain and integrate in larger federations or networks.

A deeper understanding of simplicity can be the key to a new generation of modeling paradigms, languages for design and development, and tools for the management of the artifacts and for the better communication of knowledge and intent.

We found evidence across all the communities that simplicity is a foundational issue, and that it informs and enables many desired characteristics of ICT systems such as reliability, usability and trust. The most advanced thinkers and doers in this direction were the researchers and chief developers in companies in the US West (e.g. Silicon Valley and Redmond) who have already taken action to implement several simplicity-driven changes in their products or in their way of working to define and then implement easy-to-deal-with technologies and products.

Thus, we believe that systematic knowledge gained through research on simplicity can provide the EU with a sustainable competitive advantage. To gain this knowledge we must develop the proper research directions and create an attractive environment that constructively fosters the creation, adoption and wider spread of simplicity-driven cultural traits and ultimately artifacts.

Our investigation comprised

- a review of IT literature, looking for articles that address the concept of simplicity
- discussion groups, focus groups and individual interviews with researchers and practitioners to identify current thinking of simplicity and how they may have applied it in their research and professional activity
- an online survey that we sent to specific communities of experts

Through these investigations we identified:

- evidence of past research in hardware and software where the concept of simplicity has been consciously applied or at least collaterally considered as an informing leading trait and
- current IT research that employs or targets simplicity in some manner.

Ultimately this effort is an attempt to discuss and to possibly understand what it would mean for IT research to shift from a focus on efficiency to a primary one of simplicity.

## 1.1. Structure of the Report

The current report describes briefly the methodology we followed (Section 2), summarizes our findings (Sections 3 and 4) and then presents our preliminary conclusions (Section 5).

A short chapter (Section 6) contains some recommendations for possible lines of action at the European level that emerged from the consolidation of the suggestions and in-depth discussions carried out during our interviews.

# 2. Methodology

We briefly describe the methodology we followed in conducting the literature review and the interviews, focus groups, and surveys that were conducted with experts in Computer Science, IT, Health Care, and other areas of relevance.

## 2.1. Literature Review

The identification and study of research articles in Computer Science and other key fields in the course of the ITSy project had two main objectives:

- Review relevant literature that deals with the essence of simplicity as a philosophy or as an essential paradigm (both for development and use of IT and for IT problem solving in a generic sense). This was done through systematic review of available bibliographic resources.
- Identify gaps regarding the notion of simplicity as a leading paradigm in current ICT research. This was tackled by specific readings on simplicity, largely of books and scholarly articles, as well as by following pointers to specific literature suggested by the experts during and after the interviews.

### 2.1.1. Systematic Review

Systematic literature reviews were applied in order to identify, evaluate and interpret available research relevant to the concept and application of simplicity. In the course of the ITSy project a systematic literature review has been considered to be an adequate method to reveal a potential research gap regarding the notion of simplicity in the area of computer science. Hence, a search strategy has been developed that aims at incorporating any relevant publication in a fair manner. At the same time we attributed great importance to preserve the transparency of the review process in order to allow for traceability and the assessment of the completeness of the search.

The approach of the review process has been aligned to the general requirements for systematic reviews in the context of computer science research. In particular, following the procedures described in [1], we performed the review in the following stages:

**Planning the Review**

1. Development of a review protocol that specifies the research question being addressed and the methods that will be used to perform the review
2. Definition of a search strategy that aims at detecting as much of the relevant literature as possible as well as its documentation so that readers can access its rigour and completeness
3. Specification of explicit inclusion and exclusion criteria to assess each potential primary study

**Conducting the Review**

1. Identification of research
2. Selection of primary studies
3. Study quality assessment

4. Data extraction and monitoring
5. Data synthesis
6. Reporting the review

Although these steps appear to be sequential, some of them emerged to involve iteration for reasons such as due to alignment of data with quality criteria.

In the course of the literature review we identified and analyzed philosophical, research, and practitioner articles on the concept of simplicity in various domains including business, computer science, human computer interaction, and healthcare among others. Articles on complexity and complex systems and the relationship of these concepts to simplicity were also included.

The review was also conducted in order to identify a set of exemplars that illustrate the implementation of simplicity in various research arenas. We found the latter most important in terms of developing a language of simplicity with various functional experts we interacted with. Through our initial interviews, we found that they often focused on user interface issues, and we wished to provide alternative thematic “anchors” from which to develop a broader perspective in the interviews and focus groups.

### 2.1.2. General Review

We have found that the concept of simplicity is, in itself, by far not a simple concept. There are many perspectives on the concept of simplicity and many proposed metrics suggested for measuring simplicity. For example, simplicity can be related to size as in the number of components a system possesses. It can also reflect the amount of effort a user of the system has to expend to use the system or the level of effort and amount of knowledge to understand the system. We did not find an exhaustive framework that captured and elucidated all the various conceptualizations of simplicity nor any agreed upon definition.

In our general review we also looked at the literature on complexity. In doing so we have found that the literature on complexity and complex systems is equally difficult. We found that there are many relevant dimensions that interplay with these two concepts, but importantly if seen as cultures, and in particular as research and innovation cultures, these two worlds and dimensions have different objectives and different characteristics, and cannot be just simply seen as being one the opposite or the inverse of the other. For example, simplicity may be seen as being manifested through systemic thinking and if the culture of Computer Science is one of layering and decomposition then difficulties in the interfaces between discordant layers will remain a key constraining factor in any attempt to arrive at truly innovative designs with a mandate of simplicity.

Through our work we find that simplicity, though not easily defined, is an important concept in many domains and is at odds with many design issues. For example, researchers such as Dr. Donald Norman, an expert on Human Computer Interaction, suggest that in a user’s selection of information-based systems, the concept of simplicity is often overridden by the need for features. Yet, it is the vast selection of features and how they are designed and implemented that lead to unusable systems. Other industry pundits call for computer scientists to move away from the focus on efficiency and embrace simplicity as an area of research. Thus, in a preliminary phase we used these observations to flesh out these understandings to inform our conversations with our experts. We continued to refine this understanding throughout the project so as to effectively include additional insights as our work progressed.

## 2.2. Expert Interviews

The following sections comprise the methodology we followed in conducting expert interviews as well as discussions with focus groups.

### 2.2.1. Analysis of Interview/Survey methods

We examined a variety of methods and techniques with which to elicit expert knowledge from the different communities. Our aim was to identify a proper mix of qualitative and quantitative concepts that adequately reconcile the width of the possible aspects and communities under investigation with the need of finding **relevant common ground** and **relevant criteria** for potential clustering/segmentation. In addition to developing our own questions, we explored other survey efforts on assessing simplicity/complexity that exist in the research community in order to fully inform our survey and to provide a foundation as solid as possible for our work in understanding this complicated topic.

From the survey results, our goal has been to draw significant conclusions from the responses provided by the experts participating in our study and to complement them with our own findings to inform the development of a research community on simplicity. Much of this work has been dovetailed with our efforts in performing some preliminary interviews. This effort is discussed in the next section.

### 2.2.2. Conducting preliminary Interviews and Focus Meetings

During the first three months of our project we conducted interviews with leading experts in Computer Science and Software Engineering who have a focus on research, and with experts in the IT industry who have a focus on the use of IT in business, government and industry. The preliminary aspects of these interviews was such that it served as a means to gather initial information and insights to better inform our work on subsequent information gathering techniques such as written surveys, interviews and other methods. It also helped to inform the efficacy and usefulness of the questions we were posing to the interviewed experts.

For example, preliminary interviews and focus meetings were conducted at the European Joint Conferences on Theory and Practice of Software (ETAPS) which is the primary European forum for academic and industrial researchers working on topics relating to Software Science. Interviews have also been conducted in several other research-oriented and technology-focused conferences to test the approach and response of experts from different sub-communities to our initial batch of simplicity-related questions and issues. The material we used at ETAPS is reported in Appendix B.

We recorded the conversations with the research and practitioner experts[1] so that we would be able to re-examine and analyze the discussions, the topics touched as well as the depth in which they had been discussed.

We observed from our findings so far that there is a strong difference in the two sets of preliminary conversations. We were finding that the rather practice-oriented conversation partners seemed to be more at ease with our questions and seemed to have a more direct relationship with the topic. This was expressed in terms of examples they gave, both positive and negative, of naming technologies that are simple or promise simplicity and why, and also in terms of mentioning other

[1] We have been authorized by the participants to record the conversations only for the purpose of our own later evaluation, not for being made public.

sources to consult, both literature/online sites and other experts that should be interviewed. They also offered more comments and suggestions on revisions to draft chapters of the report. On the contrary, experts active in more theoretical areas of Computer Science were more prone to identify abstraction as the central issue in simplification, and to suggest mathematical clarity as the source of improvement over the status quo. Thus, it was important for us to find examples of simplicity-related issues and research to help better frame the problem for these high level thinkers.

Both groups agreed that the current situation is very unsatisfactory for the end users (including themselves) and for IT professionals that design IT technology and IT products. Thus, there was broad agreement that a culture of simplicity would be highly beneficial to large areas of industry and the economy and also to the society at large. We note that many companies have adopted simplicity as a mainstay in creating a competitive advantage in the marketplace.

From these early experiences we gained useful information on how to develop the methodology for our interviews and summaries.

### 2.2.3. Multidisciplinary / Interdisciplinary Panel Meetings and Discussions with Focus Groups

Based on the findings just described, we decided to proceed in a manner that would allow us to take into account the larger diversity of focus, understandings, and of professional culture that we found in the various expert communities who participated in our initial interviews.

We decided that our surveys should be shorter with some questions asked in a more personal manner addressing the experts' specific research interests, activities and - hopefully - insights. This perspective was important so that we could give our experts a better chance to provide detailed and personally grounded perspectives to the concept of simplicity.

We also decided to proceed in a *modified Delphi method* fashion that combines the contributions the experts provided in live meetings, interviews, focus groups and discussions with a second phase of consolidation and discussion online on the ITSy website.

The use of the online survey website provided opportunities for asynchronous comments and suggestions. As we had seen in preparing the *Master Plan ICT* of the Federal State of Brandenburg, this feature can be extremely helpful in enriching the pool of open questions, problems, initiatives and groups that could be further investigated during the preparation of the final report.

To facilitate our work, we transcribed the recordings of the interview sessions to create an easily accessible document. These documents established a basis of evidence on which to develop a complete and structured review as well as the categorization and evaluation of the comments made during our interview/focus group sessions. The recordings of these sessions provided a means to capture comments made without the potential for loss if only note taking would have been employed.

With these documents in hand, we set up a catalogue of categories that allowed us to classify propositions extracted from the interviews with the multidisciplinary panels of experts. This approach allowed us to report a condensed and clearly arranged representation of the key propositions and the important issues to be addressed. In this analysis, we created a condensed set of main propositions to which we applied a very basic weighting based on recurrence and variance of particular types of statements throughout various interview sessions and conversations. We found that this was sufficient information to allow us to identify clusters of common understanding towards the topic of simplicity in IT and to isolate very subjective point of views reflected.

### 2.2.4. Online Survey

In using a variety of different research methods, we were able to involve a lot of experts from different communities. Besides the observation and the content analysis (literature review) as indirect methods, surveys as a direct method are used to collect data. The provided survey is a scientific method with the intent of a systematic determination of issues, by activating individuals to answer a collection of questions[2]. Surveys are often used in attitude and opinion research and in personality psychology. Three specific methods of surveys can be distinguished: the personal face-to-face interview, the telephone interview and the written interview. An online-survey is a special type of a written interview that is very economic in time and money[3]. This way a wide variety of experts can be reached and there is no need for a separate data entry. Influences from the interviewer (Social-Desirability-Response-Set) are systematically avoided.

The study has been conducted as a cross-sectional questionnaire with open, unstandardized questions to get an idea of what the experts think of what simplicity can be. To create and define these questions we used the literature reviewed so far as well as the input from the initial focus groups and interviews. We examined various models of simplicity proposed by others and combined them in order to achieve a general overview of these models.

The following five questions have been used in the survey:

Question 1: What research have you done or participated in where the outcome of the research made IT simple in some manner, or where IT played a particularly simplifying or complicating role?

Question 2: What aspect of IT design/development do you believe is the most complicated?

Question 3: What technologies do you use on a regular basis that you believe could be redesigned to be simpler?

Question 4: Can you think of a "principle" that you believe underlies simplicity? Can you identify an example of the use of this principle?

Question 5: How would you define simplicity?

The selection process of the sample can be separated in random or purposive sample.[4] Only results from random samples can be generalized. Considering the very specific content of the survey, we chose the purposive sample. Furthermore it has been an explorative survey and we did not need to generalize the results. Like in written interviews, the design of a survey influences the participation and responsiveness[5]. The text of introduction should be short to encourage people to participate. To brief the experts we decided to contact them directly with an individualized e-mail, containing a short introduction and the link to the online-survey[6]. Furthermore this e-mail was supplemented with a two-page document discussing the nature of the project and its rationale.

[2] c.f. [2] p. 54

[3] c.f. [3] p. 465

[4] c.f. [4] p. 62

[5] c.f. [5] p. 423

[6] http://www.cs.uni-potsdam.de/sse/ ITSy/survey.php

The structure beyond the 5 questions resulted from the intent to start with the individual expertise and experience of the respective participant and proceed from the personal research/activity level (instances - Question 1) to the level of needs in the profession (tactical - Questions 2 and 3) to a more abstract level (strategy - Questions 4 and 5).

In initial focus groups we had started with the opposite ordering and thus with a top down approach to the subject (from question 5 to 1). This resulted in a dissipation of energy and time on the more abstract level, which tended to keep the discussion stuck within that level, making it difficult to proceed towards more practical examples and discussions. Hence, we decided to choose the bottom-up order of questions, in particular for the online survey, in which we do not have the chance of interactive correction of track because it is self-introduced through the e-mail and accompanying document.

# 3. Literature Review aiming at Models of Simplicity

In our review of the literature, we examined various models of simplicity proposed by others. This section of our review presents a brief overview of four of these models and discusses how these models influenced our work in a general sense. We build on this understanding later in this document.

- The first model presented in subsection 3.1, *Simplicity Wins* reflects work done in an industrial setting and provides insight into the very real economic payoff that can be attained through a focus on simplicity. We envision such payoffs as being possible in the realm of IT. The actual simplicity parameters will be different, but we believe that the potential for an enormous payoff is at hand.
- The second model presented in subsection 3.2 by John Maeda is very abstract and presents some very interesting, high-level dimensions of simplicity. We used some of the concepts in this model in creating our questions and in facilitating our discussions with IT experts. It provides excellent insight into how to frame the many dimensions of simplicity.
- The third model presented in subsection 3.3 is more focused in the notion of complexity as it relates to size and the difficulty in understanding systems with vast numbers of components. The discussion focuses on some fundamental thinking by key researchers in biological and other organic systems. In particular it points to the issue synthesis and emergent behaviour. Such problems exist in such large disparate systems. Many of our discussions with IT experts focused on the notion of decomposition and layering and then combining subsystems together and the difficulties involved in doing so and in predicting / understanding system behaviour.
- The model by Don Norman presented in subsection 3.4 is important in its presentation and refutation of simplicity as the key driver of interface design. It challenges us to think clearly about the ideas of simplicity and how the multifaceted concept can be addressed in IT design.
- The last model presented in subsection 3.5 was brought forward by one of the founding fathers of Software Science. Edsger Dijkstra saw the structural backbone of programming in mathematics, and proposed aesthetic criteria of mathematical beauty and elegance to measure the “goodness” of a solution, an algorithm or a system. While software in industry is often not realized the way he wished, large part of academic research in the communities of

programming languages and techniques, algorithms and data structures is implicitly imprinted by his school of thought. However,

*In theory there is no difference between theory and practice. In practice there is.*

J.A.L. van de Snepscheut

## 3.1. Simplicity Wins

In their book, "*Simplicity Wins: How Germany's Mid-Sized Industrial Companies Succeed"*, authors Rommel, Kluge, Kempis, Diederichs and Bruck discuss their understandings of why certain Germany manufacturing companies have outperformed others in the market place. They place most of the success on notion of simplicity. For them, simplicity is measured in how manufacturing companies organize their competitive strategy and, thus, how they align key managerial decisions to co-inside with this strategy. Managerial decisions were made on issues such as product offering, number and variety of customers targeted and served, organizational system and the types of business systems. For example, the more successful companies focused their marketing efforts on a smaller subset of suppliers and customers and worked to optimize their interaction with those customers. Internally, the more successful organizations focused efforts on simple goals that were in direct alignment with understanding how each business operation provided value to their customers. Table 3.1 lists key simplicity objectives for these successful organizations:

| **Management Decisions** | **Simplicity designs** |
|---|---|
| Product range and customer structure | Concentrate on volume segments and core products offering optimal customer value. |
| Vertical Integration | Capitalize on areas where the company is strong by expanding vertical integration; otherwise, when outsourcing, reduce the number of suppliers and integrate them better. |
| Development | Increase downstream development efficiency by reducing risk upstream, and innovate in small, rapid steps |
| Location Structure and Logistics | Configure locations around products – with dedicated plants or "plants within plants" - and optimize materials flow |
| Technology | Simplify engineering before automating |
| Organization | Ensure transparent, simplified, decentralized structures, and create entrepreneurial spirit. |

Table 3.1: Designing for simplicity in a German manufacturing environment

While there are specific design suggestions based on simplicity that lead to a manufacturer's success, there are some abstract principles that can be derived such as 'reduction' and 'focus'. Such principles resonated with the IT experts that participated in our work. Such principles are also suggested by researchers in other domains as presented in the following paragraphs. It is useful to note that contribution illustrated by the study presented in this book on manufacturing shows that simplicity is a value per se when looking for focused expertise, for a continued relation that builds on familiarity, and with a consistent attention to leanness of structures, products, and organizations. In a sense, it underlies the importance of "less is more" for specialized competences and focussed ecosystems and, from our perspective, the research of, development of, and use of IT systems.

## 3.2. The Laws of Simplicity

In his book, *"The Laws of Simplicity"*, John Maeda of MIT, provides his insights into how his set of laws pertain to design, technology, business and life. The laws are shown in Table 3.2.

In our work with our IT experts, we referred to these principles and asked if they had any experience and/or examples of IT work that informed or supported these simplicity laws. Often our IT experts presented issues such as reducing and focusing on the 'core'. They also mentioned quite often that simplicity is context specific and that some systems are simple because the user is so very knowledgeable. At the same time, they mentioned that some systems are so complex because it is so very difficult to learn the system. Oddly, we rarely received responses from our IT experts on issues such as emotion or savings in time. We anticipate evaluating further the types of comments offered by our IT experts.

| **Ten Laws** | |
|---|---|
| Reduce | The simplest way to achieve simplicity is through thoughtful reduction. |
| Organize | Organization makes a system of many appear fewer. |
| Time | Savings in time feel like simplicity |
| Learn | Knowledge makes everything simpler |
| Differences | Simplicity and complexity need each other |
| Context | What lies in the periphery of simplicity is definitely not peripheral |
| Emotion | More emotions are better than less |
| Trust | In simplicity we trust |
| Failure | Some things can never be made simple |
| The One | Simplicity is about subtracting the obvious, and adding the meaningful |
| **Three Keys** | |
| Away | More appears like less by simply moving it far, far away |
| Open | Openness simplifies complexity |
| Power | Use less, gain more |

Table 3.2

The laws presented in this book are very insightful, especially given thought and consideration in their application. While the author provides many examples, there is a wealth from which one can draw on from these insights. At the same time, there is more that can be done and the actual application of such principles is far from an easy task. It should be noted that the first law in this book, *Reduce* - is consistent with Albert Einstein's famous quote

*"Everything should be made as simple as possible, but not simpler".*

## 3.3. Simply Complexity

In his book "*Simply Complexity: A clear guide to complexity theory*" Neil Johnson explains the essence of complexity theory. A chapter of the book that resonated significantly with our work was the chapter on mob mentality, Chapter 4. One central idea from this chapter is that 'systems' seem to adjust from one state to another state without any overall controlling mechanism (no 'invisible hand'). The author suggests that the key is 'feedback'. He also suggests that while 'individuals' in a group may behave with their own individual characteristics, when put together with other individuals, the group as a whole may have certain 'stable' characteristics that describe the group. "… emergent phenomena have universal properties." One quest this study was to understand how individual system characteristics can impact the overall emergent phenomena of information and communication systems. We propose that simplicity will, in mass, change the nature of IT based systems, making them more usable as is well known, but also, importantly, more robust and reliable. While this seems evident in hindsight, it remains an important concept to be explored more thoroughly as we begin to do in this project.. We also believe that the notion of feedback is an important aspect of simplicity. Through feedback, better decisions can be made. In particular, we have scarce resources and so are always making decisions that determine the allocation of those scare resources.. If there was an abundance of everything, then one could make poor decisions and it would not matter, but given there are scarce resources, one must decide and to do so wisely, one needs feedback. Thus, we suggest that feedback is one of the key aspects of system design that would enhance system simplicity.

## 3.4. Living with Complexity

In his book *"Living with Complexity"*, Don Norman argues against simplicity as a key design criteria. He suggests that people wish features and that simple systems are at odds with these wishes. Instead, Dr. Norman suggests that designers should adopt ways to 'live with complexity'. Thus, he presents various rules for managing complexity. These 'rules' became salient in our discussions with IT experts as ways of actually providing simplicity for users or for understanding the determinants of simplicity (e.g., knowledgeable users). For example, Dr. Norman discusses training and the importance of the end –user being knowledgeable in the problem domain. In fact, Dr. Norman advises end-users to invest time learning about the system and to not put all the burden on the designer.

Some of the rules that a designer should follow expressed by Dr. Norman include 'Making things understandable'. To do so, he advises using conceptual models and that the designer should clearly communicate conceptual models to the end user. Some techniques for doing so include using signifiers, providing clear organizational structure, providing automation where useful, modularization of the system, learning tools (e.g. manuals and help systems).

## 3.5. Beauty and Elegance

In his EWD896 Note[7] Turing Award receiver Edsger Dijkstra wrote in 1984

> *"As computing scientists we know that in our area, perhaps more than everywhere else, mathematical elegance is not a dispensable luxury but decides between success and failure. […]*
>
> *It is nice to know the dictionary definition for the adjective "elegant" in the meaning 'simple and surprisingly effective'. […]*
>
> *Also our academic reward system works against us. One can get credit for some complicated concepts one has introduced, it is hard to get credit for the discovery how some established but complicated concepts had better be avoided: those unaware of these concepts won't notice your discovery and those with vested interests in them will hate you for it. Hence my urgent advice to all of you to reject the morals of the bestseller society and to find, to start with, your reward in your own fun. This is quite feasible, for the challenge of simplification is so fascinating that, if we do our job properly, we shall have the greatest fun in the world."*

The *Oxford Advanced Learner's Dictionary* reports today "attractive and designed well" as a definition of elegance. Elegance and simplicity go hand in hand: the beauty of a theory is its structure, the aesthetics of a programming language is in the essence of its constructs, each of which should be either a primitive, i.e. a basic and non-derivable concept that co-defines the expressive power of the language, or a concept derived or composed from other concepts. It should then come with a clear construction rule specified within the language itself that reduces it to a legal combination of primitive concepts. Such languages are easy to compile, easy to understand, and easy to teach. Violating these principles of clarity of explicit definition brings with it high costs, in particular, for the translation to other languages and for comprehension.

Dijkstra created a school of thought that lasts to this day. The foreword of the book his colleagues and disciples edited for him in honor of his 60th birthday [10] concludes with the following statements:

> *"While many know of Dijkstra's technical contributions, they may not be aware of his ultimate goal, the mastery of complexity in mathematics and computing science, and of his belief that beauty and elegance are essential to this mastery. Even in the early EWD32 he wrote "... the greatest virtues a program can show [are] Elegance and Beauty". And later, in 1978, in conveying some beautiful arguments using mathematical induction in EWD697, he wrote the following:*
>
> > *When we recognize the battle against chaos, mess, and unmastered complexity as one of computing science's major callings, we must admit that* ***'Beauty is our Business',***
>
> *which gave us the title for this book."*

[7] http://www.cs.utexas.edu/users/EWD/transcriptions/EWD08xx/EWD896.html

## 3.6. Summary

These researchers have developed a variety of structured ways of looking at and characterizing simplicity concepts. We incorporated this thinking into our work and use it to reflect upon in analysing and characterizing the responses of our IT experts.

# 4. Systematic Literature Review

The following sections comprise the report of the systematic literature review. According to the general approach described in [6] the report covers a summary of the planning stage, the actual conduction as well as the evaluation process and the results achieved. Hence the respective sections are arranged by research question and furthermore subdivided into search strategy and the actual results of the review.

## 4.1. Research Questions

In the following the research questions to be answered by means of a systematic literature review are listed.

- Q1. How many publications in the area of computer science do explicitly discuss simplicity related topics? Is there a significant difference to the number of publications discussing complexity related topics?
- Q2. How many publications in the area of computer science do explicitly discuss the impact of simplicity as a driving paradigm in ICT development, maintenance or use?
- Q3. Can one identify specific sub-areas of computer science research that have obtained awareness for simplicity as an instrument for solving well-defined problems within an isolated context?

## 4.2. Question Q1

The following sections comprise the strategy and the results of the review stage aiming at answering

- Q1. How many publications in the area of computer science do explicitly discuss simplicity related topics? Is there a significant difference to the number of publications discussing complexity related topics?

### 4.2.1. Strategy

As finding an answer to question Q1 requires pure quantitative analysis, we decided to search for appropriate search terms in reasonable search fields of adequate literature databases. In order to identify this set of prerequisites we generated the following search strategy.

**<u>Search Terms</u>**

We have drawn up the following list of synonyms and antonyms of “simplicity”, whereas the antonyms are considered synonyms of “complexity”. In order to additionally cover adjectives in the course of the quantitative analysis we have drawn up a list of synonyms and antonyms of “simple” as well[8].

[8] The lists of synonyms and antonyms has been generated by adopting the most reasonable terms proposed by *Thesaurus.com*

**Synonyms of "simplicity"**

- simplicity
- simplification
- clearness
- easiness
- ease
- obviousness
- ingenuousness
- primitiveness
- purity
- severity
- straightforwardness
- elementariness
- simpleness
- chasteness
- restraint
- clarity

**Antonyms of "simplicity"**

- complexity
- complication
- difficulty
- difficultness
- complexness

**Synonyms of "simple"**

- simple
- simply
- simplex
- simplistic
- simplified
- easy
- effortless
- comfortable
- uncomplicated
- unproblematic

**Antonyms of "simple"**

- complex
- complicated
- difficult
- compound
- composite

## Search Fields

For the purpose of a pure quantitative analysis we decided to conduct separate searches in each of the most reasonable search fields and compare the results. Hence we conducted three separate searches for each of the search terms listed above, spanning the article title, the abstract of each article as well as the keywords associated with it. The underlying assumption thereby is that if an article explicitly discusses any kind of overarching simplicity concept its authors would have mentioned a corresponding keyword within any of these three search fields.

The results of the quantitative analysis following this strategy proved helpful for generating the further search strategy aiming at identifying a set of potentially relevant publications to be inspected in more detail in order to answer research questions Q2 and Q3

## Search Universe

For the systematic review we have chosen the option of using bibliographic databases because the alternative, i.e. searching each publication directly, would entail a much bigger effort of integrating the search results. We had a choice between the following digital bibliographic libraries.

- *Inspec*™ (http://www.theiet.org/publishing/inspec/)
- *Scopus*® (http://www.scopus.com)
- *IEEEXplore*® (http://ieeexplore.ieee.org/)
- *ACM Digital Library* (http://dl.acm.org/)
- *SpringerLink*® (http://www.springerlink.com/)
- *ISI Web of Knowledge*SM (http://www.webofknowledge.com/)
- *ScienceDirect*® (http://www.sciencedirect.com/)

A study to select an appropriate bibliographic database for systematic reviews in the area of computer science has been presented in [6]. This study investigated every database listed above except for *Inspec™*. The authors found that there were significant differences among databases that could affect search properties. While on the one hand the databases *IEEEXplore*®, *ACM Digital Library* and especially *SpringerLink*® as well as *ScienceDirect*® cover only a small group of journals, on the other hand especially *IEEEXplore*® as well as *ACM Digital Library* has the limitation to not provide an abstract for every article. We were able to confirm this appraisal by our own investigation. Hence we considered these bibliographic resources too limited to match our requirements.

Furthermore, we decided to disregard *ISI Web of Knowledge*SM because it misses the ability to search for terms in the abstract or keywords of an article, which we considered an essential feature for our search strategy.

Finally, we decided to shortlist *Inspec™* and *Scopus*® as our candidates to conduct the first stage of the structured review. Apart from including *Inspec™* this decision is congruent with the selection taken place in [6] where the authors selected *Scopus™* because they rate it to have fewer weaknesses than the other databases listed. *Scopus*® provides a wide coverage of publications especially in the field of computer science (c.f. *Appendix E*), it maintains a consistent database and allows the exportation of search results. However, our initial investigation did not reveal any disadvantages of *Inspec™* in comparison with *Scopus*®*.* Hence we decided to use both libraries for the first review stage. Besides answering question Q1 this strategy would allow a comparison of the quantitative analysis of the search results between two different databases having a wide coverage of publications.

### 4.2.2. Results

The table in *Appendix F* provides the detailed numbers achieved by the quantitative analysis in the first stage of the review. It lists the number of articles found for each search term (synonyms and antonyms of simplicity) separated by the respective search field (title, keywords and abstract) as well as by the literature database that have been inquired (*Inspec™* and *Scopus*®). Additionally, we have added a column that lists the number of articles for that we found the respective search term in the article title as well as in the keywords, which allows for a better assessment of overlap in these two search fields.

The following notes comprise some peculiarities of these results.

- The number of articles whose search fields contain a synonym of complexity by far exceeds the number of those whose search fields contain synonyms of simplicity. This phenomenon can be observed best if comparing the results for the terms “simplicity” and “complexity” themselves.

| | INSPEC | | | | Scopus | | | |
|---|---|---|---|---|---|---|---|---|
| Search term | Article Title | Keywords | Title & Keywords | Abstract | Article Title | Keywords | Title & Keywords | Abstract |
| simplicity | 313 | 1122 | 104 | 15700 | 168 | 115 | 18 | 8480 |
| complexity | 15401 | 72638 | 12798 | 122785 | 10259 | 62276 | 6973 | 86245 |

Table 4.1

Although this view does not provide deep insights into the content of the respective articles it gives a first impression of the general weighting of simplicity related topics in the computer science literature.

- In general, the occurrence of a search term within the abstract of a paper by far exceeds the occurrence within the title or keywords. One of the reasons obviously is that in the descriptive text of the abstract the search term is used for the explanation of secondary aspects or in sentences that are not related to the main topic directly.

## 4.3. Question Q2

The following sections comprise the strategy and the results of the review stage aiming at answering

Q2. How many publications in the area of computer science do explicitly discuss the impact of simplicity as a driving paradigm in ICT development, maintenance or use?

### 4.3.1. Strategy

Stage 1 of the structured literature review followed the approach of finding as many articles as possible that discuss simplicity related topics within the scope of computer science. The second step builds on these results and applies deeper investigation of the content of each article found this way in order to answer question Q2. The underlying assumption is that if any article addresses simplicity as a driving paradigm in ICT development, maintenance or use it most likely is amongst these initial findings. As potentially relevant articles need to discuss the notion of simplicity as a driving paradigm in an immediate and primary sense in order to fall in the category of literature characterised by question Q2, we established our selection criteria accordingly. Hence we disregarded articles discussing a simplification approach that is tailored to a particular problem-solving technique within a limited context and instead rather looked out for fundamental approaches and overarching concepts supposed to be applied on a preferably large set of issues in ICT development, maintenance or use.

However, in order to identify those relevant articles we had to gain deeper insight into the content of each article that could not be achieved by a keyword search within literature databases. Instead, we had to go through each finding and have a closer look at the respective abstract as well as at the full text if information extracted from the abstract would not be sufficient. As this can only be done by pure manual work we decided to restrict the search universe for this second stage of the survey to a manageable subset of articles covering the most reasonable references.

In order to generate a set of the most reasonable articles to be investigated in more detail, we pursued the following two-step strategy.

1. After completion of the first review stage we took a closer look at a set of random samples from the results of both *Inspec™* and *Scopus®* databases and found that the results of *Inspec™* were too widespread although having limited the search to the field of information technology and related disciplines. Our investigation of the abstracts and full texts revealed that many articles within the search result discussed topics that could not or just barely be brought in any relation to information technology. On the other hand nearly all appropriate findings of the spot sample have also been found via the search in *Scopus®*. Hence we decided to stick with *Scopus®* for the further stages of the literature review, especially because stage 2 as well as the following review stages would require manual work in order to analyse each finding in more detail and we were interested in saving time by not going through the high number of bland articles received by the search in *Inspec®*.

2. We made the additional assumption that if any article addresses simplicity as a driving paradigm in ICT development, maintenance or use it most likely contains a hint in its title and the keywords associated with it. Although this decision was mainly driven by the need for a small enough set of literature to be sorted through by hand, our investigation of a set of random samples from those articles that mention any synonym of simplicity in only one search field (i.e. in the abstract, the article title or in the keywords) showed that in all of the investigated cases these terms were only used in the context of secondary aspects or placed incidentally. However, at this point we would like to clearly point out that due to this assumption relevant articles might have been rejected by mistake, although we expect their number to be very small.

The application of the two restrictions above led to a set of 1033 articles found by a keyword search for synonyms of simplicity in the article title and keywords within *Scopus®* database, which we considered a manageable number of articles to go through gradually. Just as before, the subject area has been restricted to computer science and multiple disciplines. To be even more precise, we used the following search string in the advanced search of *Scopus®*.

```
(   KEY("simplicity") OR TITLE("simplicity")
    OR ( KEY("simplification") AND TITLE("simplification") )
    OR ( KEY("clearness") AND TITLE("clearness") )
    OR ( KEY("easiness") AND TITLE("easiness") )
    OR ( KEY("ease") AND TITLE("ease") )
    OR ( KEY("obviousness") AND TITLE("obviousness") )
    OR ( KEY("ingenuousness") AND TITLE("ingenuousness") )
    OR ( KEY("primitiveness") AND TITLE("primitiveness") )
    OR ( KEY("purity") AND TITLE("purity") )
    OR ( KEY("severity") AND TITLE("severity") )
    OR ( KEY("straightforwardness") AND TITLE("straightforwardness") )
    OR ( KEY("elementariness") AND TITLE("elementariness") )
    OR ( KEY("simpleness") AND TITLE("simpleness") )
```

```
    OR ( KEY("chasteness") AND TITLE("chasteness") )
    OR ( KEY("restraint") AND TITLE("restraint") )
    OR ( KEY("clarity") AND TITLE("clarity") )
) AND ( LIMIT-TO(SUBJAREA,"COMP") OR LIMIT-TO(SUBJAREA,"MULT") )
```

Please note that we included all articles whose title *or* keywords contain the search term "simplicity" as on the one hand the number of additional articles found this way does not significantly impact the total number of articles to be investigated in more detail and on the other hand we ascribed this exact term an eminent role.

The set of 1033 articles obtained this way has been taken as the basis for this stage as well as for the following stage of the systematic literature review.

### 4.3.2. Results

The full list of articles obtained by the search in *Scopus®* is enclosed as *Appendix G / Appendix H* in order to provide a consistent basis for the reader to assess the coverage of articles that have been analysed in more detail.

The selection of articles that discuss the notion of simplicity in a narrower sense is listed beneath. These articles represent the result of deeper investigation of the 1033 articles obtained by the keyword search. However, only some of them have been considered worth a detailed discussion of their content by means of adopting embedded aspects in the sense of discussing them within the final report of this project.

The list of articles is very small compared to the number of articles found via the keyword search as they only make up about 3% of them. The list is even smaller compared to the huge number of articles in the field of computer science hold by the underlying literature database *Scopus®*. Hence an appropriate answer to question Q2 is that simplicity as a driving paradigm is rarely addressed by literature in the area of computer science. Overarching concepts or simplicity-focused aspects are only addressed sporadically. However, this fact confirms our previous appraisal that the potential of the notion of simplicity is widely unexplored.

*[Please note: For the sake of completeness, the listed articles contains the paper [Margaria2010] that has been generated and published in the course of this project. We have considered the fact that we found it via our systematic literature review a partial verification of our search strategy.]*

[Bonawitz2008] Bonawitz, E.B., C. I. C. C. L. T. (2008), 'Ockham's razor as inductive bias in preschooler's causal explanations', *2008 IEEE 7th International Conference on Development and Learning, ICDL* , cited By (since 1996) 0, 7-12.

[Buschmann2010] Buschmann, F. (2010), 'Learning from failure, part III: On Hammers and Nails, and falling in love with technology and design', *IEEE Software* **27**(2), 49-51.

[Calisir2004] Calisir, F., C. F. (2004), 'The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end-user satisfaction with enterprise resource planning (ERP) systems', *Computers in Human Behavior* **20**(4), 505-515.

[Chafin1980] Chafin, R. L. (1980), 'SIMPLICITY IN COMMAND AND CONTROL SYSTEMS - A HUMAN FACTORS CONSIDERATION.', *International Telemetering Conference (Proceedings)*

**16**, cited By (since 1996) 0, 519-525.

[Chang2007] Chang, A., G. J. Z. J. I. H. (2007), 'Simplicity in interaction design', *TEI'07: First International Conference on Tangible and Embedded Interaction* , cited By (since 1996) 4, 135-138.

[Clark2009] Clark, K. (2009), 'Engaging and Adaptive: Going beyond Ease of Use', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **5619 LNCS**, 46-54.

[Dalcher1999] Dalcher, D. (1999), 'Simplicity and power: when more means less', *Computer* **32**(5), 117.

[Davis2001] Davis, S., W. S. (2001), 'The mediating effects of intrinsic motivation, ease of use and usefulness perceptions on performance in first-time and subsequent computer users', *Interacting with Computers* **13**(5), 549-580.

[Edmonds2000] Edmonds, B. (2000), 'Complexity and scientific modelling', *Foundations of Science* **5**(3), 379-390.

[Fogg2009] Fogg, B. (2009), 'A behavior model for persuasive design', *ACM International Conference Proceeding Series* **350**, cited By (since 1996) 0.

[Goodman1958] Goodman, N. (1958), 'The test of simplicity', *Science* **128**(3331), 1064-1069.

[Hale2007] Hale, G. (2007), 'Hidden complexity makes for simplicity', *InTech* **54**(3), 7.

[Karahanna1999] Karahanna, E., S. D. (1999), 'The psychological origins of perceived usefulness and ease-of-use', *Information and Management* **35**(4), 237-250.

[Keil1995] Keil, M., B. P. K. B. (1995), 'Usefulness and ease of use: field study evidence regarding task considerations', *Decision Support Systems* **13**(1), 75-91.

[Kelly2004] Kelly, K. (2004), 'Justification as truth-finding efficiency: How Ockham's razor works', *Minds and Machines* **14**(4), 485-505.

[Kim2009] Kim, H., L. W. (2009), 'Designing unobtrusive interfaces with minimal presence', *Conference on Human Factors in Computing Systems - Proceedings* , cited By (since 1996) 0, 3673-3678.

[Kuo2007] Kuo, W. (2007), 'Compatibility and simplicity: The fundamentals of reliability', *IEEE Transactions on Reliability* **56**(4), 585-586.

[Lewis2007] Lewis, C. (2007), 'Simplicity in cognitive assistive technology: A framework and agenda for research', *Universal Access in the Information Society* **5**(4), 351-361.

[Maeda2004] Maeda, J. (2004), 'Simplicity', *BT Technology Journal* **22**(4), 285-286.

[Margaria2010] Margaria, T., S. B. (2010), 'Simplicity as a driver for agile innovation', *Computer* **43**(6), 90-92.

[Mathieson1998] Mathieson, K., K. M. (1998), 'Beyond the interface: Ease of use and task/technology fit', *Information and Management* **34**(4), 221-230.

[Meunier2008] Meunier, P. (2008), 'Software transparency and purity', *Communications of the ACM* **51**(2), 104.

[Miles2002] Miles, J., M. L. C. J. (2002), 'Matching computational strategies to task complexity and user requirements', *Advanced Engineering Informatics* **16**(1), 41-52.

[Munoz2010] Muñoz, P., G. P. P. V. (2010), 'Refining interaction designs through simplicity', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **6439 LNCS**, 31-40.

[Norman2008] Norman, D. (2008), 'Simplicity is not the answer', *Interactions* **15**(5), 45-46.

[Nowak2006] Nowak, T., C. M. S. R. G. L.-L. (2006), 'Simplicity pays', *ABB Review*(1), 55-58.

[Paola2011] Paola, C. (2011), 'In modelling, simplicity isn't simple', *Nature* **469**(7328), 38.

[Papanis2005] Papanis, J., K. S. K. E. D. N. P. G. V. I. (2005), 'Facilitating context-awareness through hardware personalization devices: The simplicity device', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **3744 LNCS**, 252-262.

[Picking2010] Picking, R., G.V.M.J.C.J.G.H. (2010), 'Simplicity, consistency, universality, flexibility and familiarity: The scuff principles for developing user interfaces for ambient computer systems', *International Journal of Ambient Computing and Intelligence* **2**(3), 40-49.

[Pothos2002] Pothos, E.M., C. N. (2002), 'A simplicity principle in unsupervised human categorization', *Cognitive Science* **26**(3), 303-343.

[Roberts2004] Roberts, E. (2004), 'The dream of a common language: The search for simplicity and stability in computer science education', *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)* **36**(1), cited By (since 1996) 1, 115-119.

[Sha2001] Sha, L. (2001), 'Using simplicity to control complexity', *IEEE Software* **18**(4), 20-28.

[Stark2000] Stark, J. (2000), 'Observing complexity, seeing simplicity', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **358**(1765), 41-61.

[Wirfs-Brock2007] Wirfs-Brock, R. (2007), 'Toward design simplicity', *IEEE Software* **24**(2), 9-11.

[Wolff1990] Wolff, J. (1990), 'Simplicity and power. Some unifying ideas in computing', *Computer Journal* **33**(6), 518-534.

## 4.4. Question Q3

The following sections comprise the strategy and the results of the review stage aiming at answering

Q3. Can one identify specific sub-areas of computer science research that have obtained awareness for simplicity as an instrument for solving well-defined problems within an isolated context?

### 4.4.1. Strategy

Finding an answer for question Q3 required the application of meta-analysis on the findings generated in stage 2. In particular, we applied analysis in the following separate steps.

1. Identification of the sub-area of computer science of each article found so far
2. Identification of the context of the notion of simplicity within each article found so far

Unfortunately, *Scopus®* does not provide a consistent classification of its articles. Hence we needed to identify the main sub-area of computer science by looking at every finding gradually. Thereby we used the following simplified taxonomy of areas of computer science that bases on the taxonomy proposed by the article about "Computer Science" at *Wikipedia.org*[9]. In turn, this article points to the *Computing Sciences Accreditation Board (CSAB)* as the source of the areas listed[10].

**<u>Categories of computer science used for the structured literature survey</u>**

- Theory of computation
  - Automata theory
  - Computability theory
  - Computational complexity theory
- Algorithms and data structures
  - Algorithms
  - Data structures
  - Computational geometry
- Mathematical foundations
- Concurrent, parallel and distributed systems
- Interdisciplinary application
- Computer architecture and engineering
  - Microarchitecture
  - Computer networks
  - Databases
  - Systems architecture
  - Programming languages
- Computer graphics and visualization

[9] http://en.wikipedia.org/wiki/Computer_science (retrieved June 2011)

[10] c.f. CSAB: "Computer Science as a Profession" (28 May 1997), http://web.archive.org/web/20080617030847/http://www.csab.org/comp_sci_profession.html

- Computer security and cryptography
- Scientific computing
  - Numerical analysis
  - Computational physics
  - Computational chemistry
  - Computational linguistics
  - Bioinformatics
  - Medical computing
- Artificial intelligence
  - Machine learning
  - Computer vision
  - Image processing
  - Pattern recognition
  - Cognitive science
  - Data mining
  - Evolutionary computation
  - Signal processing
  - Robotics
  - Natural language processing
- Information science
  - Information retrieval
  - Knowledge representation
  - Computer interaction
  - Human-computer interaction
- Software engineering and formal methods
  - Software engineering
  - Formal methods
- Scientometrics and Education
- Management and Governance

As soon as we had identified the main area of computer science of each of the literature findings found in stage 2, we applied the second step of the analysis, i.e. assessing the concept of simplicity as well as the context it is applied to in the respective article. In combination with the preceding categorization of the articles, following this approach should allow for identification of recurring simplification approaches within particular areas of computer science and hence help us to find out whether there exist distinguished domains that have developed some kind of awareness for simplicity as an instrument of solving a well-defined sub-set of problems. In turn, this is exactly the question Q3 to be answered in this stage of the review.

However, the identification of recurring semantic patterns was only possible by achieving a better understanding of the content of the articles, which consequentially required reading associated abstracts and full texts. As a structured documentation of this semantic analysis is merely impossible, we decided to at least document helpful hints on the context the respective notion of simplicity was used in by taking notes on the immediate context of the search term. It turned out that this approach would allow further insights, although merely shallow ones.

### 4.4.2. Results

Figure 4.1 shows a distribution diagram illustrating the number of findings by the main categories of computer science we have chosen for the classification of an articles' content.

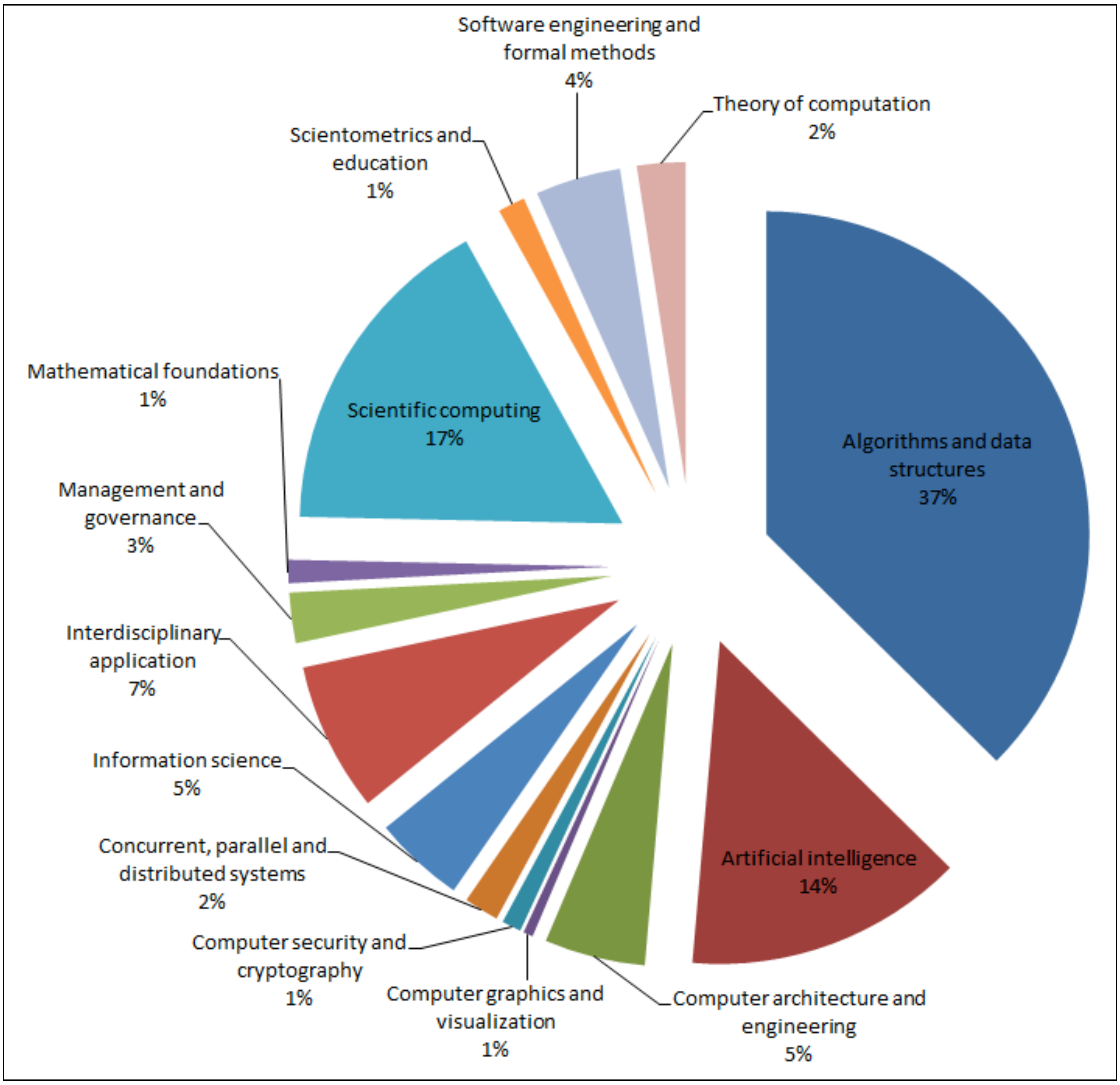


**Figure 4.1: Findings by category**

The following figures (Figure 4.2 to Figure 4.6) show details about the occurrence of each search term for the five categories representing the main portions in Figure 4.1. Interestingly they show that the distribution of the main occurrences of search terms differ significantly depending on the main area of computer science the article has been assigned to.

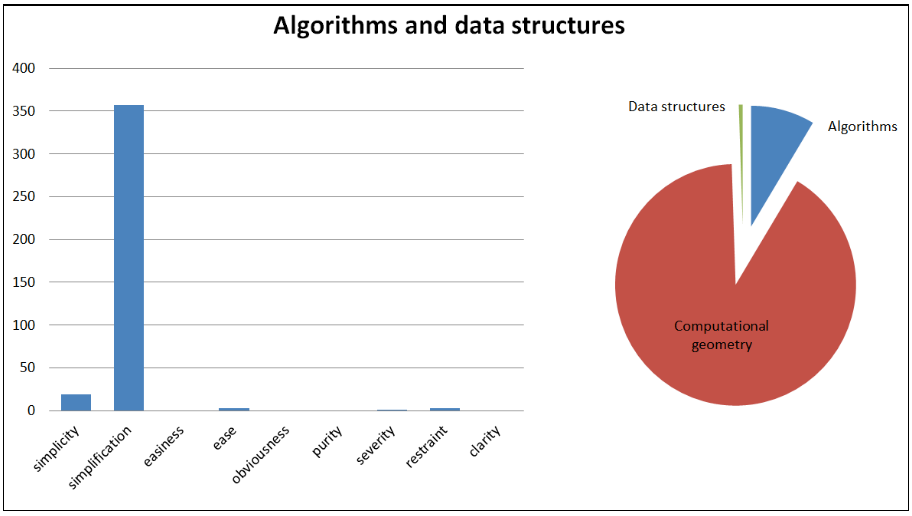


**Figure 4.2**

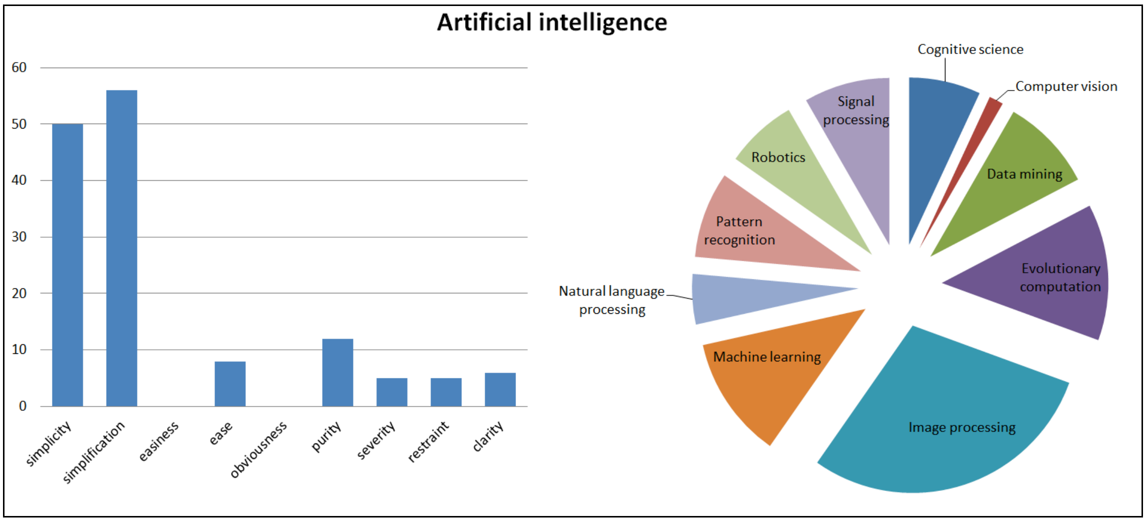


**Figure 4.3**

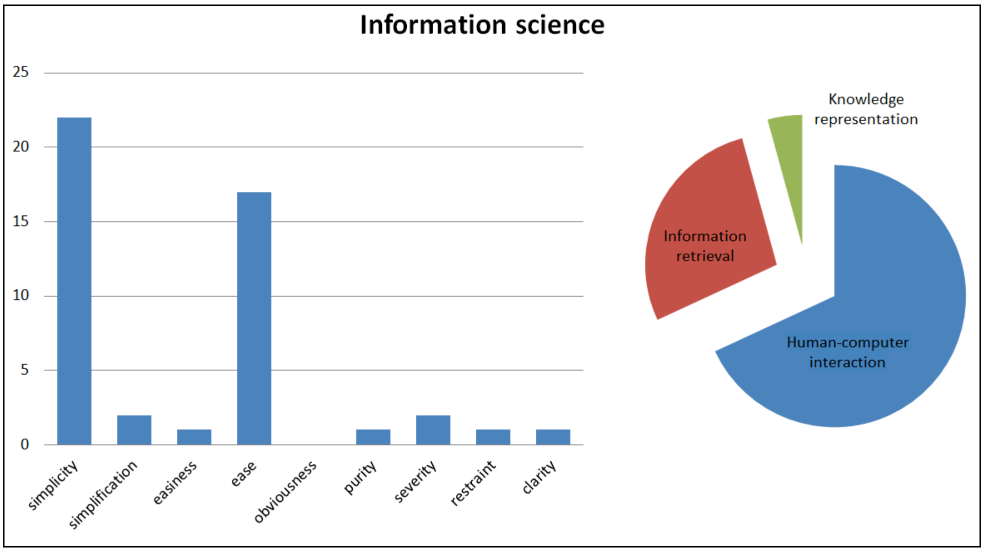


**Figure 4.4**

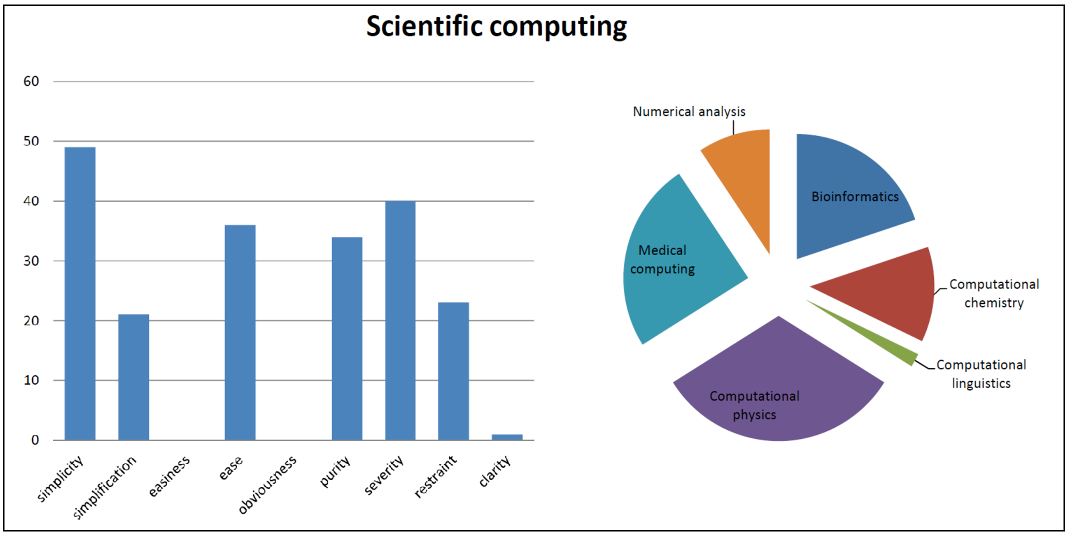


**Figure 4.5**

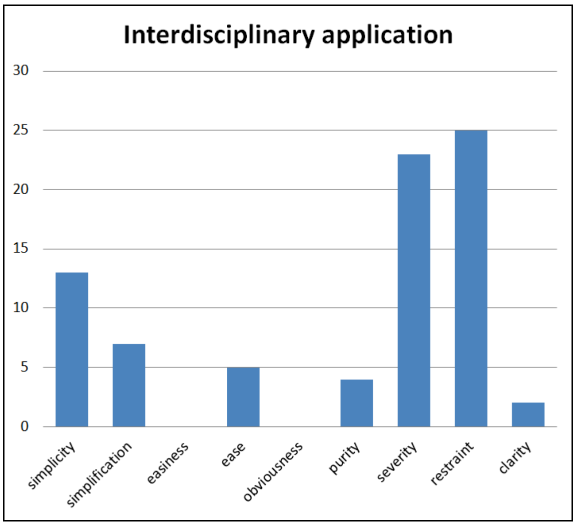


**Figure 4.6**

Observing the immediate contexts of search terms within article title and associated keywords we found that this fingerprint-like feature relies on the occurrence of recurring combinations of words that are typical for the respective area of computer science these articles are related to. For a further investigation of this observation we introduced the notion of wider contexts in order to group word combinations that are syntactically different but semantically similar under a unifying umbrella term. As an example, the most recurring wider context generating the massive number of findings for "*simplification*" in the sub-area of *Computational Geometry* (c.f. Figure 4.2) is "*mesh simplification*", whereby the latter is an umbrella term for similar phrases like "*simplification of meshes*", "*polygon simplification*", "*3D model simplification"* and so on. However, we have chosen the term "*mesh simplification"* as the umbrella term because the recurrence rate of the exact term itself is very high.

We were able to make similar observations within the findings in other areas of computer science. In particular, the high number of findings for the search term "*ease*" in the area of *Human-Computer Interaction* (c.f. Figure 4.4) could be explained by the high recurrence of "*ease of use*" whereas the significantly higher findings for "*severity*" and "*restraint*" in the area of *Interdisciplinary Application* (c.f. Figure 4.6) could invariably be explained by their use in domain-specific terms but never in a synonymous sense regarding simplicity. Instead, the immediate contexts they occur in were word combinations like "burn severity", "risk severity", "*belt restraint systems*" and the like. Similar observations have been made in articles within the area of *Scientific Computing*. For the sake of completeness, the full list of recurring contexts by category of computer science we have identified within article titles and keywords has been appended in *Appendix I*.

Leaving aside those word combinations that are not synonymous to "*simplicity*" we were able to identify a set of fixed terms in a synonymous sense that have been found significantly more often than others. Table 4.2 provides a summary of the most reasonable and recurring terms we found, listing both the findings of the exact term as well as findings in a wider context.

| | exact term (wider context) | | |
|---|---|---|---|
| **fixed term** | **# findings (title)** | **# findings (keywords)** | **# findings (title OR keys)** |
| conceptual simplicity | 1 | 25 | 26 |
| computational simplicity | 1 | 17 | 18 |
| data simplification | 8 (10) | 3 | 10 (11) |
| ease of use | 25 (26) | 19 | 30 (31) |
| function simplification | 1 (9) | 1 (6) | 2 (12) |
| graph simplification | 1 (3) | 1 (2) | 1 (3) |
| image simplification | 8 (10) | 9 | 10 (12) |
| mesh simplification | 93 (129) | 110 (133) | 127 (173) |
| model simplification | 38 (41) | 36 | 52 (55) |
| network simplification | 3 | 3 | 3 |
| program simplification | 4 (7) | 7 (8) | 8 (11) |
| rule simplification | 1 (3) | 0 (3) | 1 (4) |
| structural simplification | 5 (10) | 3 (5) | 10 (12) |
| system simplification | 2 (3) | 1 | 2 (3) |
| text simplification | 5 | 3 | 5 |

**Table 4.2**

# 5. Simplicity Topics

Our stated goals in this project were two-fold. First we proposed to develop a set of research topics on the philosophy of simplicity in its application to the development, maintenance and use of ICT and secondly to inform diverse research communities about the discovered research directions, acting as a catalyst for research initiatives and FET proposals. In this section we list and discuss the many topics of simplicity we identified. We use the term *topic* to organize various aspects of simplicity including characterizations of simplicity, dimensions of simplicity, metrics for measuring simplicity, principles of simplicity, and design aspects that lead to simplicity. These topics were extracted from our review of the IT literature, through discussions with IT and functional experts in our interviews and focus groups, and through feedback given through our online survey. These topics provide the essence of what our colleagues believe are the underpinnings of research needs which, if embraced, can form the basis of a multidisciplinary research community leading to a competitive advantage to the EU in the IT marketplace. The topics that we discuss provide an initial organizing framework but we recognize that other organizational schemes may be derived from our thinking after additional research has been conducted. We also acknowledge that other schemes have been presented by other researchers, but this presentation is one gleaned from the eyes of our IT experts and thus impacts the work that they do and plan to do.

In our discussions with IT and other experts, our beginning discussions revolved about the question of 'simple for whom?' The realization is that often when making some aspect of the system easier for one user; another aspect of the system becomes more complicated, performing more functions, storing more data, interacting with more components. Another aspect of our discussions concerned the inherent nature of the IT environment. IT is not inherently simple. Computer languages are difficult to learn and to use, determining requirements from users who are uncertain about their needs is not simple, and many other aspects of this world confront us in our quest for 'simplicity'. That said, our experts had some opinions about what could make their world better through a focus on simplicity. We discuss their thoughts and comments here.

## 5.1. The Art of Knowing

> *'... that is, when a tool does something, you know what it has done and why it has done that.'*

The essence of simplicity from the perspective of our participants is knowledge. The more users, both end users of IT as well as IT developers and practitioners, know about an IT system, the more 'simple' it appears to that user. This perspective is in alignment with the mission of most IT research, that is, the quest for knowledge. Through our discussions researchers lamented the lack of efficacy in how knowledge is shared in their community. HCI researchers state that providing a conceptual model (e.g., the work by Don Norman mentioned earlier regarding helping end users) to users is important for user understanding of the system. But our researchers expressed concerns about the multiple IT models that exist, the many standards that exist (sometimes conflicting) as well as the level of abstraction (often at the wrong level) IT researchers apply to their work and to convey the essence of their work to others. IT research and development is often done in silos with only specific communities fully understanding what has being accomplished. While overarching frameworks such as the enterprise information systems Zachman framework [7] exist for development of enterprise IS systems, such organizing frameworks are not prevalent in general and certainly not ones that lead towards a direction of simplicity. Instead most of our researchers stated that their research added to the complexity of IT rather than made IT simpler. Moreover our participants felt that they did not deliver on providing their findings in a manner that is accessible to others.

*' … this year I went to a meeting and there was this whole session on model-based testing and nobody could understand a word any of these speakers said, it was way too complicated. It is no wonder nobody buys these tools, it is because nobody - even the developers can't even explain to you what these things are doing, so how can anybody use it? And now I am really like completely disillusioned by model-based testing because in theory it is a good idea but in practice it seems that these tools are unusable.*'

Providing usable knowledge to others often means tapping into some conceptual framework that the user already possesses. Thus the notion of a 'desktop' in user interface design remains an illustrative example. That said, providing knowledge to users is not often considered a key task or is done in a manner that is not readily accessible to the users. Manuals though helpful are not often read, though some researchers state that it behoves users to be prepared to use their IT systems (ref is Norman). Some suggest that the best IT systems are ones where the user may explore the system to learn about its functionality. IT researchers and developers must address simplicity by effectively dealing with IT's inherent complexities, its actual complexities, and, importantly, the user's felt complexities. Thus how knowledge of the IT is acquired by the user remains a key quest for developing a system built on simplicity.

*' … a person knows how to make the system operate but has no idea about how it does so or the mechanisms behind it. Does he / she need to know? When? Under what circumstances … simplicity is a function of education …'*

Many of the statements below touch on the manner in which IT researchers and professionals have attempted to in still simplicity into their work and into IT in general. But the current statements suggest that an initiative that fostered better communication among researchers and IT professionals is needed.

*' … making things understandable with as little knowledge as possible about that particular thing.'*

As a final thought on knowledge, some of our participants believe that if one can reduce the amount of knowledge needed, to cut to the essence or core of the design or the methods used to develop systems, then simplicity can be better achieved.

*'In general in order to be able to use a programming language or framework you have to know so many tiny little details. You really have to be an expert before you can get the first working application. I find this very difficult so knowledge is important but this suggests that knowledge in many, many domains and at a very detailed level is important. Can the amount of that you need to know be reduced in some way?'*

*'The absence of ambiguity - if something is ambiguous it cannot be simple.'*

But, knowledge is not enough to gain simplicity; more is needed as stated by one of our participants.

> *No, it is not just a question of knowledge. ... It's about having notations, having solution methods that are powerful.*

## 5.2. Structure

Structure was a common theme running through our discussions and through research that we reviewed (e.g., John Maeda's comments about 'organization in his list of simplicity principles). Structure was important for a number of reasons both in terms of design and in terms of use. In human computer interaction, designers are called upon to structure tasks concisely to reduce cognitive load and to make each task independent and easily learnable and doable. Structure as a design tool in IT was presented as in terms of 'modular design', 'separation of concerns' and in 'layering'. An interesting quote on structure was the following:

> *'Another thing that I believe is very important is structure. I have the kind of memory that is very bad at collection of random facts. Some people can do that but I need structure to recall things. So for me structured things tend to be simpler than unstructured things'*

This means that the person has abstracted or generalized or created an internal model of the artefact under discussion and this becomes the measure of all instances and how they are interpreted and ultimately used (or thought of how to be used) by the consumer / user. For IT use, this perspective is often characterized as a conceptual model. One researcher described this notion in terms of the mathematical concept of a 'basis'.

In addition to system use, structure in IT design and development plays an enormously important role in how the system can be modified and maintained. We believe that a key aspect of increasing simplicity in IT is through allowing users to add features and functionalities as they desire them. We discussed this approach with key designers at a major IT firm and their response was that most architectures do not support such an approach, that one must live with the set of functions that have been provided or do extensive tailoring and customization. Yet, an extensive set of features and functionalities that are never used nor desired leads to a lack of simplicity, and, we believe higher total costs of ownership. Moreover, such extensive feature sets and monolithic structures become too difficult to maintain.

The notions of *layering* and *decomposition* were mentioned repeatedly in our focus group sessions as means of organizing problems so that focus could be directed to a smaller subset of the problem thus leading towards a simpler problem domain. Whether this lead to an overall reduction of complexity in the system was left open to question; many felt that making one part of the system under analysis lead to less simplicity in other parts of the system. What is needed are better guidelines towards selecting layering and decomposition boundaries. For example, one participant stated the following:

> *What is most complicated: The decomposition of systems, particularly how to handle crosscutting concerns in that decomposition. Because there are always concerns that*

*relate to the entire system, things like security quality, performance and you nevertheless have to handle it.*

In systems such as in the nuclear energy field, one specialist stated that decomposition is a critically important topic and that certain components of the system must be decomposed. For example:

*'And what they do in the nuclear energy field is that safety and control are separated. If you think about the most important part of the reactor, it is the shutdown system. So don't integrate it with the control system that maintains a particular power level, because you exponentially increase the size of the code and requirements and everything else you have to verify. And this is something that not everybody gets …'*

If designers in the nuclear energy field do not understand the essence of this argument, then it is possible that other critically important principles of decomposition that lead to simplicity are also not understood in fields such as medicine and aircraft design, that a focus on efficiency has taken precedence over simplicity and will ultimately lead towards system failures.

Herb Simon postulated that system complexity can be measured by hierarchy and that a law of hierarchy needs to be developed. In considering how technology is evolving with service oriented architectures and service-oriented creation of complex systems,there is a potential for flattening hierarchical systems as found in instantiations such as in large ERP systems. The key simplicity in such systems lies in determining **which kind of structure is helpful to design, organize and, ultimately, comprehend the system components.**

For decades, research and development in the field of software engineering structure has been almost instinctively associated with architecture. Composition and decomposition has traditionally meant that components can be plugged together and their (static) interfaces matched in some fashion. To effect such structure, our IT experts identified metamodels in the interviews as the primciple mechanism for describing the structural compatibility of architectures. These metamodels were mostly cast in terms of blueprints.While this perspective has proven useful, what is missing is the time dimension of systems that can be expressed in a real sense as "stories" as a key structural element. Structure does not exist only in space, describing the components that coexist and communicate, but also in time: what can logically happen before, after, during and, as alternative, in parallel to other actions. As a simple example, one may have a TV receiver in a car, which is a component of the bought product and structurally there all the time, yet its use might be restricted to situations where it does not interfere with navigation needs or functionalities for driving. To explain when the use of the ever present TV set is legal, we need to explain the behaviour – we need a story.

The behavioural structure of a system's functioning is still surprisingly neglected as a primary means of organizing (structuring), although it is well known that for throughout time stories have been the first and most extraordinarily effective means of transmitting fundamental and abstract information (e.g., entities, systems) to other individuals, including such important artifacts as concepts, values, and culture. Stories, myth, fairy tales are vivid and easy to remember in complicated contexts. They are also easy to tell, to pass on to others. Their potential for fascination, retaining attention and store in long term memory makes "story telling" and storyboarding a powerful tool to organize complex behaviours and interactions as plots and to turn the complicated into something which is understandable and, thus, simpler. While many HCI researchers characterize metaphors as possessing some aspect of this thinking, storied go beyond the notion of metaphor. Thus, professional methods for remembering many items suggest one should invent a story that connects the structural elements of a systems. In the world of ICT, such stories have taken on the personal of a use case, and are insufficient. More work needs to be created to better understand the use of narrative in developing simplicity principles in IT.

## 5.3. Orthogonality

Along the same theme of structure was the issue of system components being ‘orthogonal’. This design concept is keenly important when dealing with modularization. As was recognized by many researchers, we have techniques for decomposition and modularization, but when we move to synthesis, we experience difficulties. Putting components together can result in unexpected (and unwanted) system behaviours. These manifestations are referred to as ‘emergent behaviour’.

> *Complexity hides in the simplest setting. Sources of complexity for me are concurrency, possibility of failures, de-capsulated software and emergent behaviour. When you put things together which were designed and built independently you don't wish emergent behaviour. And we have not yet any overall science to deal with emergent behaviour.*

A key characteristic of IT systems that our participants deemed ‘simple’ are ones where the system acts in a **predictable** manner TM: don’t we have predictability later on as an own dimension?; given known inputs, one can expect known outputs. That said, in today’s world of IT expertise in system development, we find that system designs result in components that interact in manners in which we do not expect. What is not known is how to develop systems where such behaviours do not occur. In essence, systems where the components are orthogonal and the results are predictable are viewed as simpler (and more desirable) IT systems.

> *‘The trouble is if you take a big system and you separate the concerns down and you solve them and compose all back together – that’s one thing. But the question that was being asked is about integration of systems from different areas. One of the big things you get is they don't speak common concepts when they do so you have this awkward featuring interaction problem. And I think feature interaction is core to emergent complexity when you have two systems that are relatively simple and are simple in themselves. And then you compose them together. There are many ways to do that and some resolve this feature interaction in different ways. And the classical one is the phone system, where I press “call forwarding”, so if I am busy it forwards my phone somewhere else. And if I am busy I press “call waiting” which tells me which one of those gets precedence. And they are both nice ideas on their own, but if you have them together then…’*

The problems arise when orthogonal behaviours get combined in an uncontrolled integration. Feature interaction can be seen as a context-sharing problem (c.f. [8]). One might restore predictability by augmenting the context information.

It was frequently remarked that more informed approaches for composition/decomposition need to be created and that they must become clearer and easier to control than what we have today. A frequently mentioned example was the handling of multicores: they have become standard even in handheld smartphones, but it is not known today how to optimally use them in a simple way.

> *In practice, one needs to develop methods that combine theoretical issues, like normalization in database design, with performance issues, such as usability. While normalization is well known, decomposing processes seems to be more of an art. Moreover it is the decomposition of a system based on more than one criteria ... e.g., database normalization focuses on anomalies) but what if we wished to decompose a system which included data, process, networking, etc?*

## 5.4. Size

Research on successful manufacturers in Germany found that companies that specialize by having a smaller set of customers and a smaller set of products outperform their competitors. They are able to develop economies of scale and to fully develop an understanding of the needs of their customers. Steven Jobs when he returned to lead a floundering Apple company in the 90's reduced their product offerings from 50+ products to less than 10.

Our participants were clear in stating that simplicity is related to 'size', be it number of functions to produce, number of modules, etc. That said, size can also be a surrogate for other issues such as allowing one to *focus* as in the case of the business example just stated. By having fewer customers, the company was able to focus limited resources on fewer customers and on fewer products thus assuring a closer fit between delivering the right product to match their customers' needs. Size, in this example, becomes more a matter of managing range and variants. IT systems which are smaller, such as those that do just one task, are perceived as simpler and more easily managed. The goal of simplicity in IT then is to reduce aspects of systems, such as the number of functions offered, to the *core* ones desired by users. This task was perceived by our participants as being extremely difficult because users often do not know their needs and developers do not know how to easily extract user requirements, this task still being described as an art rather than a science. Getting at the core is critically important in a quest for simplicity and guidelines, both analytical and managerial need to be researched.

> *Just of the top of my head I think the integration is the most complicated. We kind of agreed during our discussions that simple means more or less small, deterministic, predictable and whatever. But in order to achieve functionality you need to somehow put together these simple concepts or components and integrate them, make them interact in ways. And this makes things complex.*

## 5.5. Transparency

> *'One thing that is important is transparency and being explicit about assumptions, basically having all facts on the table.'*

Transparency was described as a desirable system property that acts on the perception of systems: transparent systems are perceived as being simpler. While encapsulation was considered an important design concept, hiding details from the user, there was a tension in that not knowing various aspects of the system also made the perception of the system as more complex, less usable, and less desirable if too much was hidden.

> *'Another example where hiding things makes things worse is WYSIWYG editors. For me Latex is infinitely simpler than Word or PowerPoint. These things drive me crazy because you find 50 buttons, you don't know what happens when you push one of them; something screws up and you have no idea how to fix it. This is about explicitness; there are a lot of hidden assumptions. Think of having transparency to a degree where you really understand something to the necessary detail - not further than that but to the necessary detail - this is something that supports simplicity.'*

Virtualization is a key concept to steer what is transparent and what is hidden, shielded, or delegated.

More research needs to be perform that defines when (and what) aspects of systems should be made explicit to the user.

## 5.6. Predictability

Concomitant to transparency was the concept that systems whose behaviour and structure are predictable (i.e., no surprises) are perceived to be simpler. Predictability arises from the user's knowledge of the system, but also from the inherent design of the system. Systems which have probabilistic components lead to less understanding. NASA, for example, espouses this view as a major design factor in there IT creation. They wish systems to be deterministic in nature and in practice.

> *'Simplicity means: No surprises.'*

Many researchers praised methods, techniques, and tools that help spotting conceptual errors or under/over-specifications early in the lifecycle of a system. For example, the main contribution of Model Driven Design was repeatedly mentioned to be the existence of "useful" models, phrased as models that can be used in interdisciplinary teams and that can be "interrogated" in various fashions, via simulation, (formal) verification, and brought to life as executable prototypes. Early feedback to the user, early feedback to the designers, early discovery of mismatches in the descriptions of different facets of a system were repeatedly named to be fundamental assets for the engineering of complex systems, each of them contributing to eliminate, or at least to mitigate through early discovery, the surprises systems used to give only at testing, deployment, or usage time.

## 5.7. Communication

Transparency and predictability taken together led often to the fundamental question of how much, how, and when to communicate. Experts in Human-Computer Interaction mentioned repeatedly Grice's Cooperative Principle [11] and the 4 Gricean maxims: *quality* (Be Truthful), *quantity* (How Much Information), *relevance* (Be Relevant), and *manner* (Be Clear). Grice himself summarized it as

> *"Make your contribution such as it is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged."*

This observation on how to proactively communicate was complemented by other experts mentioning Nonaka and Takeuchi's theory of organizational knowledge creation [12], where the ever growing knowledge of an organization is accrued, shared, and persisted according to the SECI model (Socialization, Externalization, Combination, and Internalization) as the model that explains and steers the sharing and transfer of knowledge between humans, either directly (via socialization) or mediated by some information support. Dealing with explicit and implicit knowledge and with the "right" amount of context information was mentioned as one of the crucial points in handling perceived simplicity.

## 5.8. Automation

One researcher characterized simplicity as a measure of how little the end user has to do. If one can have a working system where the user consistently has to do less, then the system is easier to use. Hence one sees defaults on the low end and adaptable systems that learn user needs/preferences on the high ender end.

> *That is the more interaction you have between different things the worse it gets … This is why in most of our research for us making things simpler has been automation, doing it for the person.*

> *'I have never really worked on simplifying the code I write because that is not really my strength as a developer. Most of the research I have done has been in making principles in IT simpler for people to use which means automating, for example compositional verification which was hard to do manually. By automating it we made it easier for people to use it and apply it to new systems and for ourselves.'*

## 5.9. Abstraction

Abstraction was another common theme expressed by our participants as a driver of simplicity. By focusing on a right set of concepts and relationships, one can highlight the essence of the problem that needs to be addressed. While abstraction as an analysis and design technique is viewed as important, abstraction and the creation of models fell short for many of our participants. Some believed that achieving abstractions at the right level was difficult. Others thought that to achieve a higher level of simplicity models of end users were lacking, models that expressed users' intentions, interests, and motivations. Possessing better understandings of users would provide better designs.

In IT development, our participants stated that there often existed a mismatch of models between the various components of the IT and that if there was some means of creating a more robust scheme of integrating various models that IT overall would be simpler. UML, which touches on this concept, was thought to fall short in achieving this goal.

Simplicity is a question of abstraction - that on that level of abstraction it is simple.

## 5.10. Context and Subjectivity: *felt* Complexity

Many conversations characterized simplicity as a matter of subjective and context-dependent perception.

Complexity of systems comes in very different flavours and dimension, e.g.

- sheer size – of the solution itself or of the entities to be processed;
- conceptual complexity – the difficulty to understand any potential solution;
- and heterogeneity – the problem of integrating numerous partners, (communication) technologies, tools, and devices,

Again, all of these flavours can be individually distinguished in inherent and actual complexity. Here, the inherent complexity is due to the tackled problem, and cannot be reduced without changing the problem, whereas the actual complexity refers to the complexity of an actual solution, which often is much higher than the inherent complexity.

Simplicity does not so much relate to a problem as such, but rather to the way how this is perceived by the various stakeholders. The actually felt complexity may in fact still be quite a different matter: 'Divide and Conquer', or the 'Separation of Concerns' may split a global complexity into a number of aspect-specific complexities, each of which, individually, may well be comfortably tackled at different times, by different people, with different means, often exploiting powerful standard solutions. For example, a global business process management system has an enormous essential complexity. However, designing a complex business process can be done independently of integrating the involved applications and devices, independently of managing the often thousands of corresponding process instances on a network, and in particular, independently of the construction of the Internet, without which worldwide end-to-end processes would be hardly possible. Such effects of `giant canons´ (here, e.g. the Internet) are premier examples for solutions that are `easy for the many, difficult for the few´. For example, designing a complex business process can be done independently of integrating the involved applications and devices, independently of managing the often thousands of corresponding process instances on a network, and in particular, independently of the construction of the Internet, without which worldwide end-to-end processes would be hardly possible.

In [9] we addressed the importance and the role of the **'felt' complexity**/ies, which of course is quite subjective: the 'felt' complexity of any system depends on the individual roles sensing it. Usually, the business process designer does not feel the complexity of the realization and of the enactment, let alone the complexity of the required infrastructure. Conversely, those responsible for the infrastructure may not feel the complexity of the business-critical End-to-End process, its legal and economic consequences, and its vital implications for the company.

The central issue for a good and informed design of complex applications is therefore a method that reconciles the subjective views and competencies of the individual stakeholders into an adequate joint communication and decision-making framework. The goal is to comfortably manage an adequate division of labour and allowing to easily exploit standards and available solutions in order to minimize the felt complexity for all stakeholders.

Reducing this kind of *felt complexity* [9] has to do with the right kind of virtualization, e.g. taken the Internet as global communication media for granted, but also with an adequate view on the original problem. A good example for the latter is tracing products in a logistic chain. For a long time, image recognition was considered a key solution to this problem, however with unsatisfactory results. The change to RFID technology and simple tagging caused a radical change. Such disruptive innovations reveal a central aspect of simplicity: the right view on the problem to be solved. In practice, it is often advisable to change the problem formulation in order to drastically reduce the required effort, an approach often quoted as *80/20 principle*. Whereas virtualization, or more generally divide and conquer solutions are well established for complexity consideration, the 80/20 principle is not. It has to do with an adequate judgment of cost/benefit relations in larger context, like total cost of ownership, change management, training of personal, etc.. It is in fact this additional dimension of freedom, which characterizes the essence of simplicity but makes it at the same time so difficult to grasp.

# 6. Conclusions

We have spent significant time speaking with IT and other experts on the concept of simplicity. It is a large field with many concepts and attributes. It is clear to us that work in this area will prove fruitful and that elucidating all of the thoughts will take significant time. That said, we have extracted a set of key thoughts that touch on much of what we believed we have learned from our work. These thoughts are listed below and organized as a set of questions.

**What is simplicity?**

- There are difficulties in **defining simplicity** and its impact on IT development and use. Moreover there are difficulties in defining what happens when simplicity is not present. While it is said that IT seems to focus on efficiency, it is certainly not the case that these two are simply trade-offs with each other. The models and principles we have discovered needs more work to both organize and understand them, and then to turn the concepts into tools and techniques.
- Along the same lines, a definition of simplicity to be effective would need **a set of metrics** to measure one's progress (assuming simplicity falls on a scale).

**What are the status of research and the level of consciousness about simplicity as a value right now**?

- Researchers and practitioners in all the IT domains researched view simplicity as an enormously **important concept** in the design and development of IT.
- Simplicity is viewed as providing the **foundation** for important IT characteristics such as reliability, usability, stability, and modifiability, among many others. How it does so is not only characterized in an empirical fashion but also anecdotally.
- There is no unifying framework of simplicity that guides research or IT design and development. While recognized as enormously important, this key area remains a **fragmented** set of concepts, theories, and methods.

**What is the scope of simplicity?**

- Simplicity, where it exists at all as a focal point, occurs within the Human Computer Interaction (HCI) area where the focus is however just on only one aspect of IT. Also colloquially, simplicity is often associated with user interfaces. Work needs to be done to **expand this focus**.
- Increased simplicity in one aspect of an IT system may lead to **increased complexity** in other parts of the system. While this is an interesting thought, no language or thinking has been developed to understand this critical component of IT design and development.

**How to achieve simplicity? There is no silver bullet:**

- Simplicity may be based singularly on **knowledge**. That is, when researchers discuss simplicity, a key question is for whom? A knowledgeable user may find a system simple while a less knowledgeable user may not find the system simple at all. Thus, there is no absolute measure of simplicity, but a relative one. This, among many other topics, is an idea that needs to be explored.

- If simplicity is based on knowledge, then issues arise when systems are so complex that the designers of different components do not share a **working vocabulary** that allows them to combine systems without a high degree of potential failure. Work needs to be done to develop clearer, **situated communication** among complicated system components. If layering and decomposition are important instruments for creating simplicity, then techniques of synthesis and composition need to also be more fully developed.
- We do not know how to deal with composition (synthesis of system components) in a good way, **feature interaction** causes too many problems and so the major paradigm of decomposition and layering does not work well for overlapping yet independent contexts.

**What are open issues that would advance the understanding and adoption of simplicity as a leading guideline in designing and implementing systems?**

- One problem is that IT is an artificial science rather than a natural science and so it may have artificially created an environment of IT that does not allow a kind of natural simplicity.
- We are at a loss for designing simple systems. What makes matters worse is that our techniques for modifying systems, and standard maintenance initiatives seem to make systems more complex. We have no way of dealing with maintaining simplicity in the face of change. Simple systems lose simplicity through maintenance and evolution (modifications, add-ons, etc.)
- Reasoning about normal behaviour is difficult enough, reasoning about special cases and exceptions makes the task impossible. So how can one deal with simplicity without having to enumerate all the special cases? Do special cases make systems complex? Is that where the problem lies?

# 7. Recommendations

The discussions with our experts provided a strong foundation of ideas and motivations on why and where (and often how) to focus support for building concerted interest and action among researchers and industrial constituents on the principle of simplicity. They typically provided very specific hints on concrete needs and on how to provide incentive and support the interest of the researchers and industrial uptake. For the most part, a significant amount of direction came from our experts who are **active in industry** and from those in organizations that closely collaborate with industry. These individuals had middle to long-term initiatives in mind as well as specific ideas and concrete actions to be taken shortly. The impact thus ranged from help for the individual organizations in the short term to the development of a EU-wide simplicity initiative. This set of ideas is per se not adequate as a recommendation at the program or strategic level, however, we believe it illustrates acceptance and desire for support and incentives to develop simplicity as a leading value in IT throughout society. These ideas and motivations provided to us a foundation on which to define recommendations and proposals of measures as presented in this section of our report. This summary statement is gleaned from these ideas, which have a very clear and succinct common message:

> The widespread and systematic adoption of **simplicity as a leading value** in the ICT and European society can become a key competence and capability of the European research and economic space in the next two decades and beyond.
>
> We need to develop in the short to mid-term a **community of practice** that researches, adopts, adapts, extends, and systematically promotes in Europe a culture of simplicity in the ICT domain.

If the EU does not adopt a culture of simplicity in IT research, design, development and use, other significant actors such as Silicon Valley and the BRICS countries will make this transition first, and leave our cultural and socio-economic space trailing in the highly competitive race for new initiatives based on simplicity, on developing high profile research and industry sectors and, importantly, new value and value propositions for government, business and society. The concomitant benefits that will arise from embracing this critically important concept pervade the entire society.

Towards this end, we identify and briefly discuss four specific priority strategic areas of recommended intervention:

1. Foster **communication and networking**
2. Expose the **economics** of simplicity
3. Develop **key technical** simplicity directions
4. Foster simplicity in **education and innovation**

These strategic areas of initiatives are described in the following sections and we conclude with a proposed timeline for their realization.

## 7.1. Foster communication and networking

The first step is to help simplicity to emerge from the underground and establish it as a value in the existing communities.

A central barrier to the adoption of simplicity as a motivation for research and practice is the lack of communication on the initiatives, techniques, and experience that are already available. While pockets of research related to simplicity as well as centres of practice that value simplicity exist, located in a few academic institutions, often this research and experience are not shared. To the contrary, research that may be useful in the development of simplicity as a discipline is typically hidden or considered anecdotally or in a peripheral fashion to the researcher's primary agenda. Such hidden work then is not disseminated, thus remaining unknown, with other researchers being unaware of the work performed.

We propose that steps be taken towards the creation of an interdisciplinary community of practice on simplicity. The first goal in this direction is to make simplicity more salient to the researcher and bring them into contact with industry practitioners with the goal of information sharing. In particular, actions such as the following could occur:

1) Interdisciplinary conferences specifically devoted to simplicity could help establish and then maintain and grow a community of interest and then a community of practice for this topic.

2) Forums, social networking and expert-based communication platforms in the form of a "Simply-pedia" could be established to document and build on expert advice regarding simplicity.

3) Similar to the business model successful for Open Source Software, an open and free first level offer could originate an ecosystem of specialized companies that offer commercial advice and services.

To implement these suggestions at a European level, a **coordination action** was repeatedly suggested. This foots in the evidence that there is in the EU sufficient interest, competence, and drive to plant this way the seeds of the necessary coordination and communication by making explicit and visible the expertise and knowledge that are currently submerged. We also believe that there is sufficient pressure by the economic stakeholders to take up these platforms in a self-sustaining fashion after the initial CA- based impulse.

## 7.2. Expose the economics of simplicity

Through our discussions, many experts of ICT and other areas (economy, healthcare) believed that there are real economic outcomes to be achieved from a focus on simplicity. This possibility has already been documented in the work by Rommel et. al. who studied simplicity principles at work in German manufacturing. Thus, a specific proposal concerns revealing, quantifying, and showcasing the economics of simplicity.

Such a proposal could take many forms including applied research into key areas (e.g., hospital information systems) as well as financially supporting initiatives focused on the identification and adoption of simplicity methods and technologies. That is, the adoption of simpler products and simpler solutions against the market trends is a risk, which is usually associated in companies and organizations with costs aimed to offset this additional risk. In the best tradition of an incentive-

oriented measure, a certain compensation and encouragement for the adoption of (certified) simpler techniques could be introduced, until those technologies become mainstream.
Measures supporting **co-creation and adoption of simpler methods, technologies, and products** could target the economic sector that is likely to benefit most of the advantages: the SMEs.

**SMEs** have proven difficult to reach and to mobilize within existing research programs, sometimes even in spite of the specific focus towards SME majority or leadership in consortia. Specific targeted measures that keep the consortium small and the overhead low, coupled with direct addressing of the individual SME's needs might be realistically successful at a European scale, and directly impact the most dynamic and sensitive segment of the economy.

## 7.3. Develop Key technical simplicity directions

To achieve progress in creating a specific community of practice on simplicity, both financial and organizational support and encouragement for **targeted research** were most often mentioned by our expert population. While the first section proposed lifting a community of practice to public awareness and the second section proposed increasing our understanding of and openness towards the economics of simplicity, this section identifies some key technical issues that researchers in simplicity could address. The recurringly mentioned issues are as follows:

**Representations / abstractions**:
- How to keep consistency between models and between representations.
- How to deal with abstraction in a systematic way, crossing areas of competence and areas of responsibility.

**Increased or self-Explanatory power**
- How to make the technologies more intuitively explainable to users as well as developers.
- How to reconcile the need for specificity in explanations with the need for intuition, especially when different profiles of the engaged professionals imply different backgrounds and cultures

**Managing change**
- When to branch and when not to in products, releases, and technologies
- How to manage the trade off between compatibility with previous versions/platforms and clear-cut approaches that really lose the link to the past.
- Focus on the core activity rather than on other side characteristics and added features

**Transparency**[11]
- What is inside a product (components, built-in modules, external technology that is integrated directly by the vendors)
- What are the preconditions for the use of a technology (e.g. technical infrastructure or communication platforms)
- What is the optimal usage profile and what are more or less recommended usages

[11] This was mentioned in analogy to what is being today increasingly adopted for food and nutrition products. There, the consumer is now better informed about ingredients (more detailed list), about the composition of the product (how much sugar, fat, carbohydrates and proteins it contains), about the relation to her/his own needs (e.g. in form of % of the daily needs per portion), and often also calories.

A compact but standardized information fact-sheet for complex technological products and technologies are currently not available. Even off-the-shelf products, that should be directly adoptable by SMEs and deployable on their own, without need of consultants or significant deployment projects, this kind of information is hard to get together.

## 7.4. Foster simplicity in education and innovation

Two critical components of the future success of the EU are education and innovation. Our experts believe that simplicity remains an unspecified, important aspect in each of these domains. While critical thinking has been identified as essential we suggest that simplicity be made a more salient topic. In sum, we propose the following:

1) Promote simplicity as essential part of education, supporting the creation and diffusion of curricula and courses that present simplicity as a key value in critical thinking and in the IT creation chain in particular.

2) Promote simplicity and a specific topic in the innovation-related programs.

3) Help enforce the change that is needed in current educational structures and practices in order to adopt simplicity thinking while fostering the change management initiatives that precede and ultimately lead to the adoption of this educational mission and through offering support and know-how to make it happen successfully.

## 7.5. Suggested timeline

**Directions 1 and 2)**

*The coordination and networking and the explicitation of the economic returns on simplicity were generally perceived as activities for **immediate action**.*

Accordingly, we plan to submit a CA proposal in Call 8 under Objective ICT 2011.9.12 in order to leverage the impetus we created with the present ITSy strategic action in the several communities addressed and turn it into a much more visible platform for networking and coordination over the next 3 years.

**Directions 3 and 4)**

*The targeted research and the large-scale education and innovation initiatives were perceived as activities that could start in the **medium term**, as soon as a critical mass of proactive response is available.*

Ideally, Simplicity in ICT could become a new specific topic inside the FET Proactive program, and be featured as an Objective in future calls, attracting project proposals.

In parallel, advanced education, knowledge transfer and innovation initiatives could be pursued within the PEOPLE program, using instruments of the Marie Curie actions.

# 8. Literature


[1] B. Kitchenham: *"Procedures for performing systematic reviews"* Technical Report TR/SE 0401, Keele University and NICTA, 2004

[2] H. Reinders, H. Ditton, C. Gräsel, B. Gniewosz: *„Empirische Bildungsforschung. Strukturen und Methoden"* Wiesbaden. (2011)

[3] M. P. Couper: *"Web Surveys. A Review of Issues and Approaches"* In: Public Opinion Quarterly Vol. 64, Iss. 4 , Page 464-494. (2000)

[4] M. Maurer, O. Jandura: *"Masse statt Klasse? Einige kritische Anmerkungen zu Repräsentativität und Validität von Online- Befragungen"* In: N. Jackob, H. Schoen, T. Zerback: *„Sozialforschung im Internet. Methodologie und Praxis der Online-Befragung",* Wiesbaden. (2009)

[5] Y. Baruch: *"Response Rate in Academic Studies - A Comparative Analysis"* In: Human Relations Vol. 52, Iss. 4, Page 421-438. (1999)

[6] O. Dieste, A. Grimán, N. Juristo: *"Developing search strategies for detecting relevant experiments"* Proceedings of the First International Symposium on Empirical Software Engineering and Measurement (2008)

[7] J. A. Zachman: "*A framework for information systems architecture"*, IBM Systems Journal, Vol. 26, No. 3, S. 277-293 (1987)

[8] A. Nhlabatsi, R. Laney, B. Nuseibeh: *"Feature interaction as a context sharing problem"* In: International Conference on Feature Interactions in Telecommunications and Software Systems, 11-12 June, University of Lisbon, Portugal (2009)

[9] T. Margaria, B. Steffen: *"Service-Orientation: Conquering Complexity with XMDD"* In M. Hinchey, L. Coyle (Eds.): "*Conquering Complexity"*, Springer Verlag, ISBN 978-1-4471-2296-8, (2012 - in print)

[10] W.H.J. Feijen, A.J.M. van Gasteren, D. Gries, J. Misra, (Eds.) *"Beauty is our Business"* - A birthday salute to Edsger Dijkstra., Springer Verlag, ISBN 0-387-97299-4 I. (1990)

[11] P. Grice: "*Logic and conversation*" In: Syntax and Semantics, 3: Speech Acts, ed. P. Cole & J. Morgan. New York: Academic Press. Reprinted in Studies in the Way of Words, ed. H. P. Grice, pp. 22–40. Cambridge, MA: Harvard University Press (1989)

[12] I. Nonaka, H. Takeuchi: *"The knowledge creating company: How Japanese companies crate the dynamics of innovation"*, Oxford University Press, pp.284, New York, 1995.