TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

A Generic Model for Sustainability

Birgit Penzenstadler, Henning Femmer

TUM-I1219

A Generic Model for Sustainability

Birgit Penzenstadler, Henning Femmer

Technische Universität München, Germany
penzenst|femmer@in.tum.de

November 26, 2012

Abstract

Motivation: Software systems as we know them have either a primary economic purpose or fulfil human or social needs of their users. The economic purpose is analysed in depth by the economy itself; the latter goals are analysed in software engineering by service orientation and usability analysis.

Yet, as software systems are involved in nearly all daily business and non-business processes they also have an impact on resources and environment. Hence, besides economic, human and social goals, environmental sustainability should be a major goal for software development projects.

Problem: Without a tangible definition and applicable guidance, sustainability remains an unreachable ideal. Therefore, we need a definition and a concrete decomposition of sustainability to make it tangible for software systems development.

In addition, due to second and third order effects, it is not sufficient to analyse environmental sustainbility on its own, but its interplay with economic, social, human and technical sustainability in order to define appropriate actions and understand their effects.

Principal idea: We analyse the dimensions of sustainability, their values with respective indicators, and activities to support them. These elements are composed to a conceptual model that allows for analysis as well as constructing an outline of actions both for a company or a product point of view.

Contribution: We propose a generic sustainability model with a meta model and with illustrating instances for companies and projects that are taken from various case studies. By building an tangible model we make these effects explicit and thus enable analysis, support and assessment of environmental sustainability in software engineering in a broad view.

Keywords: sustainability, software system, goal, requirements, definition

Contents

Chapter 1

Introduction & Motivation

Sustainability has been recognized as a relevant topic in software engineering, e.g., by the 2012 ICSE theme and a number of workshops. It refers to environmental, social, and economic aspects of software development and the usage of software systems.

When looking at software systems and these aspects of sustainability, we find that most software systems usually are developed with an economic benefit in mind and that this benefit shall also last over a longer period of time. In case of software systems that are not intended to serve economic purposes, for example, open source software, the main purpose of the software system is a social benefit which can be making specific work tasks easier or just plain entertainment and having fun.

Taking into account that the economic and social aspects of sustainability are already present in most software systems, our work especially focusses on supporting the environmental perspective. Our proposed approach shall help integrating the objective of environmental sustainability in and via software systems with other system development objectives.

Hilty et al. [1] provide an analysis of the relevance of information and communication technologies for environmental sustainability and conclude that there is no such thing as a "general ICT policy for environmental sustainability" [1].

Especially for an adequate analysis of the latter, we need a tangible decomposition of the concept of sustainability and supporting methods in requirements engineering. These methods can then enable to include the concept of sustainability during requirements engineering and help to develop such a general ICT policy for environmental sustainability over time.

Problem Currently, we are still missing a definition of what software engineering can contribute to improving the sustainability of the systems under development (except for some aspects of Green IT). Thus, sustainability is not tangible enough as a concept to actually be able to transform it into software requirements.

There is no guidance available describing how to decompose sustainability for a concrete project and we are missing procedures for incorporating sustainability as an explicit goal into requirements engineering.

Apart from the decomposition, we are also lacking a reference catalogue of sustainability-improving activities linked to indicators they affect. These activities and indicators would allow for a sustainability assessment in order to evaluate whether the activities affect the sustainability of a project or company.

Contribution In this report, we present a reference model for sustainability that decomposes sustainability into five dimensions: environmental, human, social, economic, and technical sustainability. The model provides a number of activities and relates them to the values they support and the indicators they can be assessed with.

It is intended to serve as a reference model for a *process engineer* who instantiates the model for a software development company or for a *requirements engineer* who instantiates it for a specific system under development. They instantiate the values in the model with system-specific goals and derive specific actions from the activities, thus accounting for the realisation of the system's (or company's) sustainability goals. The instantiation is illustrated by two examples taken out of current case studies.

A specific aim is to show how supporting environmental sustainability can be aligned with the other sustainability dimensions, as environmental sustainability is the aspect that is least supported by our current ways of developing software systems.

Outline The report is organized as follows: Chapter 2 discusses related work and positions the contribution. Chapter 3 presents our understanding of sustainability and how it builds on and relates to existing definitions. Chapter 4 describes the stakeholders for the different aspects of sustainability and their potential roles. Chapter 5 presents the generic sustainability model and its meta model. Chapter 6 describes the process of how to instantiate the models. Chapter 7 presents the application of the instantiation in a number of examples and case studies. Chapter 8 draws conclusions and points out future work.

Chapter 2

Related Work

Related work can be differentiated into sustainability modeling approaches, modeling approaches from the economic domain, definitions of sustainability, and analysis of sustainability in specific domains.

2.1 Sustainability Modeling Approaches

Sustainability is starting to be acknowledged in software engineering, with the Journal on Sustainable Computing, the conferences on Sustainable Computing, the conference series EnviroInfo, the 2010 Workshop on Services, Energy and Ecosystems, the 2009, 2010, and 2011 Workshops on Software Research and Climate Change, the 2012 Workshop on Green and Sustainable Software (GREENS), and the 2012 Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy) at REFSQ'12.

Framework for sustainable software engineering Naumann et al. [2] provide a framework for sustainable software engineering. They investigate how web pages can be developed with little environmental impact, i.e., energy-efficiently, and offer a respective guideline for web developers. In such a framework, the reference model proposed in the work at hand could be used as key element in order to better promote the method.

Strategy models Gu et al. [3] propose a green strategy model that provides decision makers with the information needed to decide on whether to take "green" strategies and eventually how to align them with their business strategies. They consider such strategies as "green" which achieve lower energy consumption and perform a case study with Dutch data centers. In contrast, the paper at hand considers a broader definition of sustainability and thus gives a broader view on sustainable software engineering.

i-star modeling Cabot et al. [4] report on a case study for sustainability as a goal for the organization of the ICSE'09 conference with i*-models to support decision making for future conference chairs. Stefan et al. [5] extend that work for managing environmental sustainability with quantitative goal modeling techniques. Both works provide model instances for specific case studies while the work at hand also provides a generic reference model.

Requirements engineering techniques Mahaux et al. [6] present a case study on a business information system for an event management agency that advertises environment-friendly events. They assessed how well some current RE techniques support modeling of specific sustainability requirements in that case study. In contrast, our aim is to provide modelling means explicitly for integrating sustainability into the software development process as a major objective.

Goal Modeling Lamsweerde [7] decomposes business goals into system requirements, but does not explicitly reflect on sustainability. His work relates positive and negative influences between goals, but our work provides activities for direct realisation.

2.2 Economic Domain: Balanced Score Card

One exemplary approach from the economic domain that can help to operationalize and monitor sustainability in a company is the Balanced Score Card (BSC) by Kaplan and Norton'92 [8]. They combine financial and operational measures. Their used perspectives are:

- Customer: How do customers see us?
- Internal: What must we excel at?
- Innovation and learning: Can we continue to improve and create value?
- Financial: How do we look to shareholders? [8, p. 4]

Matching of BSC perspectives to SRM dimensions as instantiation One major goal of the instantiations of the BSC is the monitoring and improvement of the financial prosperity of the implementing company. The major goal of the instantiation of the GSM for a company is to monitor and improve the sustainability of that company.

The balanced score card (BSC) can be seen as instantation of the generic sustainability model (GSM). The goals identified in the BSC are thereby instances for the values of the GSM. The goals do not necessarily refine the value into a more specific goal but they may do so.

2.3 Definitions of Sustainability

The background for the presented definition approach is structured into frameworks for defining sustainability, the relation between ICT and sustainability, and earlier approaches to defining sustainability in the context of software engineering.

2.3.1 Frameworks for Defining Sustainability

The following frameworks are designed to serve for defining sustainability in general, without a specific reference to software systems. We present them to emphasize the necessity of adequate definitions and use them as context to discuss our proposed definitions.

Dobson [9] suggests a framework for the comparison of sustainability notions that poses conceptual questions, which any conception of sustainability must answer: What is to be sustained? Why is it to be sustained? Who/what is concerned? In what respect is substitutability allowed? This is sufficient for a topology for comparison but avoids formality.

Burger and Christen [10] propose a capability approach of sustainability. To avoid "yet another sustainability conception", their methodological approach is to first formulate adequacy conditions for concepts of sustainability, then illustrate a categorial framework with the required general concepts, and finally propose a conception of sustainability based on the capability approach. They consider the idea of sustainable development as a problem-solving strategy (for the "developmental dilemma" [11][1]) and their adequacy conditions are future-orientation, normative power, justice, universality, limited natural conditions, and high-level strategic actions [10, p. 788]. The developed approach is supposed to serve as basis for empirical research in development studies.

Robèrt et al. [12] present the Framework for Strategic Sustainable Development that is now mainly promoted by the Swedish NGO 'The Natural Step'[2]. Their objective is to show how the increasing number of tools and approaches to develop sustainability (e.g., the ISO14001 [13], Life Cycle Assessment [14], and Ecological Footprinting [15]) relate to each other and build on each other when used for planning for sustainability. The framework consists of five interconnected levels: the understanding of constitutional principles of the system, principles of sustainability, principles for sustainable development, activities for sustainable development, and tools for monitoring compliance and impacts of the actions. Thereby, it reflects sustainability from the foundation to the practice. Following a constructive rather than an analytic approach, this framework also avoids formality but is intended as guideline.

Following a stricly analytic approach in a theoretic contribution to the discourse in sustainable development, Christen and Schmidt [16] counter-propose

[1] Developmental Dilemma: Millions of people are in need of societal and economic development to surpass the poverty line, however this development could overstrain the natural conditions necessary to guarantee further development. [11]

[2] `http://www.naturalstep.org/`

(w.r.t. the approach by Robèrt et al. [12]) a Formal Framework for Conceptions of Sustainability that intends to solve the problem of arbitrariness. The framework consists of five modules that serve to help to elaborate the elements answering the 'what' and 'how' questions. The modules are the sustainability problem, the normative principle of justice, the descriptive principle of integration, the criteria for sustainability, and the transformation into practice.

2.3.2 Defining Sustainability in the Context of Software Engineering

Penzenstadler et al. [17] provide a systematic literature review that points to two definitions of sustainability in the context of software engineering, i.e. by Mahaux et al. [6] and by Naumann et al. [2].

Mahaux et al. base their research on the Brundtland definition [11] plus the statement that IT changes behavior and therefore has considerable effect on society and environment, which is supported by greenIT concepts and analysis of the usage processes of a software system [6].

Naumann et al. define "green and sustainable software" such that "direct and indirect negative impacts on economy, society, human beings, and environment that result from development, deployment, and usage of the software (...) ha(ve) a positive effect on sustainable development", and "Green and Sustainable Software Engineering" such that the "negative and positive impacts on sustainable development (...) are continuously assessed, documented, and used for a further optimization of the software product" [2, p. 296].

These two works already show that the term sustainability is strongly dependent on the taken perspective. Even the second definition cannot yet be operationalised, which is crucial if SE researchers want to contribute.
Consequently, scope and context have to be clearly defined to be able to make any statements. A first approach is taken from systems thinking [18] by using three general parameters: Which *system* shall be preserved in which *function* over which *time* horizon?

2.4 Analyses: Relation between ICT and Sustainability

Hilty, Lohmann, and Huang [19] provide an overview of the fields of ICT in the service of sustainability: Environmental Informatics, Green IT, and Sustainable Human-Computer Interaction. As technological efficiency alone will not produce sustainability (cf. Jevon's paradoxon [20]), they state that sustainable development requires a combination of efficiency and sufficiency strategies, inter alia by decoupling economic growth from environmental impacts and from the use of natural resources.

Furthermore, Hilty, Arnfalk, Erdmann et al. [1] analyse the relevance of information and communication technologies for environmental sustainability. They present the impacts of ICT on environmental sustainability on different levels: first order effects like increasing electronic waste streams, second order effects such as improved energy-efficiency of production, and third order effects like a product-to-service shift or rebound effects in transport. In their prospective study, they present an analysis of the anticipated future (positive or negative) environmental impacts of different types of ICT applications: ICT applications supporting a product-to-service shift, for heating management, for passenger transport efficiency, for mobile work, and for freight transport efficiency.

With a wider scope, the Smart 2020 report [21] by the Climate Group presents a way of enabling the low carbon economy in the information age. The book Vision 2050 [22] by Hiroshi Komiyama and Steven Kraines presents a roadmap for a sustainable earth.

Chapter 3

Dimensions of Sustainability in Software Engineering

This chapter presents the dimensions of sustainability that serve to identify the stakeholder for sustainability in Chapter 4 and represent the dominant decomposition means for the generic sustainability model in Chapter 5. Furthermore, this chapter discusses their relations, and introduces the concept of value that is also a building block for the generic sustainability model.

3.1 Dimensions of Sustainability

We consider five dimensions of sustainability as important for the analysis of software systems. The basic dimensions for a general sustainability analysis (without referring to software systems) are human, social, economic and environmental as defined by Goodland [23]. The three latter ones are the dimensions also known from the UN definition of sustainable development [24].

But these four dimensions do not offer a possibility to claim and support long-term evolution of technical systems and adequacy for long-term use. Consequently, when looking at (software) systems, we need technical sustainability as an additional dimension.

Human sustainability Human sustainability refers to the maintenance of the private good of individual human capital. The health, education, skills, knowledge, leadership and access to services constitute human capital [23].

Social sustainability Social sustainability means maintaining social capital and preserving the societal communities in their solidarity. Social capital is investments and services that create the basic framework for society [23].

Economic sustainability Economic sustainability aims at maintaining assets. Assets do not only include capital but also added value. This requires to define income as the amount one can consume during a period and still be as well off at the end of the period, as it devolves on consuming added value (interest), rather than capital [23].

Environmental sustainability Environmental sustainability seeks to improve human welfare by protecting natural resources. These are water, land, air, minerals and ecosystem services; hence much is converted to manufactured or economic capital. Environment includes the sources of raw materials used for human needs, and ensuring that sink capacities recycling human wastes are not exceeded [23].

Technical sustainability From a point of view of (software) systems engineering, there is another dimension that has to be considered. Technical sustainability has the central objective of long-time usage of systems and their adequate evolution with changing surrounding conditions and respective requirements.

These five dimensions are used in the sustainability model to decompose sustainability into more tangible units.

The dimensions Social, Economic, Environmental and Technical can be analyzed on a mirco as well as on a macro level. The decision on which level they are investigated depends on the scope of the system under analysis.

These five dimensions are not necessarily encompassing. One can argue, for example, for considering politics & law as separate dimension instead of including it in the social dimension. On the other hand, the government can be considered as institutionalization of society and therefore can be treated as subdimension of the social dimension.

However, for the analysis scope of software systems, we believe the five given dimensions to be an adequate representation as they serve only as structuring means for the to-be-derived model.

3.2 Relations between the Dimensions of Sustainability

Naturally, the dimensions are strongly related to each other. Figure 3.1 depicts the dimensions as layers for how the dimensions generally build upon the others. This is one of various possible visualizations and is rather meant to be interpreted as a general idea than as an apodictic explanation.

Other common visualizations depict the dimensions as concentric or intersecting circles, either of which serves to strengthen a particular aspect in discussion. However, for our purpose of explaining which dimension of sustainability is a precondition for another one, the layered visualization is the most suitable one.

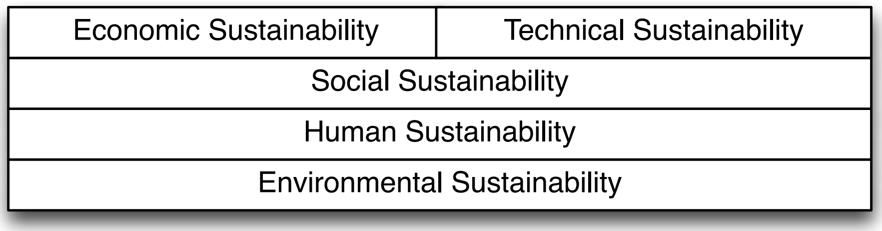

Figure 3.1: Dimension Relations

Environmental S. — Human S. Environmental sustainability is the basis for all dimensions. Without natural resources, life is impossible. A healthy human is also part of environmental sustainability. Therefore, environmental sustainability is the basis for human sustainability.

Human S. — Social S. In turn, humans are the individual elements that society is based on. Therefore, human sustainability is the basis for social sustainability. On the other hand, humans define themselves strongly by their relations to other humans, so human sustainability can hardly be regarded independently of a social context.

Social S. — Economic S. Society is organized by individual humans, and institutionalized by means of a government and its infrastructure. This infrastructure also enables economy. Therefore, social sustainability is a prerequisite for economic sustainability. On the other hand, economy again interacts with society by providing wages for its employees, which is fed back into community by supporting family and infrastructure, and by establishing small communities of colleagues and customers.

Economic S. — Technical S. Technical systems are developed within a regulated economical context. At the same time, technical sustainability, the longevity of systems and services, is a fundamental element of the economy in terms of goods and services that are traded in that economy. Therefore, economic and technical sustainability depend on each other to a certain extent. Economy motivates research and development and vice versa.

3.3 Resulting Central Questions for Analyzing Sustainability in Software Engineering

Splitting up the matter of sustainability into the five dimensions mentioned above leads to five software related questions to be answered by research, society and each individual project:

Environmental Sustainability How does software affect the environment during development and maintenance [25]?

Human Sustainability How can software be created and maintained in a way that enables developers to do their job happily over a long range of time?

Social Sustainability Which effects do software systems have on the society (e.g. communication, interaction, government, ...)?

Economic Sustainability How can software systems be created so that the stakeholders' long term investments are as safe as possible from economic risks?

Technical Sustainability How can software be created so that it can easily adapt to future change?

These questions are meant to be starters for further discussion and research. They are questions that build upon the definitions of the sustainability domains and are based on the values we consider important for the domain.

3.4 The Concept of Value

One central question laying the basis for the analysis of sustainability support is: What is the value notion behind sustainability? Is the notion nature-centered or human-centered?

Plato According to Plato, objects that are attributed values can be distinguished into objects with intrinsic value and objects with instrumental value. An intrinsically valuable thing is worth having for itself. [26]

In the context of our research, we understand a value as intrinsic, independent of a specific justification.

Kant A value is a reason by which an action or state is judged, which is only justified in itself (*atomicity*). For example when looking at working rights, most people agree that an employee's physical and mental health should not be risked through the work he or she is employed for. Various arguments support this goal, for example, the fact that a sick person is prevented from working for the company until health is reestablished. Yet, this argument can be abstracted up to the company's goal of long term prosperity. In contrary, another reason why the individual's health should not be risked, is a value that is perceived as being right in most cultures of our time: the physical integrity of the individual. This value can hardly be abstracted or based in a higher reason, which is why it can be classified as a value. [27]

In contemporary software companies this issue relates, for example, to working overtime.

Chapter 4

Stakeholders for Sustainability

This chapter details on the stakeholders for sustainability, their interests, and their relation to the sustainability dimensions introduced in the preceeding chapter.

4.1 Motivation

When discussing sustainability in the IT context very often people ask: Who are the stakeholders[1] for sustainability? This question is motivated from two backgrounds:

1. Domain-Knowledge: The stakeholders of a certain domain usually has widespread knowledge in this field. This leads to the point that the stakeholder can judge whether or not a certain indicator is a good approximation for a value, if suggested activies can be put in place, how the activities relate to the indicators, etc. For example, when discussing the impact of establishing use of green data centers, software developers, finance and advertising stakeholders should discuss the costs and benefits of this avtivity.

2. Stakeholder-Support: Besides the knowledge of the stakeholders, it is a well known fact in industry that many projects fail because of missing support from management stakeholders. The same holds for all other stakeholders in the company. For example, if the developer understands the rational behind a the decision to use a green data center, he will be more willing to support the idea, which will decrease the risk for the project.

[1] By a stakeholder we refer to anyone who has interest in or is affected by the system under consideration.

4.2 Overview of General Stakeholder Classes

Values differ by the community under analysis. When facing Sustainability in Software Engineering we usually refer to this community in terms of stakeholders, thereby including customers, users, developers and anybody else who has an interest in the system under analysis.

Exemplary classes of stakeholders in software engineering for the sustainability dimensions introduced in Chapter 3 are:

- *Management*: Managers concerns are represented primarily the economic dimension because their major task is to preserve and improve the value of a company's shareholders. As employees they are also subjects to the human and social dimension.
- *Employees / Developers*: Concerns of employees or developers are primarily represented in the social dimension in terms of the need for making a living and desiring a pleasant environment, but they are also subjects to the human dimension with basic needs like health and nutrition.
- *Suppliers*: Subcontractors are necessary for raw material, hardware and software, therefore they influence primarily the economic and technical dimensions.
- *Legal Representatives*: These stakeholders' concerns influence all dimensions of a system under analysis, as there are laws and regulations for human rights, environment and society as well as economy and technology.
- *Customers*: The clients influence primarily the economic and technical dimensions as they want to get value for their investment and want that value to be preserved.
- *Users*: The actual users of a system influence primarily the technical, but also the social dimension as they want the systems to contribute to some aspect of their life but generally will want them to last.
- *Destructors*: Companies occupied with recycling or disposing waste mainly take influence on the environmental dimension, but also on the economic one in case this affects costs for the customers or users.

These classes are not necessarily encompassing but intended to serve as checklist when analyzing the stakeholders for a particular system.

4.3 Stakeholders according to Sustainability Dimensions

For the elicitation of stakeholders once again the generic sustainability model can serve as a guideline. This is performed by going through the five sustainability dimensions and understanding which stakeholders are possibly affected by this dimension. In the following we will give a generic, very reusable list for possible stakeholders. This list is just a rough overview and in no means complete.

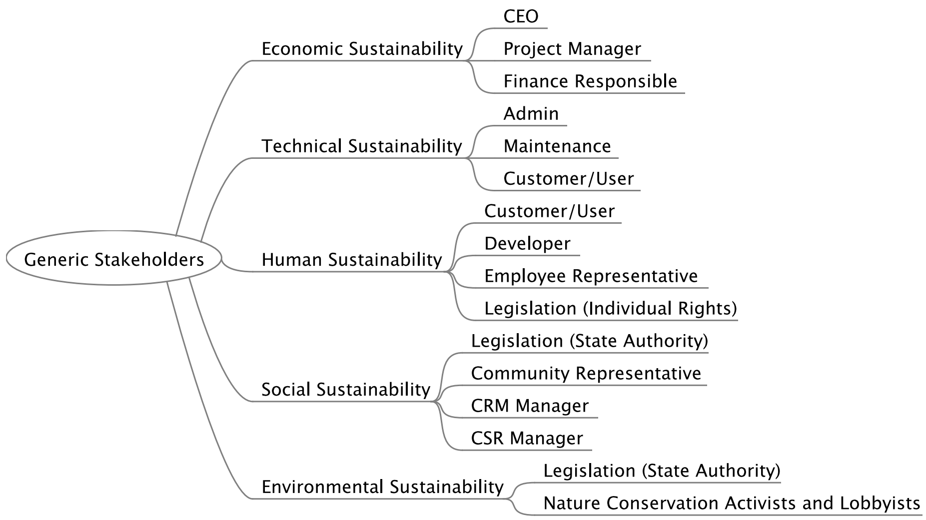

Figure 4.1: Generic Stakeholders of the Sustainability Model.

- Economic Sustainability:
  - CEO: The chief executive officer (CEO) is the highest-ranking executive of a company. The CEO has to integrate the sustainabilty goals into the company's vision.
  - Project manager: Especially when instantiating a product-specific sustainability model, it is important to have the project manager agree in what ways the project should support sustainble aspects.
  - Finance responsible: As sustainable software engineering affects many parts of the company, among others, many financial decisions have to be made to implement a sustainable software engineering model in a company.
- Technical Sustainability
  - Admin: The administrator of a software system has a very strong motivation for long-running, low-maintenance systems as it makes his work easier.
  - Maintenance
  - Customer/User: Users are often interested in persistency of the systems they are using []. This regards to user interface and required

soft- and hardware. Many users, however, are interested in gaining features with updates.

- Human Sustainability
  - Customer/User: The user must be able to use a certain system.
  - Developer: The developer is heavily involved in creating the system. This process needs to be careful about sustainable pace and growth of the developer.
  - Employee representative: The mental and physical safety of individuals during the development of a product needs to be maintained. Employee representatives watch the rights of the employees involved.
  - Legislation (individual rights): Systems must respect the right of their users. Hence, a legislation representative must respresent the privacy and data protection laws that are in place.
- Social Sustainability
  - Legislation (state authority): The state has a strong interest in understanding a systems influence on the society. In contrary to the individual rights legislation representative, the state authority representative speaks from the perspective of the state as a whole.
  - Community representative: In addition to the state authority, other communities such as the local government (e.g. the mayor) or non-government clubs might be affected by a software system. A complete analysis must take their views into account.
  - Customer Relationship Manager (CRM): The CRM wants to establish long-term relationships with their customers and create a positive image for the company.
  - Corporate Social Responsibiliy (CSR) manager: Some companies created a special position that initiates and implements a company-specific vision of its social responsibility.
- Environmental Sustainability
  - Legislation (state authority): Environment protection laws are in place to ensure sustainability goals. These laws must be reflected in the model.
  - CSR manager: The CSR manager is often also responsible for environmental aspects.
  - Nature conservation activists and lobbyists (e.g., WWF, Greenpeace, BUND)

Again, many stakeholders can be reused through various projects. However, in order to be of greater use, this list needs to be instantiated according to project or company needs.

Chapter 5

The Generic Sustainability Model

This chapter describes the generic sustainability model built upon the concepts introduced in the previous chapters.

Our proposed approach comprises the generic sustainability reference model (M1 level[1]), the respective meta model behind it (M0 level), and the instances (M2 level) that are derived from the reference model. These can be derived either for the development process in a specific software company or for a specific software system or product.

5.1 The Meta Model

The meta model is comprised by the types *Dimension*, *Value*, *Indicator*, *Regulation*, and *Activity*. The model including relations between the elements is depicted in Figure 5.1.

Type <Dimension> A dimension is an aspect of or viewpoint on sustainability, for example, "environmental sustainability". A dimension is represented by a set of values that express the abstract objectives of the dimension.

Type <Value> A value is a rationale that is rooted in itself, and is approximated by indicators.

Type <Indicator> An indicator is a qualitative or quantitative metric and is related to a value.

Type <Regulation> A regulation is an optional element belonging to a value.

Type <Activity> An activity is related to values that it supports and potentially improves.

[1] similar to the OMG's Meta Object Facility's levels [28]

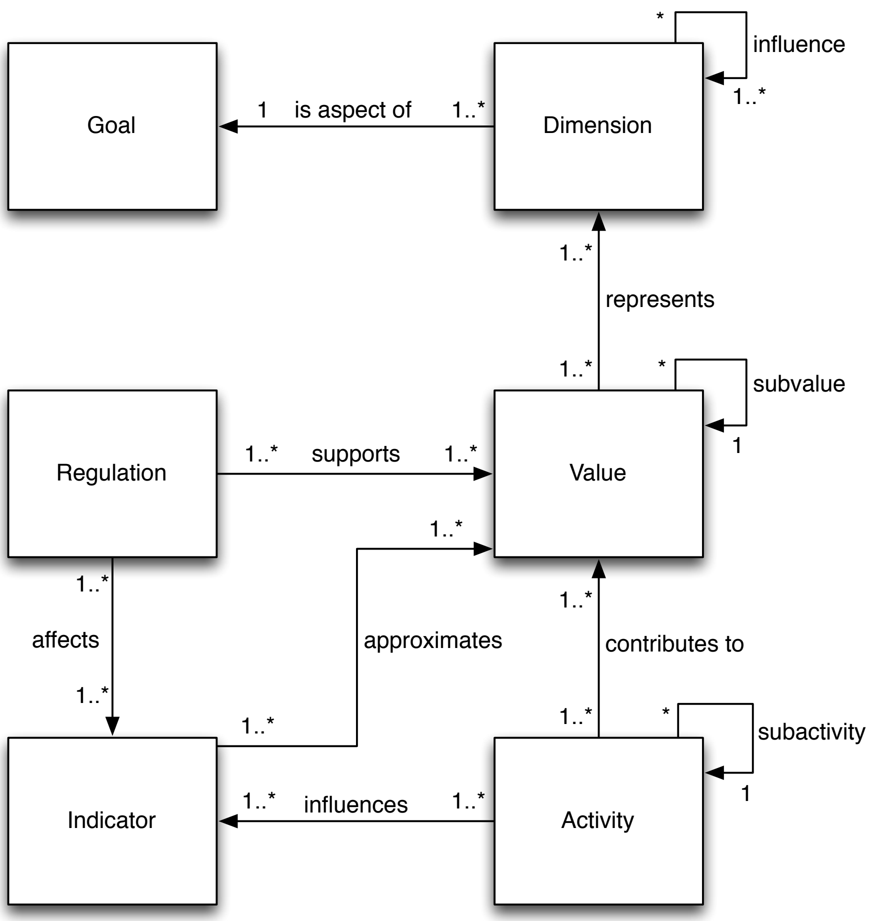

Figure 5.1: Meta model of generic sustainability model (M0).

Hypotheses The following hypotheses have been made for the meta model:

1. A dimension can be represented by a set of values.
2. A value can always be approximated by indicators. They may be quantitative or qualitative.
3. Qualitative and quantitative indicators cannot be aggregated to find an absolute measure for a value in case it is approximated by a set of indicators instead of one single value.

5.2 The Generic Sustainability Model

An excerpt of the generic sustainability model is provided in Fig. 5.2. The model consists of three levels: the top level contains the dimensions; the middle level contains values, indicators, and regulations; and the lower level contains activities. Each element in the generic sustainability model is an element of a type from the meta-model explained before. In the following we will explain examples from the generic model, structured by their type.

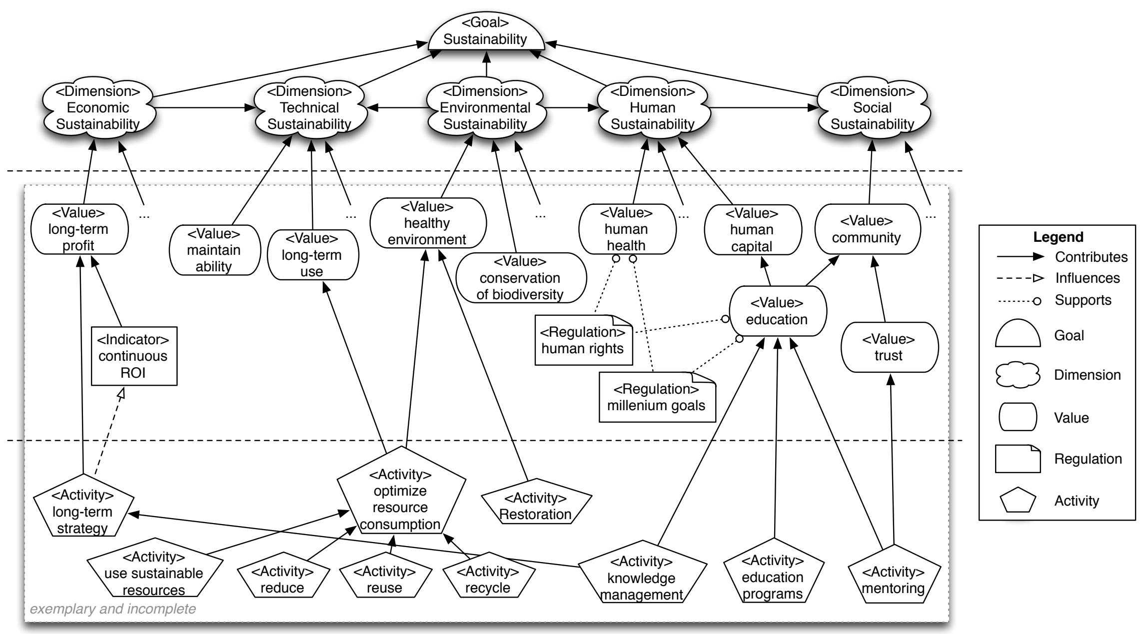

Figure 5.2: Exemplary excerpt of the generic sustainability model (M1).

5.2.1 Dimensions

A dimension is an aspect of or viewpoint on sustainability, for example, *environmental sustainability*. As described in Sec. 3.1, there are five dimensions to be considered when analyzing sustainability for software systems engineering. A dimension is detailed in a set of values.

5.2.2 Values

A value is a moral or natural good that is perceived as an expression of a specific dimension. Each of the five dimensions is represented by a set of values. Values do not necessarily belong exclusively to one dimension but can be considered for a number of dimensions, for example, *healthy environment*, which applies for both the environmental as well as the human dimension.

5.2.3 Indicator

An indicator is a qualitative or quantitative metric that expresses a specific degree or score with regard to a value, for example, *satisfaction indices* as qualitative metric and *carbon emissions* or *return on investment* as quantitative metrics. A set of indicators thereby approximates a value.

For the indicators, there are catalogues by, e.g., Bell and Morse [29] and the ESI [30].

5.2.4 Regulation

A regulation is an optional element that supports and/or enforces a value, for example, emission regulations. Regulations commonly set limits for a specific indicator to be of legal use.

Many values belonging to the different dimensions are heavily regulated, either supported or restricted in order to protect them. For example, freedom of the individual is supported by the *human rights*, and healthy air is supported by the *European Union's directive on carbon emissions.*

5.2.5 Activity

An activity is a measure taken to contribute to a specific value or a set of values, for example, *use train for mid-distance traveling instead of aircraft.* The impact of these activities on a value is measured by the indicators it influences. For the travel example, using a train instead of an aircraft improves the emissions account of the traveler. For each value, there is a number of activities that can be implemented to support a value. Thereby, the impact of an activity is measured by the indicators it influences.

Whenever we say "good" in activities, the degree can be measured by the indicators.

The generic sustainability model is intended to serve as reference and as a basis for the instantiation of company- or system-specific instances.

5.3 The Generic Sustainability Model Database

An encompassing generic sustainability model requires a database. We have decided for a simple database that is extended over time. Figures 5.3 - 5.7 illustrate excerpts of that database for each sustainability dimension.

5.3.1 Environmental Sustainability

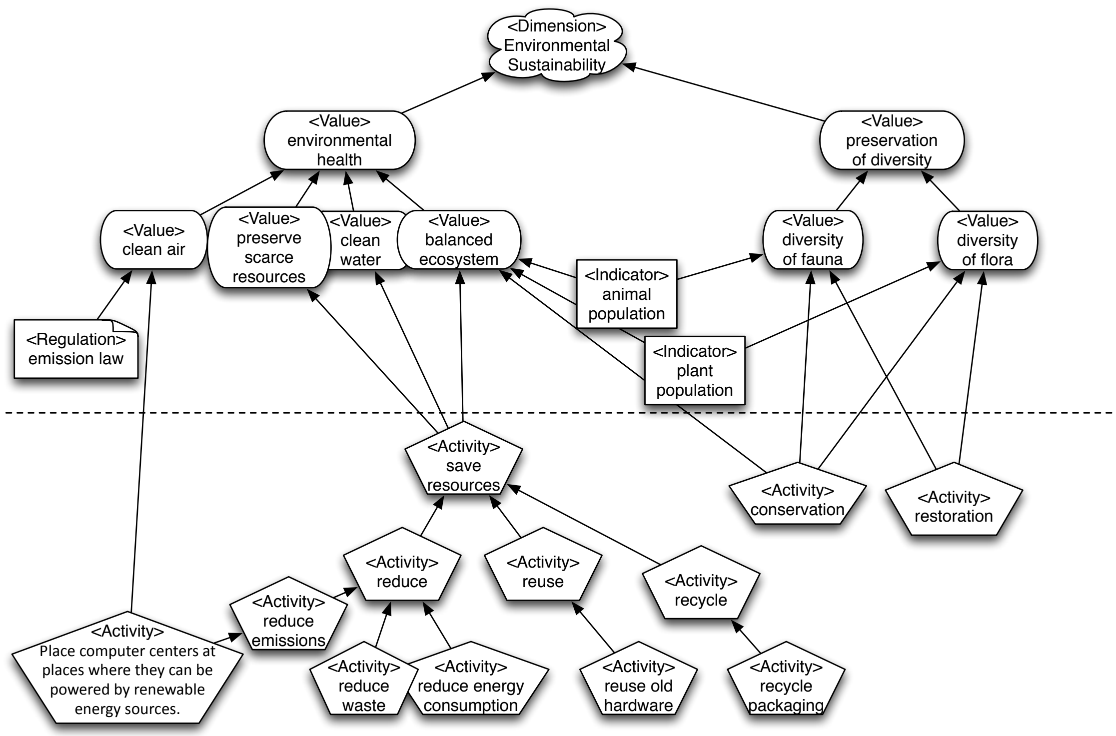

Figure 5.3: Environmental dimension of the sustainability model (M1).

The model excerpt for environmental sustainability is depicted in Figure 5.3. It is structured in the same way and with the same elements as Figure 5.2.

Below the dimension, there is a number of values that further decompose the values of *environmental health* and *preservation of diversity*. One of the sub-values, *preserve scarce resources* is realised by the activity *save resources*, which is again stepwise refined into the activity *reduce* and *reduce emissions*.

One option for realising feewr emissions is the activity *place computer centers where they can be powered by renewable energy sources.*, for example, to place data centers in islands that have a high geothermal activity, e.g., iceland. Powering data centers with geothermal energy on iceland reduces the production of emissions.

5.3.2 Human Sustainability

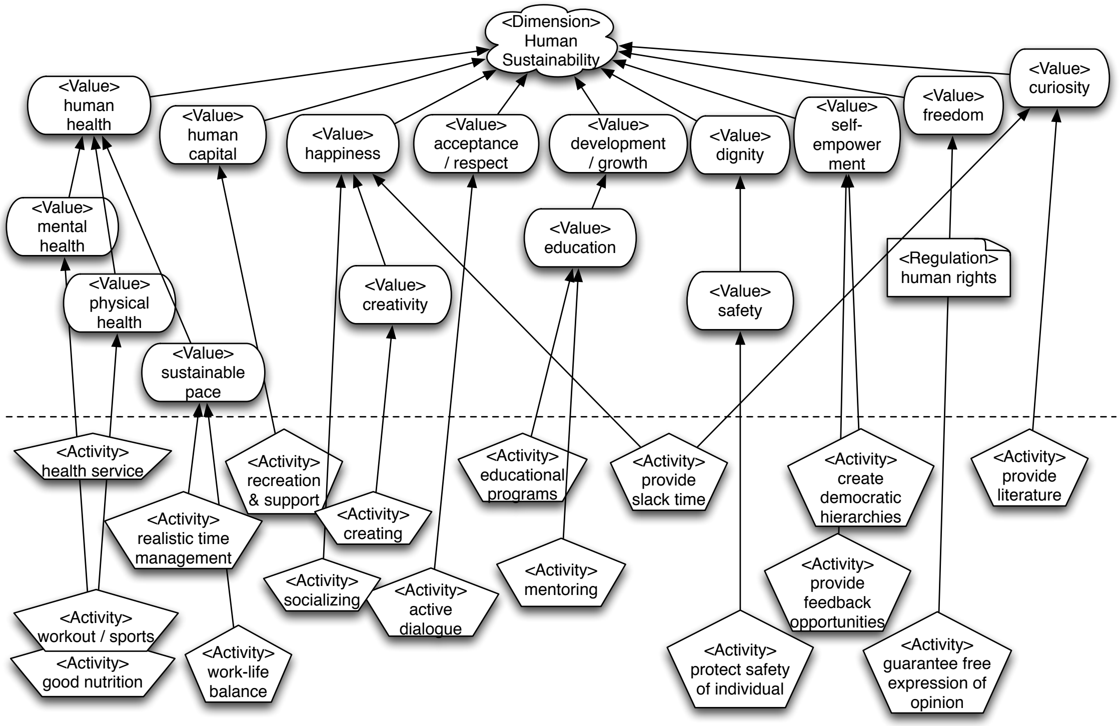

Figure 5.4: Human dimension of the sustainability model (M1).

The model excerpt for human sustainability is depicted in Figure 5.4. Here, one value is *happiness*, with the sub-value *creativity*.

The implementing activity *creating* can, for example when instantiating the model for a software company, be translated into the creativity in the developing processes of the software engineers. If they experience a certain degree of creative freedom in their work, this is likely to increase their happiness. This also has side effects on their productivity and thereby on the economic dimension, but this influence is not depicted in the figure.

5.3.3 Social Sustainability

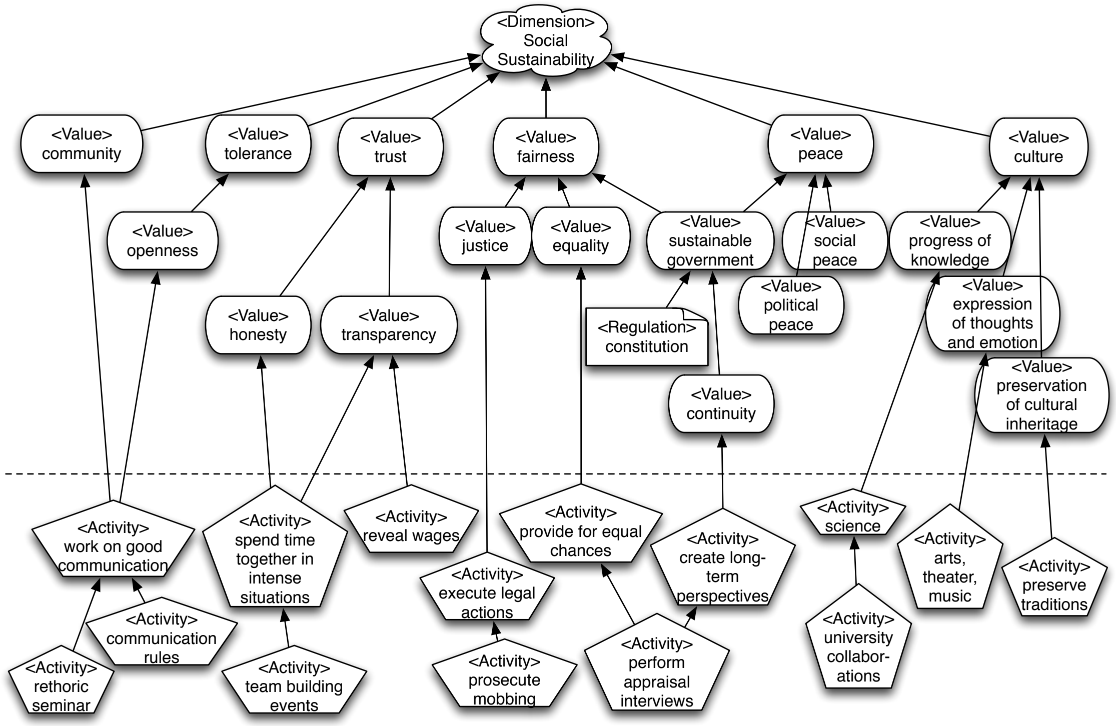

Figure 5.5: Social dimension of the sustainability model (M1).

Figure 5.5 depicts the excerpt for social sustainability. Some of the central values are *tolerance*, *trust*, *fairness*, and *culture*. Trust can, for example, be built up during team events where the members *spend time together in intense situations* as the respective activity suggests.

Tolerance requires *openness*, which can be supported by *work on good communication*. One action to improve communication in a software development company can be to set up rules for positive, open, and efficient communication.

Social sustainability in a company might best be supported via a company culture that builds on the above named values.

5.3.4 Economic Sustainability

Figure 5.6 depicts the excerpt for economic sustainability. Values are *security*, *fairness*, *safety*, *trust*, and *long-term profit*. The latter is realised by the activites *good infrastructure* and *long-term strategy*.

As depicted on the left of igure 5.6, such a *long-term strategy* can be supported by *investing in academia*, for example, in bilateral collaboration projects.

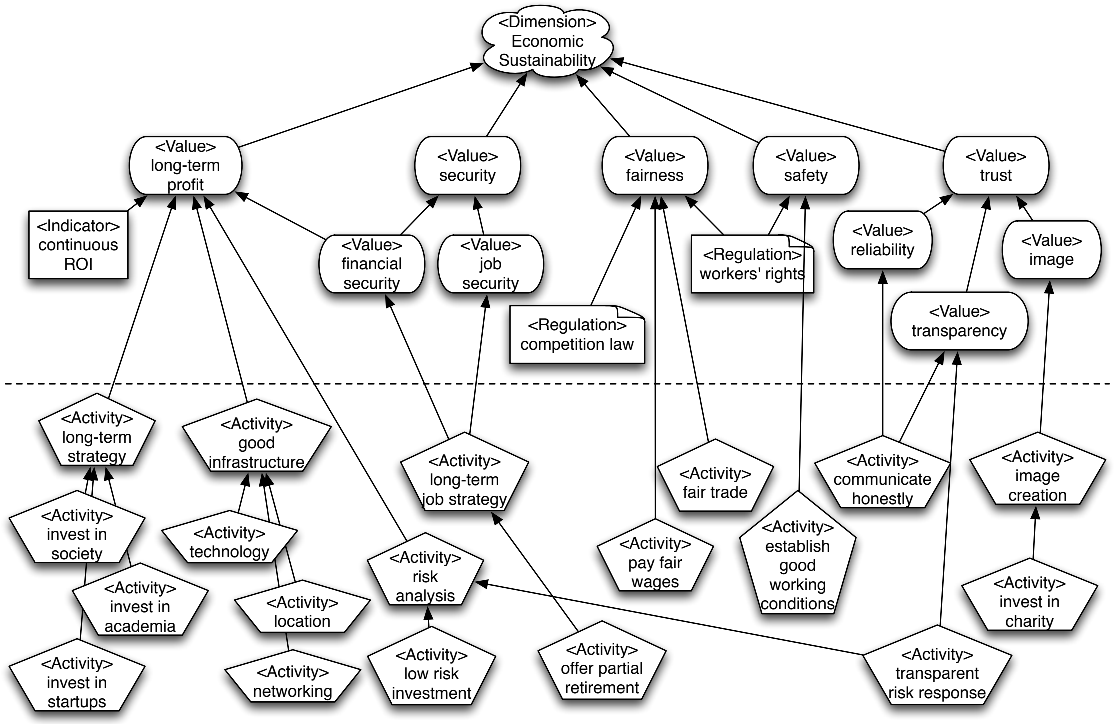

Figure 5.6: Economic dimension of the sustainability model (M1).

Another possibility that brings future prospects, but might also have a cross-influence on risk management, is *investing in startups*.

On the right hand side of Figure 5.6, the value of *trust* is decomposed into the subvalues *reliability*, *transparency*, and *image*, where the latter is realised via *image creation*, for example, be *investments in charity*.

5.3.5 Technical Sustainability

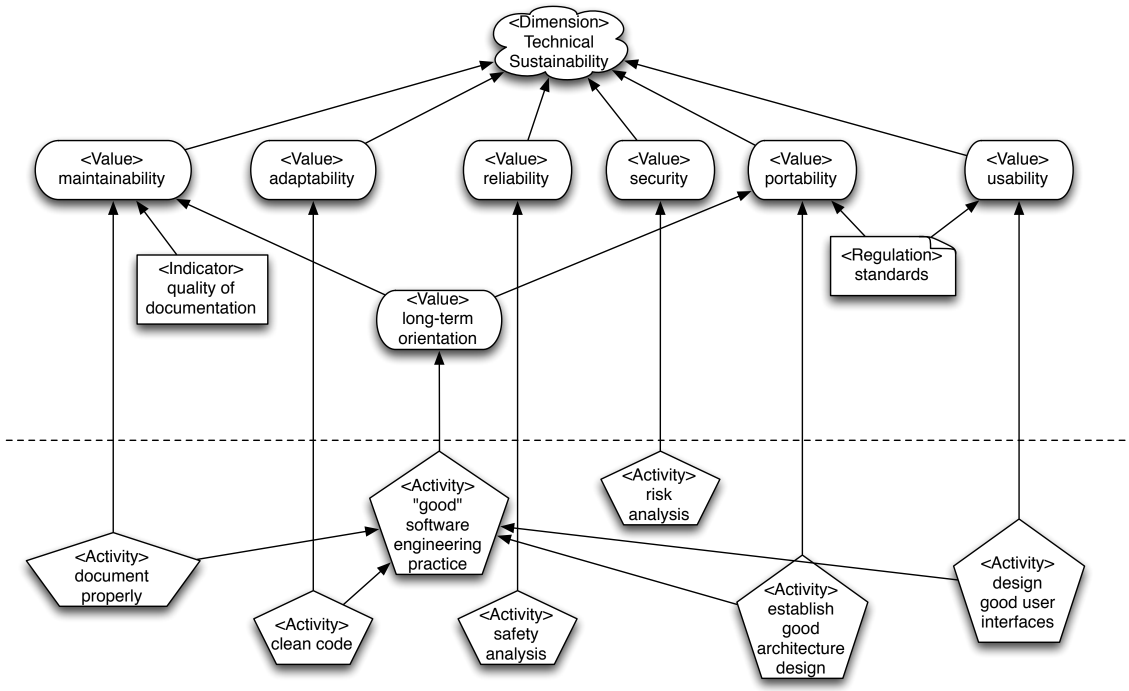

Figure 5.7: Technical dimension of the sustainability model (M1).

The excerpt for the technical dimension of sustainability is depicted in Figure 5.7. The central values are traditional quality characteristics, for example, *maintainability*, *adaptability*, *reliability*, *security*, *portability*, and *usability*.

Many of these values can be realised and supported by what might be called 'good software engineering', e.g., by writing *clean code*, performing *safety analyses*, establishing *good architecture design*, and *good documentation*. Thereby, good documentation is not the most extensive one, but a high quality documentation that serves the task of being able to recapitulate why decisions have been taken as they have.

Especially usability has become a crucial success factor in the domain of life-style devices like the range of hand-held devices during the past decade.

Therefore, the activity of *designing a good user interface* will strongly influence the technical sustainability of such a software system.

5.4 First Validation via Common Positions on Sustainability

A first indicator of whether the proposed generic model might actually be beneficial is by checking its ability to represent popoular or established positions or perspectives for sustainability.

Environmental sustainability is the conservation of biodiversity. This point of view is represented in the model by the `<Value>` of *conservation of biodiversity.*

Technical sustainability is long time usage of systems. This pragmatic definition is represented within the submodel in Figure 5.7 by the value *long-time orientation* and the supporting activities.

Technical sustainability is cheap maintenance of systems. This definition is a combination of the technical and the economic dimension and is represented within the model by these two dimensions.

Happy Planet Index The happy planet index [31] is a global measure of sustainable well-being that relates how much resources a country has to how happy its inhabitants are. This relates the social dimension to the economic one and can also be modeled in the proposed model.

Chapter 6

Process: Instantiation of the Generic Model

The generic model gives an idea of what sustainability means in general. However, the concrete ideas of sustainability in a given company or project highly depend on the values existent in this context. Therefore, it is inevitable to to evaluate, instantiate and maybe also extend these values with the stakeholders to create a common sustainability model for a company or a project. These models may then be used to create discussion for newly forming teams, which is especially important in cross-cultural projects (see for example [32] on issues in cross-cultural projects).

As the sustainability model highly depends on team members as well as project goals, it can be instantiated for development processes (companies) and for software systems (products). For the former, a sustainability goal model that represents the goals of the company is created; for the latter, this instantiation happens during the requirements engineering phase of each product. The method for developing the sustainability model, however, stays the same. In the following, we describe this method for developing a sustainability goal model first and give examples for both company-specific and product-specific instantiations afterwards.

The process consists of two phases: the analysis and the application & assessment phase. Various roles are partaking in developing a specific sustainability model within these phases.

6.1 Roles during Instantiation of the Generic Sustainabilty Model

During the analysis phase, the model is instantiated by the *Sustainability Architect*. This refers to the person, who knows most about the company's or project's ideas and goals. Usually this leads to the Business Analyst, the Requirements Engineer or the Process Engineer, depending on whether we create a project or company-specific instance.

For application we need two different roles: For one, the Sustainability Architect overviews the set of actions in total. Additionally each action needs an *Sustainability Activity Responsible*, who organizes and instruments one concrete activity.

Lastly, the assessment is performed by a *Quality Engineer*, who organizes evaluation and assessment of the indicators. He also reports to the *Management* regarding the status quo of the project or company. This life cycle is depicted in Fig. 6.1.

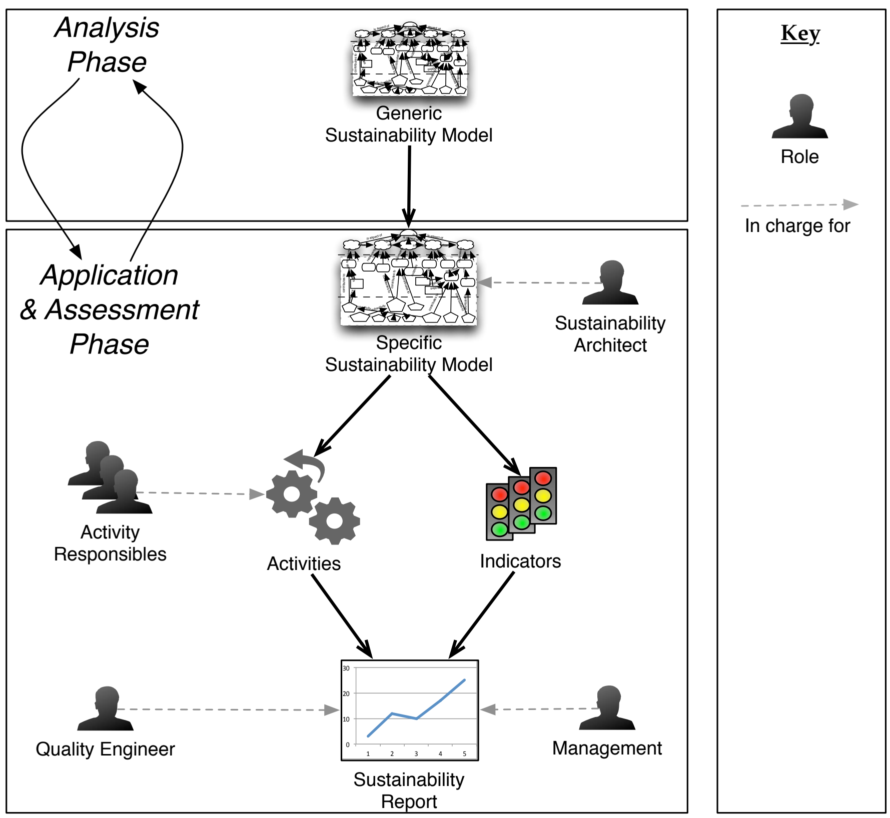

Figure 6.1: The life cycle of a specific goal model.

6.2 Modelling Process

Applying sustainability modeling within a company usually consists of two alternating phases: the analysis and the application & assessment phase.

Whereas the analysis phase mainly concretizes the definition of sustainability in the present context, the application & assessment phase starts the selected activities and supervises whether the activities are working successfully. Even though these are two separated phases we expect these phases to be alternating, thus refining and tuning the specific instance of the goal model iteratively over time.

During the whole process the model guides the tailoring of values, indicators and activities to the context (company- or project-specific) by structuring the goals and giving examples and suggestions.

6.2.1 Analysis Phase

In the analysis phase the sustainability architect tailors the generic goal model to the context (a specific company or software system). This formalizes what sustainability really means for a certain company or product. As depicted in Fig. 6.2, the analysis phase includes three steps:

1. For each dimensions the architect needs to instantiate the generic values that are provided by the generic goal model into context-specific goals. These goals need to be prioritized to help solving conflicts between potentially contradicting goals.

2. Afterwards the architect defines activities to implement the goals and defines indicators that make it possible to assess the state of this goal at the moment as well as in future situations. In this step he might find trade-offs between indicators that are more exact with respect to this value and indicators that are easier to apply. For example, the architect might have to decide whether we want to measure our resource consumption by weight or by item number (which can probably be automated). The generic goal model serves as a reference for selecting activities and indicators.

3. When values and indicators for a certain dimension are selected, the architect can now relate the activities that have a positive impact on the indicators. For example, he could find reusable items in this project and set up policies that enforce this reuse.

Indicators and activities are usually developed iteratively. The engineer can choose whether it is more adequate for them to start with defining activities or to list the most important indicators for a specific sustainability assessment.

This way we instantiated the generic model with goals, indicators and activities specific to a company or a product.

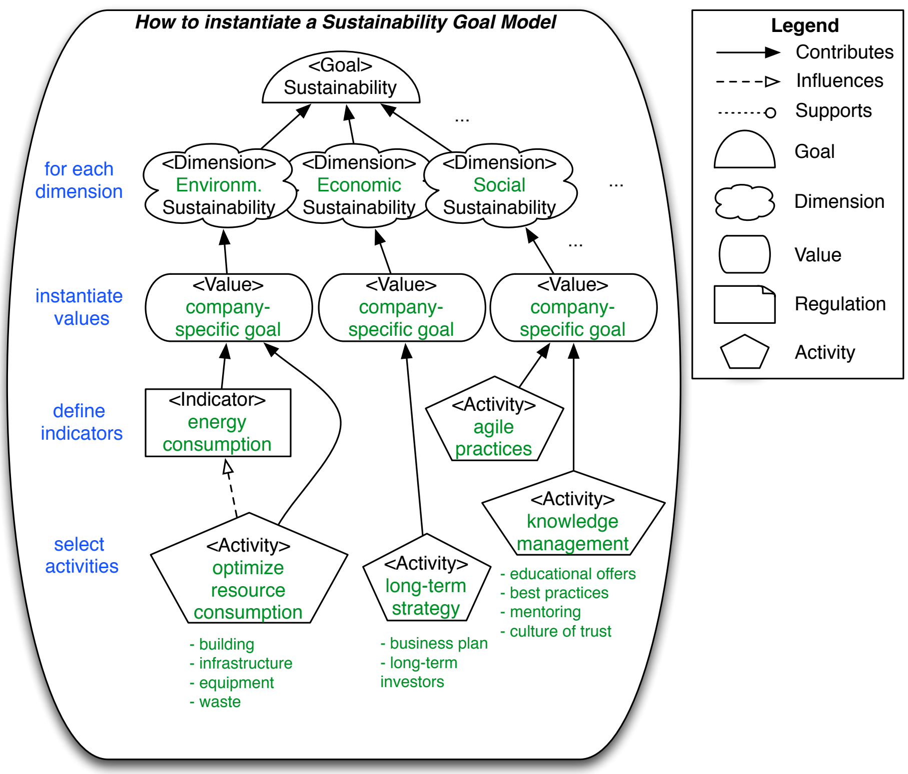

Figure 6.2: How to instantiate the generic model.

6.2.2 Application & Assessment Phase

After formalizing the company's or product's specific sustainability goal model, two parallel tasks are needed: First of all, each activity needs a Sustainability Activity Responsible who is in charge for this concrete activity. Together, the sustainability architect and the activity responsible's support and enforce the implementation of the previously selected activities. Second, the Quality Engineer needs to continuously monitor the company's state with respect to its sustainability model. He can do this by assessing the list of indicators that are defined concretely in the model. The results of this assessment are reported to the management and the sustainability architect, so that changes or adjustments can be made accordingly. The iteration time between assessments should be as short as possible to maximize the transparency of the current state. However, it heavily depends on the automation potential of the company-specific indicators and the investment the company or project is willing to make.

In summary, the main goal of the application & assessment phase is putting activities into play and monitoring how the company or product is performing with regards to its own definition of sustainability.

6.3 Example: Scenario Smart Mobility

Our first example for the instantiation of a goal model is one that was developed within the publicly funded research project ARAMiS: Automotive, Railway, and Avionics in Multicore Systems [33, 34], which build on the Agenda CPS, a study by the German academy of technical sciences (acatech) on the perspectives in CPS research, development, and application [35].

ARAMiS Project Background One class of systems of systems with the additional challenge of integrating different system types are cyber-physical systems (CPS). CPS are integrations of computation and physical processes. Lee [36] defines CPS' as embedded computers and networks that monitor and control physical processes, usually with feedback loops where physical processes affect computations and vice versa. This leads to complex functionality that spans a variety of application domains.

The projects duration is 3 years, it has a total budget of 36,5 mio Euro, and it is partially funded by the German federal ministry of education and research.

Partners are inter alia KIT, AUDI, BMW, Bosch, Continental, Daimler, Airbus, EADS, Diehl, Liebherr, Freescale, Infineon, Intel, SYSGO, Vector, Wind River, FHG IESE and AISEC, OFFIS e. V., TU Braunschweig, TU Kaiserslautern, TU München, Universität Paderborn, Universität Stuttgart, and ForTISS GmbH.

The main goal of ARAMIS is to provide for the technological basis for improving safety, efficiency, and comfort in the mobility domains of automotive, railway, and avionics by using multicore technology. The insights gained in the project build the indispensable foundation for the successful integration of embedded systems to cyber-physical systems.

The project is structured in 6 sub-projects: scenarios and requirements, continuous development method, system design, hardware, software, and demonstrators. Results that are common to all application domains are captured in the so-called Domain Common. Our research group has the academic sub-project lead in “scenarios and requirements” together with AUDI as industrial lead.

Within that subproject, sustainability was identified as a cross-cutting concern we needed to analyse.

Smart Mobility Scenario As there is a large conjoint of partners, we chose to analyse and model for a product rather than for a company.

We first decided on which of the cyber physical system scenarios to analyse sustainability for. The smart mobility scenario was selected. It is a contrived future mobility scenario that combines the automotive, the railway and the avionics domains. The scenario describes the journey of a family from Munich to Oslo and chiefly highlights the enhanced comfort and support from the viewpoint of the end users. With the help of this scenario, the reader becomes aware of the reach and sophistication of the systems envisioned in the project ARAMiS.

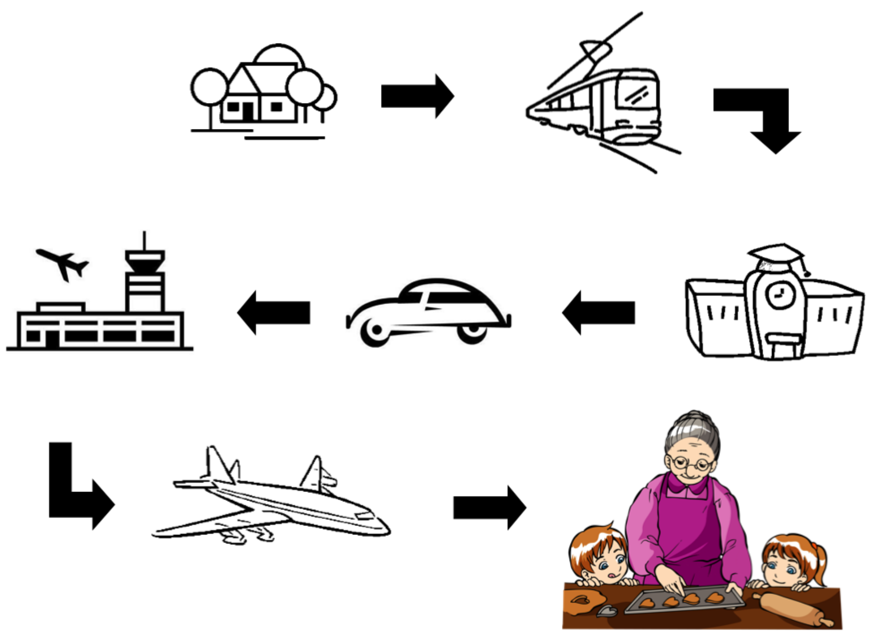

Figure 6.3: Smart Mobility Scenario Overview

The scenario's story is depicted in Figure 6.3. Ms Rosemarie Weber plans to spend the next Christmas break with her two children at her mother's, Ms Pauline Mayer. The Weber family lives in Munich, Ms Mayer lives in Sandvika near Oslo. Ms Weber's intention is to pick up her children from school and from there to travel directly to her mother. In the scenario Ms Weber plan is worked out. Ms Weber enters departure time as well as from and to locations, a maximum cost amount for the entire route as well as passengers' names in her smart device. The mobile device is connected to various providers and to Ms Weber's private cloud, and makes her suggestions for the trip from her home to pick up her children at their school in the city centre and onward to her mother. Ms Weber decides to use public transportation to the school, from there to continue with her children to the airport with a hybrid car-sharing vehicle (CSV) with autonomous driving capabilities, and finally to reach Oslo by plane, as this is the most energy-efficient and least expensive alternative. The necessary travel documents such as public transport ticket, CSV authorization and flight ticket are transmitted to the mobile device of Ms Weber. Further profile details are automatically taken into account; e.g., preferences concerning meals on board are directly transmitted from Ms Weber's private cloud to the airline.

Analysis Phase In the analysis phase (see Sec. 6.2.1), we started with the dimension of environmental sustainability, as depicted in Figure 6.4.

The goal instantiated for the values of *preserve scarce resources* and *clean air* (cf. Fig. 5.3) is *reduce resource consumption.*

There are four major indicators that can be taken into account for assessing the effectiveness of activities: *energy footprint*, *carbon emissions*, *bought items list*, and *physical waste*.

Activities to achieve the goal of reducing resource consumption in the smart mobility service are, for example, to *make owning cars superfluous for the service*, to *offer older vehicles for lower rates*, to *include car sharing into the business model*, and to *minimize the energy consumption of the participating mobility devices.*

Application and Assessment Phase In the application and assessment phase, the indicators are measured to capture the current state of the situation. Subsequently, the activities are implemented and then the indicators are reassessed.

For the ARAMiS project, however, the *smart mobility service* is only one of several vision scenarios that form the rationale for the development of demonstrator systems while the actual overall service is still a future (and futuristic) business model. Therefore, we cannot provide details on the application phase of this instance.

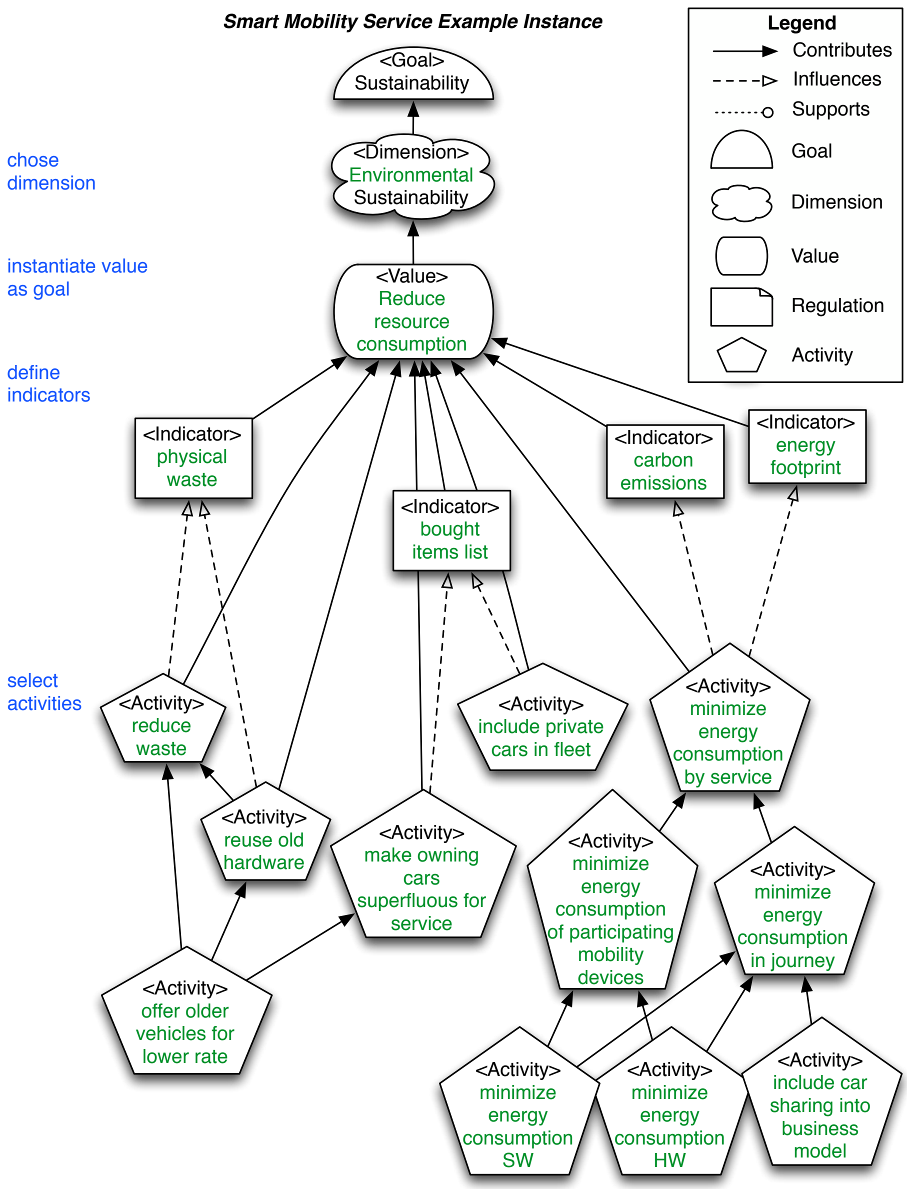

Figure 6.4: Smart Mobility Service Example Instance

Chapter 7

Application: Instances of the Sustainability Model

This sections describe example instantiations of the generic goal model. We will see examples for stakeholder analysis as described in Chapter 4 as well as Product- and Company-specific instances as described in Chapter 5 and 6.

7.1 Product-Specific Instance @BMW

The first instantion took place at BMW Munich for a car sharing plattform. With DriveNow[1], the german car manifacturer BMW created the following business model: Instead of buying a car, drivers can get a membership for the program and pick up a DriveNow car whenever they need it. The cars are maintained and refilled by the manifacturer. In consequence, the driver has no fixed costs, but pays per time using the car. The program has been established in Munich, Berlin, Duesseldorf, Cologne and San Francisco. In this instantiation we will focus on the car sharing plattform for Munich.

In the following, we will first conduct a stakeholder analysis and afterwards go into the instantiation of the sustainability model.

7.1.1 Car Sharing Platform: Stakeholders

For the concrete instance of a car sharing plattform we can instantiate the general stakeholder list from Chapter 4. Each stakeholder has a certain view on the project and has his own goals. It is important for the success of the project to be aware of these goals [7].

For DriveNow we identified the following stakeholders (see Fig. 7.1):

- CEO of BMW: The CEO makes sure that the sustainability model is in accordance with the company's general goals. As such his goals are the

[1] `http://tinyurl.com/9fc8yl3`

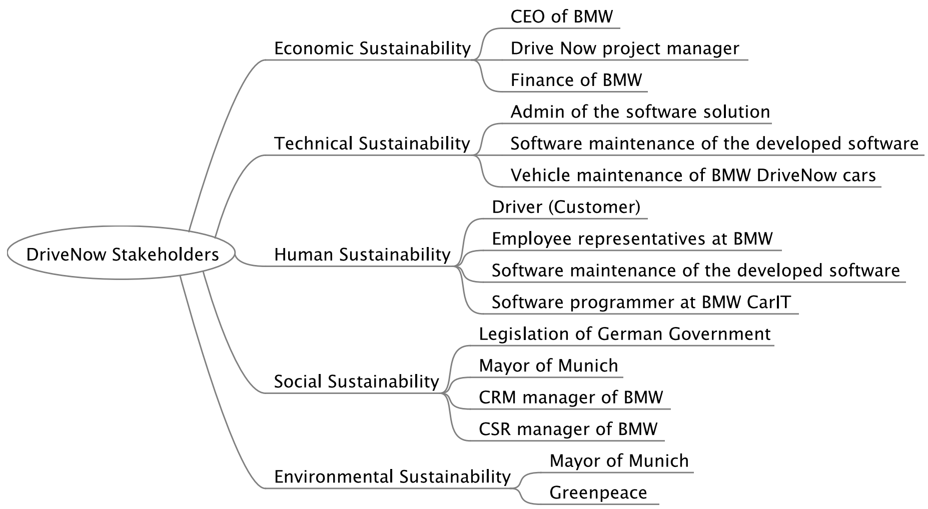

Figure 7.1: Sustainability stakeholders of the DriveNow projects.

following:

  - Survive, succeed, prosper of company ([8])
  - Financial success
  - Positive image for company
  - Customer satisfaction

- DriveNow Project Manager: The project manager's primary focus is the project itself. His goals are consequently:
  - Survive, succeed, prosper of project
  - Visibility of project
  - Big user community
- Finance of BMW: The BMW finanzing section is interested in the finanzial benefits of the project. Their goals are:
  - Financial success
  - Correct accounting
- Admin of the Software Solution: The administrators needs to watch the running software solution over the next few years. Hence, they care for a long-lasting infrastructure.
- Software maintenance of the Developed Software: The software maintenance wants to be able to change the car sharing software and fix bugs that appear during its life time. Their goals are:
  - Automated registration and management
  - As little effort as possible

- Vehicle maintenance of BMW Drive Now Cars: The people who repair the DriveNow cars are interested in having long-living vehicles with low maintenance at the vehicle. This means:
  - Little abrasion
  - Little mending
- Employee representatives of BMW: The employee representatives are elected to represent the workers rights. Consequently their goals include:
  - Good working conditions
  - Safety
- Driver (Customer): The customer is the person that pays for the DriveNow service. In order to use the service, he expects:
  - Easily understandable user interface
  - Quick registration and simple billing
  - Economic mobility services
- Software programmer at BMW CarIT (Developer): The programmer needs to create the software. Inter alia his goals are:
  - Good interface definitions
  - Successful development
  - Sleek code ;)
- Legislation of German Government: The german government is in charge for the traffic in their cities. They represent the society and the interests of the country as a whole. Among others, they focus on:
  - Safety
  - Minimal environmental impact
- CRM: The CRM wants to establish long-term relationships with their customers. The CRM's goals are:
  - Positive image for company
  - Customer satisfaction
  - Long-term profitable customers
- Mayor of Munich (Community representative): The Mayor is a representative of the inhabitants of Munich. He focusses on the goals of the community in Munich.
  - Beautiful city
  - Traffic reduction
  - Good reputation
  - Increased percentage of public transportation
- CSR manager: The CSR manager
  - Positive image with regard to environmental impact and conservation efforts

- Greenpeace (Nature conservation activists): Because nature does not have a representative to speak out for it, lobbyists and activists from Greenpeace and similar organizations act as their representatives. Hence, their goals include:
  - Conservation of nature
  - Reduction of emissions and waste

This stakeholder analysis is an example of a helpful starting point for a proper analysis of the sustainability dimensions. Together with the stakeholders we can now build a project-specific instance of the sustainability model.

7.1.2 Instantiation of the Sustainability Model

We created a product-specific instance of the generic sustainability model for DriveNow. An excerpt is depicted in Fig. 7.2. According to the method described in Chapter 6, we first instantiated the generic model in a product-specific goal model in the analysis phase. This works like the following: For environmental sustainability the value *reduce environmental impact* is chosen. We can select similar indicators from the generic goal model as in the company-specific instantiation, such as the *energy bill*. Yet, we need to tailor our activities specifically to this product: For example, in this system the usage of *green data centers* is advised. Furthermore, the business process can be varied by offering *older cars for a lower rental rate* (instead of deposing them) and thereby reducing waste, and by introducing *cars with hydro or electric power*.

Again, in the application & assessment phase, we need to make sure that these activities are followed during the development of a product and measure the impact of activities.

7.2 Company-Specific Instance @jambit

We also developed a company-specific instance the generic sustainability model, of which an excerpt is depicted in Fig. 7.3. For environmental sustainability we instantiate the value *Reduce resource consumption* into the goal *Reduce resources consumption by 30% within 12 month*. We find three indicators for resource consumption: *physical waste*, the company's *energy bill* and the size of the *bought items list*. We can immediately see differences in how easy one can measure these indicators. From these indicators we can select activities that have an influence on the indicators. For example to reduce the size of the bought items list, we can place *incentives for less resource consumption*. All values, indicators and activities can be refined within the model. For example, the unspecific activity *reduce waste* is refined within the subactivity *recycle packaging for own shipping*.

In the application & assessment phase we would now need to establish the activities and measure our indicators. From this information we can deduce

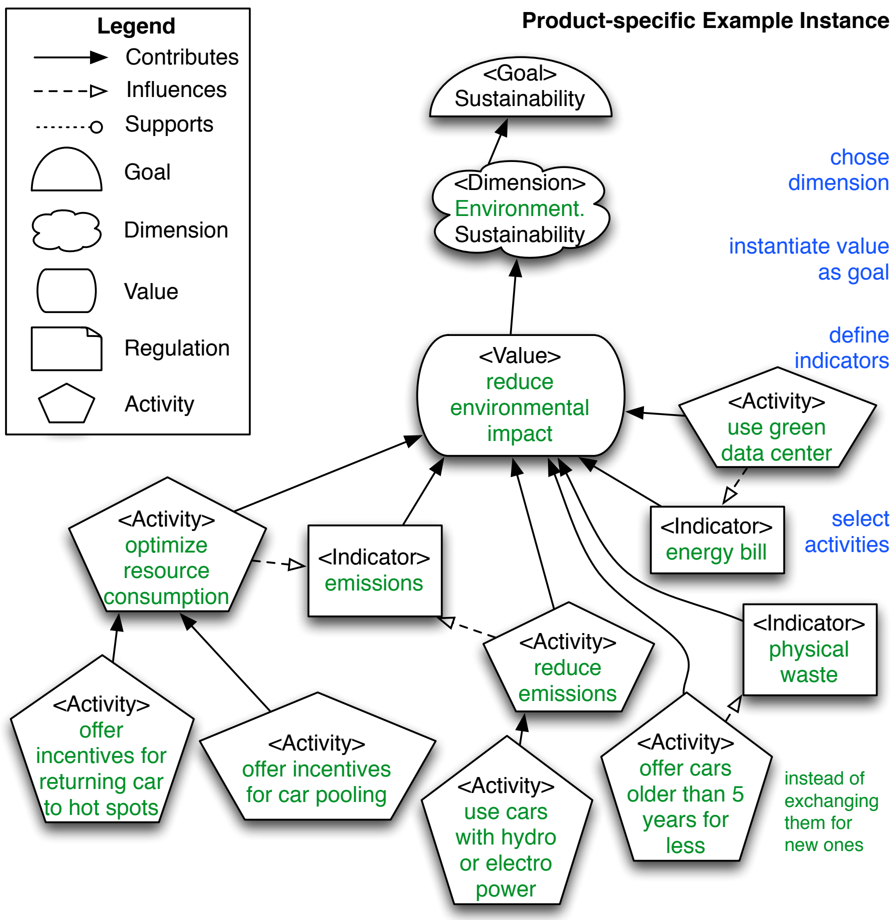

Figure 7.2: Example instances of the generic sustainability model for the product DriveNow (M2).

how well our activities are performing and whether or not we work towards the company-specific sustainability goals.

The first industrial case study for a company-specific instance is currently under development in a small software development company (jambit GmbH[2], 50 developers) in a series of workshops that are accompanied by a Master's thesis.[3]

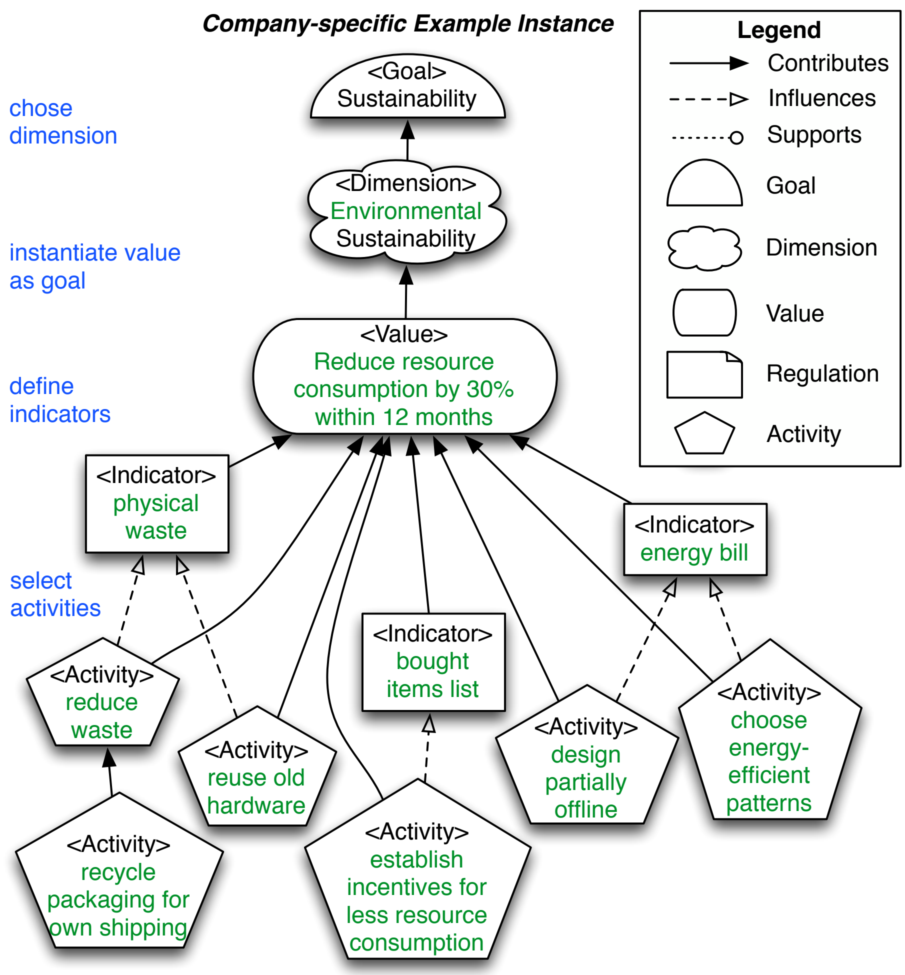

Figure 7.3: Example instance of the generic sustainability model for a company (M2).

[2] http://www.jambit.com

[3] http://tinyurl.com/9xdpqgc

7.3 Requirements Engineering Conference 2013

The model depicted in Figure 7.4 - 7.8 is used by the organizing committee of the International Requirements Engineering Conference 2013.

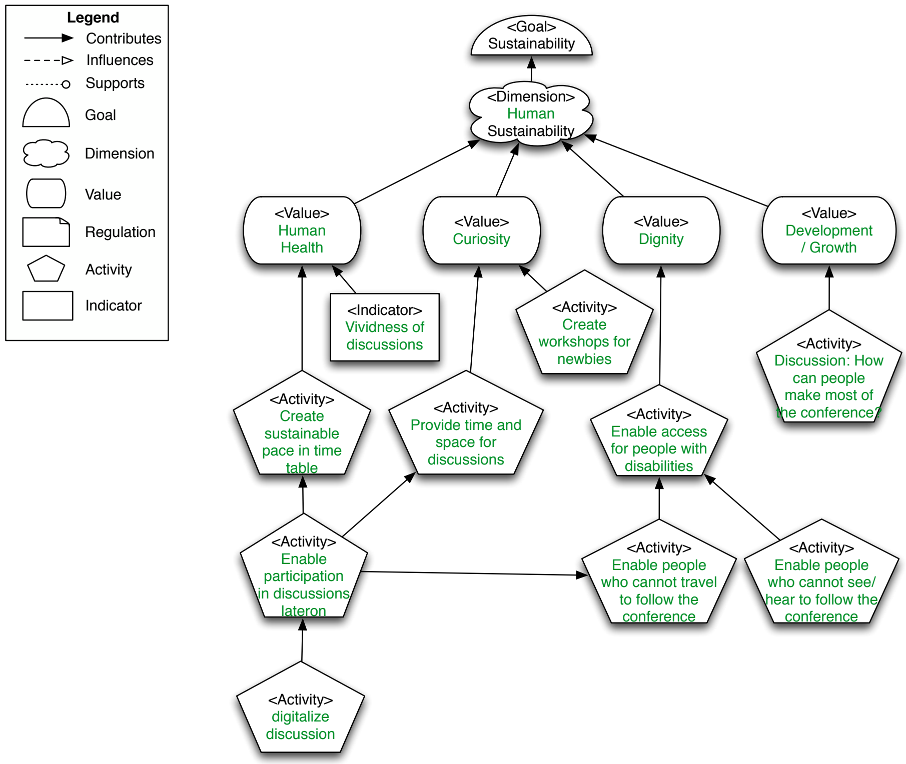

Figure 7.4: Example instance for the human dimension of the Requirements Engineering Conference (M2).

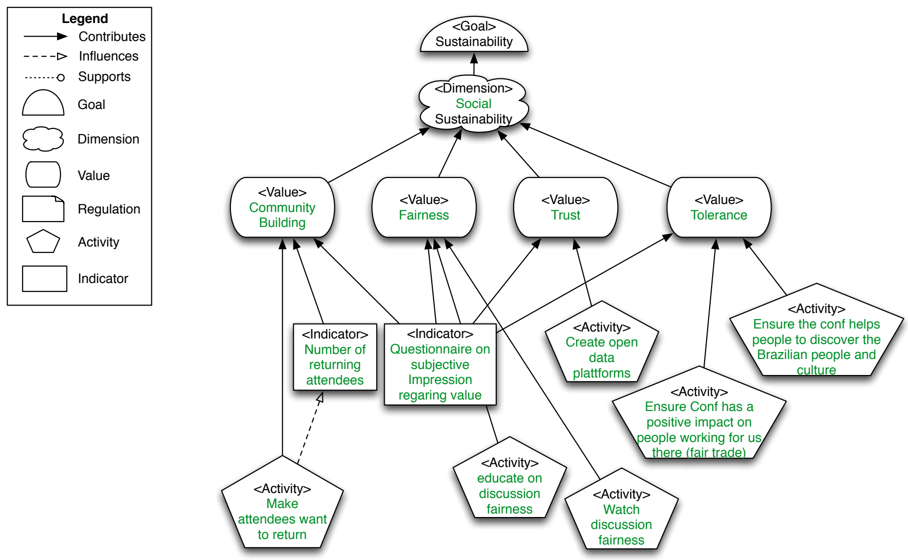

Figure 7.5: Example instance for the social dimension of the Requirements Engineering Conference (M2).

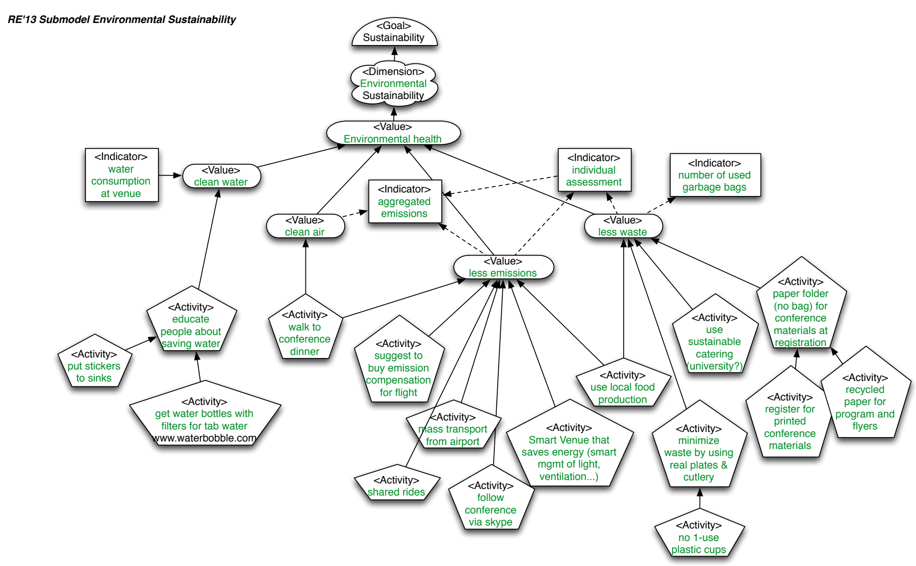

Figure 7.6: Example instance for the environmental dimension of the Requirements Engineering Conference (M2).

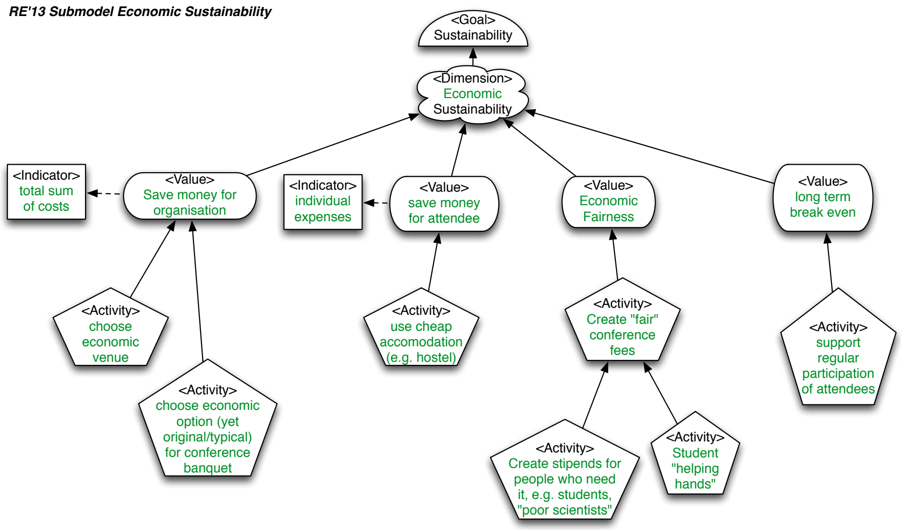

Figure 7.7: Example instance for the economic dimension of the Requirements Engineering Conference (M2).

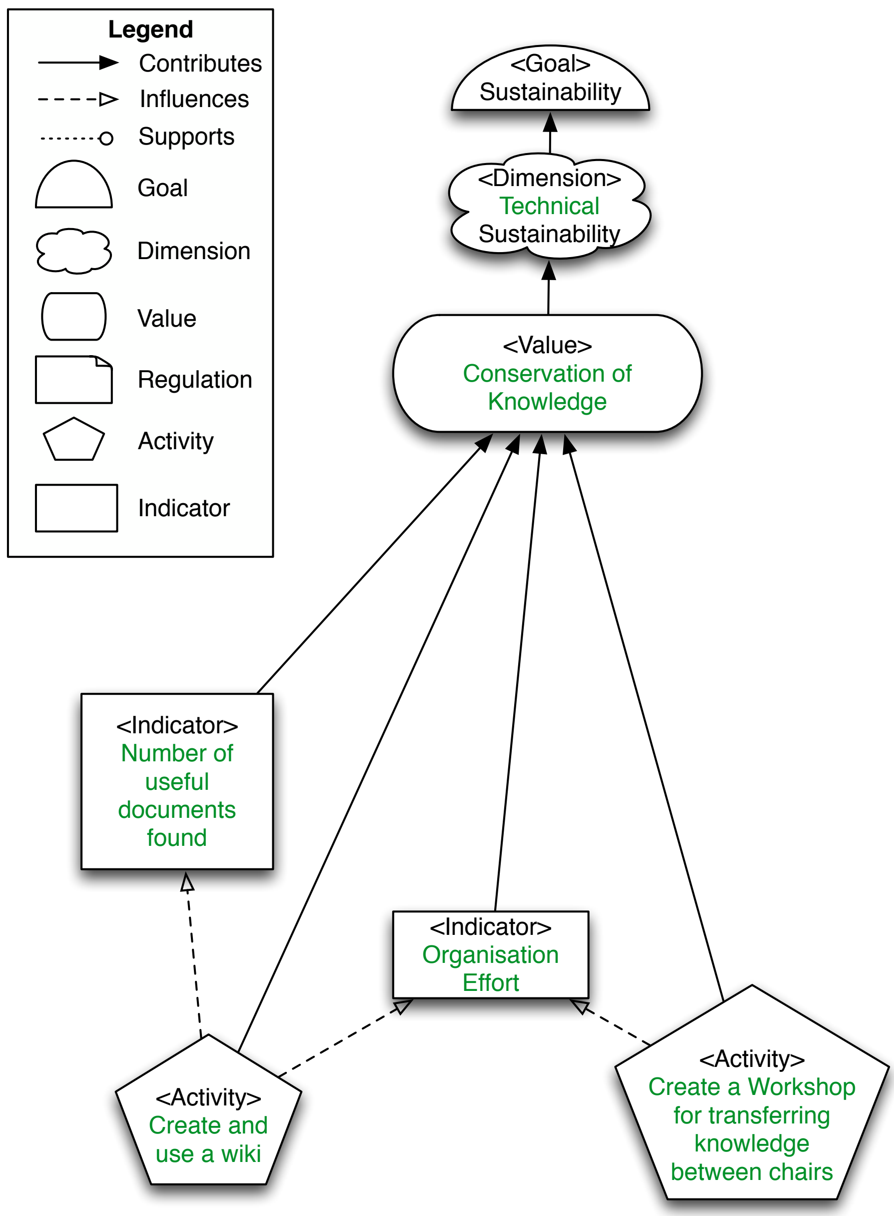

Figure 7.8: Example instance for the technical dimension of the Requirements Engineering Conference (M2).

7.4 Assessment: Evaluation and Discussion

The previous examples already show both benefits as well as problems and future work for modeling with the generic sustainability model.

7.4.1 Benefits

First, we can see the purpose of the sustainability goal model: Many elements of the model (such as the *physical waste* indicator) reappear in various instances. Hence, we can imagine a process for modeling sustainability where much of the modeling can be reused and only needs to be selected.
Second, we can directly see during the modeling how sustainability, a rather abstract concept, is turned into a concrete and measurable property of a company or a product. By making these implicit definitions explicit via the model, it is possible to discuss and evaluate a company's or product's impact on sustainability.

7.4.2 Issues

While creating the instances we were faced with the question of how to measure the impact of the proposed improvement activities for a product that does not yet exist. For example, when taking a product design decision such as using green data centers, there is no "before" that could be compared to an "after" situation as the system does not exist yet. Consequently, we can only compare to "standard design", but this is a less convincing alternative.
In that case, the model can still serve well in a constructive way for decomposing sustainability goals and selecting activities to realise them, but the analytic phase after implementation reveals less concrete results.

7.4.3 Threats to Validity

One threat to validity arises from our constructive research approach for the model. The risk is limited by our experience in developing such models, by reviews from and discussion with other researchers, and by feedback from industrial partners.
The other threat is that the evaluation in practice is still work in progress which might lead to further insights and changes in the methodic aspects of the approach.
Both threats will be addressed by industry collaborations in future work.

Chapter 8

Conclusion

We propose a generic sustainability model with process- and product-specific instances that can help requirements engineers to analyse their projects according to the different dimensions of sustainability and choose actions for improvement.

Special emphasis was put on environmental sustainability in and via software systems, as this dimension of sustainability is the one that is least supported by our traditional ways of developing software systems.

8.1 Results

The approach we suggest consists of a meta-model, a generic sustainability model and an approach on how to develop company- and project-specific sustainability models. In oder to explain how to repeat this instantiation we give examples for sustainability modelling for BMW Drive Now, for a Munich-based company and for the upcomming RE2013 conference.

When instantiating the sustainability models we realized the high potential for reuse. We understood that large parts can be reused, which argues for creating a reusable sustainability model. Nevertheless, various decisions towards sustainable development of systems and sustainable companies have to respect the specialities of these projects and companies. Hence, the sustainability model must be tailored to solve the conflicts between the different goals of the stakeholders. The resulting sustainability models are still very scarce but already serve as a good basis for discussion of goals, actions and performance.

8.2 Future Work

One important issue for future work is how to treat conflicting goals. On first sight, conflicts can be identified by contradicting influences on same indicators. For example, the ever-present conflict between a tight budget and quality improvement measures is also found in the sustainability goals and activities (i.e.

in indicators for economical vs. indicators for technical sustainability). However, more challenging is the question whether there are conflicts that are not as obvious and how to track them down if they do not relate to same indicators in the first place.

The other major issue is to understand and quantify the reuse potential of instances (how systems-specific are they really?) and the gradual extension of the reference model by knowledge gained from instantiating the model for processes and products in various companies. We expect continuous improvement by evolving the model over time in the ongoing cooperation with our industrial partners.

Acknowledgments

We would like to thank Susanne Klein, Oliver Feldmann, and Jonas Eckhardt for helpful comments on drafts of this paper as well as Debra Richardson and Bill Tomlinson for feedback on early ideas.

ToDo: include further reviewers

Bibliography

[1] Hilty, L.M., Arnfalk, P., Erdmann, L., Goodman, J., Lehmann, M., Wager, P.A.: The relevance of information and communication technologies for environmental sustainability — a prospective simulation study. Environmental Modelling & Software **21** (2006) 1618 – 1629

[2] Naumann, S., Dick, M., Kern, E., Johann, T.: The greensoft model: A reference model for green and sustainable software and its engineering. Sustainable Computing: Informatics and Systems (2011) –

[3] Gu, Q., Lago, P., Potenza, S.: Aligning economic impact with environmental benefits: A green strategy model. In: First International Workshop on Green and Sustainable Software. (2012)

[4] Cabot, J., Easterbrook, S., Horkoff, J., Lessard, L., Liaskos, S., Mazon, J.N.: Integrating sustainability in decision-making processes: A modelling strategy. In: Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on. (2009) 207 –210

[5] Stefan, D., Barrett, M., Letier, E., Stella-Sawicki, M.: Goal-oriented system modelling for managing environmental sustainability. In: International Workshop on Software Research and Climate Change (WSRCC). (2010)

[6] Mahaux, M., Heymans, P., Saval, G.: Discovering Sustainability Requirements: an Experience Report. In: 17th REFSQ. (2011)

[7] van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: 5th IEEE Intl. Symposium on Requirements Engineering. (2001) 249

[8] Kaplan, R.S., Norton, D.P.: The balanced scorecard: measures that drive performance. Harvard Business Review **83** (2005) 172–80 7.

[9] Dobson, A.: Environment sustainabilities: An analysis and a typology. Environmental Politics **5** (1996) 401–428

[10] Burger, P., Christen, M.: Towards a capability approach of sustainability. Journal of Cleaner Production **19** (2001) 787–795

[11] United Nations World Commission on Environment and Development: Report: Our Common Future. In: United Nations Conference on Environment and Development. (1987)

[12] Robert, K.H., Schmidt-Bleek, B., de Larderel, J.A., Basile, G., Jansen, J., Kuehr, R., Thomas, P.P., Suzuki, M., Hawken, P., Wackernagel, M.: Strategic sustainable development — selection, design and synergies of applied tools. Journal of Cleaner Production **10** (2002) 197–214

[13] Organization, I.S.: ISO 14001 Environmental Management System Standard (2009)

[14] Bauman, H., Tillman, A.M.: The Hitch Hiker's Guide to LCA: An Orientation in Life Cycle Assessment Methodology and Applications. Professional Pub Serv (2004)

[15] Rees, W.E.: Ecological footprints and appropriated carrying capacity: what urban economics leaves out. Environment and Urbanization **4** (1992) 121–130

[16] Christen, M., Schmidt, S.: A formal framework for conceptions of sustainability — a theoretical contribution to the discourse in sustainable development. Sustainable Development (2011) DOI: 10.1002/sd.518

[17] Penzenstadler, B., Bauer, V., Calero, C., Franch, X.: Sustainability in Software Engineering: A Systematic Literature Review for Building up a Knowledge Base. In: Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering. (2012)

[18] Meadows, D.H.: Thinking in Systems - A primer. Earthscan (2008)

[19] Hilty, L., Lohmann, W., Huang, E.: Sustainability and ICT — an overview of the field. In: Proceedings of the EnviroInfo 2011. (2011)

[20] Jevons, W.S.: The coal question. 2nd edn. Macmillan and Co. London (1866)

[21] Group, T.C.: Smart 2020: Enabling the low carbon economy in the information age. Technical report, Global eSustainability Initiative (2008)

[22] Komiyama, H., Kraines, S.: Vision 2050 - Roadmap for a Sustainable Earth. Springer (2008)

[23] Goodland, R.: Sustainability: Human, Social, Economic and Environmental. In: Encyclopedia of Global Environmental Change. John Wiley and Sons (2002)

[24] UN World Comission on Environment and Development: Our Common Future. In: UN Conference on Environment and Development. (1987)

[25] Lago, P., Kazman, R., Meyer, N., Morisio, M., Müller, H., Paulisch, F., Scanniello, G., Zimmermann, O.: First international workshop on green and sustainable software at icse (2012)

[26] Zimmerman, M.J.: Plato's intrinsic vs. extrinsic value. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Winter 2010 edn. http://plato.stanford.edu/archives/win2010/entries/value-intrinsic-extrinsic/ (2010)

[27] Johnson, R.: Kant's moral philosophy. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Summer 2012 edn. http://plato.stanford.edu/archives/sum2012/entries/kant-moral/ (2012)

[28] Atkinson, C., Kühne, T.: Model-driven development: A metamodeling foundation. IEEE Software **20** (2003) 36–41

[29] Bell, S., Morse, S.: Sustainability Indicators — Measuring the Immeasurable? Earthscan (2008, 2nd Ed.)

[30] Yale Center for Environmental Law and Policy Yale University, Center for International Earth Science Information Network Columbia University: Environmental sustainability index report 2005. www.yale.edu/esi/ESI2005_Main_Report.pdf (2005)

[31] Marks, N.: Happy planet index. 'the new economics foundation', http://www.happyplanetindex.org/ (2012)

[32] Boden, A., Avram, G., Bannon, L., Wulf, V.: Knowledge Management in Distributed Software Development Teams - Does Culture Matter? In: 2009 Fourth IEEE International Conference on Global Software Engineering, IEEE (2009) 18–27

[33] sponsored by the BMBF: ARAMiS — Automotive, Railway, and Avionics in Multicore Systems. www.projekt-aramis.de (2012)

[34] Penzenstadler, B., Eckhardt, J.: A Requirements Engineering Content Model for Cyber-physical Systems. (2012)

[35] Geisberger, E., Broy, M., Cengarle, M.V., Keil, P., Niehaus, J., Thiel, C., Thönnißen-Fries, H.J.: agendacps — integrierte forschungsagenda cyber-physical systems. Technical report, acatech — Deutsche Akademie der Technikwissenschaften (2012)

[36] Lee, E.A.: Cyber physical systems: Design challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley (2008)