



VU Research Portal

Architecture design decision maps for software sustainability

Lago, Patricia

published in

2019 IEEE/ACM 41st International Conference on Software Engineering
2019

DOI (link to publisher)

[10.1109/ICSE-SEIS.2019.00015](https://doi.org/10.1109/ICSE-SEIS.2019.00015)

document version

Peer reviewed version

[Link to publication in VU Research Portal](#)

citation for published version (APA)

Lago, P. (2019). Architecture design decision maps for software sustainability. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society, ICSE-SEIS 2019 - Proceedings* (pp. 61-64). [8797634] Institute of Electrical and Electronics Engineers Inc..
<https://doi.org/10.1109/ICSE-SEIS.2019.00015>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:

vuresearchportal.ub@vu.nl

Architecture Design Decision Maps for Software Sustainability

Patricia Lago
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
p.lago@vu.nl

Abstract—In the context of software engineering, sustainability can be defined as the “capacity to endure” and to “preserve the function of a system over an extended period of time”. These definitions mainly point towards technical sustainability over time. Sustainability, however, may entail a much broader scope including economic, social and environmental sustainability as well.

In spite of the exciting hype around sustainability, we are very much lacking suitable instruments to design software-intensive systems that are sustainable and enable sustainability goals.

To fill this gap, we advocate the treatment of sustainability as a software quality property and defined a software sustainability assessment method that helps to make sustainability-driven design decisions. The method essentially relies on the definition of so-called “decision maps”, i.e. views aimed at framing the architecture design concerns around the four sustainability dimensions mentioned above - technical, economic, social and environmental sustainability.

In this context, this paper presents the notion of decision map. We then use a number of illustrative examples extracted from industrial projects, to summarize our lessons learned and reflections with general observations and future research directions, with the goal to spark a discussion in the research community.

Index Terms—Software architecture, Sustainability, Architecture design decisions, Architecture assessment, Decision map.

I. INTRODUCTION

In the context of software engineering, sustainability is often defined as the “capacity to endure” (borrowed from [24]) and to “preserve the function of a system over an extended period of time” [16]. These definitions mainly point towards what we call *technical sustainability*, i.e. “the preservation of the long-term use of software-intensive systems and their appropriate evolution in an execution environment that continuously changes” [14].

Thanks to its growing popularity, more and more scientific works in software engineering and software architecture address, or at least mention, sustainability from a technical perspective e.g. [3]. It must be observed, however, that many such works often share two weaknesses: (i) they confuse the notion of impact (which is measured in a certain point in time) with the notion of sustainability (which is a phenomenon observable only over a significant period of time); (ii) they limit the notion of sustainability to technical aspects (e.g. evolvability, maintainability, erosion) and sometimes environmental ones (e.g. energy consumption, power efficiency). Sustainability,

however, entails a much broader scope including economic and social aspects as well.

II. THE VISION

In software architecture we are mostly used to think in terms of technical impact [3] and economic concerns [20]. Only recently we have started thinking in terms of environmental ones (e.g. [2], [26]). In one way or another, we are also more used to compartmentalize these concerns, whereas sustainability is a matter of assessing the big picture. To this end, we argue that tradeoff analysis provides the perfect mechanism to simultaneously consider all sustainability aspects that might be relevant for a certain system, and hence lay the foundation for a much better understanding of what is sustainability in software-intensive systems. We also argue that software architecture is an ideal abstraction level to gather the above-mentioned *big picture*: by combining architecture assessment methods with sustainability tradeoffs, we would be finally able to deliver software that will support sustainability *by design*.

In spite of the exciting hype around sustainability, we are very much lacking suitable instruments to design software-intensive systems that are truly sustainable or that enable sustainability goals.

To fill this gap, we advocate the treatment of sustainability as a software quality property and defined a software sustainability assessment method, called SoSA [14], that helps making sustainability-driven design decisions. The method essentially relies on the definition of so-called “decision maps” framing the architecture design concerns around the four sustainability dimensions mentioned above - technical, economic, social and environmental sustainability.

In this paper, we present the notion of decision map resulting from over four years of research in collaboration with the private- and public sectors. We use a number of illustrative examples extracted from a selection of our industrial projects to summarize our lessons learned and reflections.

The rest of the paper is organized as follows. In Section III we present the decision maps along with our lessons learned. In Section IV we present our general reflections. In Section V we describe the related work, and we close the paper and discuss future work in Section VI.

III. DECISION MAPS

Architecture evaluations should, among other, “*support decision making where architectures are involved*”, “*assess*

the quality of architectures with respect to their intended purpose” and “determine whether architecture entities address their intended purpose” [13]. If one architecture’s intended purpose is sustainability, we should provide architects suitable instruments to make decisions that lead to some stated sustainability purposes, or concerns. This is exactly the aim of our decision maps, i.e. making explicit the sustainability concerns that should be considered by an architecture or architecture entity [13].

The notion of decision map (DM) is the result of multiple research projects carried out incrementally in collaboration with industry and the public sector. After introducing the visual notation, the following uses the example DMs (see Figure 2) resulting from the projects summarized in Table I to draw our main lessons learned.

A. The Notation

The DM notation (exemplified in Figure 1) is the result of multiple revisions in projects and experiments (e.g. [7]). It essentially frames the expected impact of a software architecture on the target sustainability concerns. Accordingly, the notation entails sustainability impacts (see in the Figure, the areas in different shades of grey), sustainability architecture design concerns (the colored boxes), and the effects among architecture entities and concerns (the arrows between boxes).

1) *On Sustainability Impacts:* Our definition of sustainability impact builds upon that of Hilty et al. [10] along three levels: **Immediate impacts** refer to changes which are immediately observable. These are the concerns that are addressed within the current software project, and are expected to be directly addressed by the architecture entities. In the example of Figure 1, the DM models that the architecture modularity is expected to reflect the structure of the development team. **Enabling impacts** arise from use over time. This includes the opportunity to consume more (or less) resources, but also shorten their useful life by obsolescence (when we buy a new smart phone just because incompatible with newer applications) or substitution (when e-book readers replace printed books). In our example, team-driven modularity is expected to enable a positive impact on the development effort. While not directly measurable within the scope of the current software project, this concern can be monitored over multiple projects, hence provide understanding on the extent of such positive impact. **Systemic impacts**, in turn, refer to persistent changes observable at the macro level. This includes behavioral change and economic structural change. This may translate into (negative) rebound effects by converting efficiency improvements into additional consumption, or new risks - like our dependence on ICT networks that makes a digital society also vulnerable. In our example, a sustained reduction of the development effort required by software projects may result in less developers needed in the first place, or employees with different technical competences, and hence a change in the company hiring strategy.

2) *On Sustainability Concerns:* Architecture design concerns can be of four sustainability types [14], [15]: **Techni-**

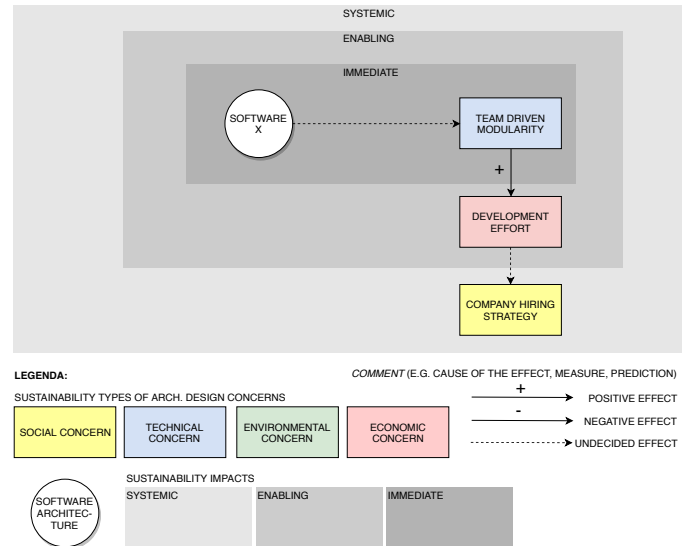


Fig. 1: DM Notation

cal sustainability addresses the long-term use of software-intensive systems and their appropriate evolution in an execution environment that continuously changes. **Economic sustainability** focuses on preserving capital and (economic) value. **Social sustainability** focuses on supporting current and future generations to have the same or greater access to social resources by pursuing generational equity. For software-intensive systems, this dimension encompasses the direct support of social communities in any domain, as well as the support of activities or processes that indirectly create benefits for social communities. **Environmental sustainability** aims at improving human welfare while protecting natural resources. For software-intensive systems, this dimension aims at addressing ecologic concerns, including energy efficiency and ecologic awareness creation.

In our example, team driven modularity expresses a technical concern, development effort translates into economic impacts, and company hiring strategy reflects the organizational social structure and as such is a concern of social nature.

3) *On the Effects:* We have identified three types of effects among software architecture entities and concerns: positive, negative, and undecided. Their semantics is quite self-explanatory, and this is in our experience an essential requirement to keep the decision maps simple and intuitive enough as an instrument for decision makers with different backgrounds and competencies.

Decision maps are also meant to be used and re-used across various projects. As such, incremental learning will allow effects to evolve and consolidate over time. For example, by observing the impact of multiple projects on the concerns shown in Figure 1, the *undecided* impact on the company hiring strategy might become positive (e.g. if the company is agile enough to adapt timely) or negative (e.g. if the company decides for a rapid turnover strategy, with a possible negative effect on the working atmosphere). Either ways, the DM can be

used to capture the decisions and the consequent effects. If so, it can provide a valid yet simple decision making instrument.

B. Lessons Learned from Example Projects

Among the projects we carried out over the years, we picked a few to illustrate how DMs can help the analysis of the sustainability concerns. For each example, we summarize our reflection and our insights in the form of lessons learned. The projects are summarized in Table I.

In the **KPMG Qubus platform** (Figure 2a) the target concerns were about the effect of different software releases on energy consumption while maintaining a satisfactory level of performance (expressed in terms of execution time). The initial DM hypothesized a positive effect of successive releases on POWER CONSUMPTION, and an hypothetical corresponding negative effect on EXECUTION TIME. Both concerns are clearly technical. Actual measures, however, showed that: (i) the main design decision responsible for variations in power consumption regards the *deployment strategy* with an effect measured up to 10%; (ii) the main feature responsible for negative effects on execution time is *data load*, i.e. the exchange of data between client- and server side necessary for executing the user services; in turn, execution time has a negative effect on power consumption; (iii) surprisingly, we could not determine any significant correlation between execution time and ENERGY CONSUMPTION, leaving the effect of the first on the second as *undecided*.

Lesson #1: Decision Maps can be refined with the actual causes of effects and effect measures.

- ▶ Initially, DMs can be used for framing the **foreseen effects** of early architecture design decisions on the target concerns.
- ▶ After architecture evaluation or implementation, observations and measures can be used to refine the DMs with the **actual effects** on the target concerns.

The exploratory nature of the **Smart Work** project makes the decision maps especially useful for identifying concerns of various sustainability types and regarding different impact levels (Figure 2b). By drawing an early decision map, the architect can reason about the potential pros and cons of alternative solutions, and explore the design space before making specific design decisions. Accordingly, the focus of such an early DM is on mapping the sustainability concerns and their potential crosscutting effects, before making design decisions.

This project was carried out in 2016, in collaboration with the Amsterdam Smart City organisation, and was followed by a focus group in the 28th International Conference on Advanced Information Systems Engineering (CAiSE). The decision map in Figure 2b captures the refined interpretation chosen as the preferred DM for the problem of providing software services for flexible work. We observe that: (i) like in the previous project, also here concerns with immediate

impact are technical; (ii) given the broad scope of the problem, relevant concerns cover all sustainability types; (iii) in some cases, the same concern (see FLEXIBILITY in the Figure) can address multiple sustainability aspects (e.g. *technical* flexibility is necessary to provide a satisfactory quality of service for home-working, and if present it can have a positive impact on *social* flexibility like the ability to balance work-, family- and leisure time).

Lesson #2: Decision Maps can be used for design space exploration, with a major focus on characterizing and scoping sustainability concerns.

- ▶ DMs can effectively illustrate the problem space in terms of the relevant sustainability concerns and their potential cross-cutting effects.
- ▶ A concern can possibly capture multiple sustainability perspectives (cf. flexibility being both technical and social).
- ▶ The three levels of impact provide the natural context to reason about the implications of decisions affecting concerns from the short- (immediate and enabling impact) to the longer term (systemic impact).

The **Mobility as a Service (MaaS)** platform [18] was a project in collaboration with the City of Amsterdam (Figures 2c and 2d). Similar to Smart Work, this project focused on providing smart city solutions for Amsterdam. In this case, however, the objective was to *imagine* a MaaS platform where individuals share their transport means (e.g. bike, car). The ultimate goal is to offer higher flexibility, better transport opportunities while reducing the city traffic as well as the number of unused transport. As individuals can be both providers and users of the MaaS platform, it became clear quickly that we had to separate the perspectives in separate DMs: the user perspective in Figure 2c and the provider perspective in Figure 2d. Naturally, similar to Smart Work, the two DMs cover a rich set of relevant concerns of all four sustainability types.

In particular, it is interesting to observe that: (i) the platform introduces a software dependency (see top-most, right-hand side of the DMs) that, in case of malfunction, leads to a negative effect in the IMPACT OF IT FAILURE; such impact has a predominant *social* connotation for the user (i.e. getting late to work or disrupting one's day schedule) and an *economic* connotation for the provider (i.e. losing incomes). (ii) The platform offers the *users* higher flexibility (cf. FLEXIBILITY OF MOBILITY MEANS), which in turn may lead to complex tradeoffs between: more CARS ON THE ROAD (hence more traffic) and more CARBON EMISSIONS; but also less CARS PARKED (hence more FREE SPACE IN URBAN AREAS) and LESS CARS PRODUCED (with negative economic effects for the car industry but positive effects in terms of its carbon footprint); for *providers*, the platform offers a source of income

Project Title	Abstract	Concerns	Decision Map
KPMG Qubus [25]	The object of this study is a platform named Qubus, which supports Governance, Risk, and Compliance (GRC) processes and is developed and maintained by KPMG. The selected software product supports a wide range of enterprise management processes including financial control management, strategic and operational risk management, compliance management and external audit management.	What is the impact of different software releases on energy consumption?	Fig. 2a
Smart Work [1]	In 2025 urban transport in the Amsterdam Metropolitan Area will be emission free. To get there, the next few years will witness major disruptions in transportation that will hinder e.g. reaching the office and school in time. The solution is to seek alternative ways to work and study so that to remove the need to physically reach one work- or study places. In doing so, however, the way we work, study, interact with out family, spend our free time will change forever. How do we make sure that this transformation toward a smart and sustainably city and lifestyle will be smooth and successful? And how do we foresee the behavioural changes that will mark our futures?	How to choose between “travel” and “un-travel” (i.e. substitute activities that require physical mobility with others that allow virtual presence)? What are the long-term implications of virtual mobility? For instance, when workers do need using physical offices much less, or the boundary between work and private will be blurred?	Fig. 2b
MaaS (user perspective) [18]	Amsterdam Smart City put forward its vision for smart mobility as follows: <i>Mobility will be a custom-made service for everyone, with shared and emission-free cars that drive themselves. The city will have more room for pedestrians and bicycles (shared or otherwise), more greenery, and less room for parking spaces</i> ”. To realize this vision, however, multiple complications must be considered. For instance, the population of Amsterdam is increasing and hence more houses are being built in the city. At the same time, the number of tourists is increasing, too, adding to the overall population growth. While this would result in an increasing need for transportation (more cars on the roads, maybe more roads), Amsterdam wants to achieve exactly the opposite (less cars, less roads, more pedestrian areas). A solution for the stress on the current/future transportation infrastructure could be offered by a Mobility as a Service (MaaS) platform, where end-users pay for the mobility they use instead of purchasing transportation means, and hence adjust the supply of transportation means optimally to the demand of mobility. This study wants to explore the potential implications of introducing such MaaS platform.	What would be the impact of an open MaaS platform on the citizen? What are the socio-economic implications for the city? How would a MaaS solution result in more efficient transportation? Would it bring higher quality of life? Less pollution in the city?	Fig. 2c
MaaS (provider perspective) [18]	<i>see Amsterdam Smart City case above</i>	What are the prospective benefits for the citizen to share his or her transportation means? And what are the potential risks or disadvantages?	Fig. 2d

TABLE I: Summary of the Projects

(thanks to the POSSESSION OF PERSONAL TRANSPORTATION MEANS - now shared), and it is expected to further shrink the automotive market (because a growing number of cars will be shared, the number of cars needed will decrease, thus, there will be fewer CARS PRODUCED). (iii) Finally, for both users and providers a data sharing is a prerequisite for the MaaS platform to be successful. However, while users are, to some extent, already accustomed to relax their PRIVACY and share their usage data, providers are FORCED TO SHARE THE DATA about their means of transport and their own mobility habits (necessary for the platform to know when a transport means is available).

Lesson #3: Different Decision Maps can be used to capture the sustainability concerns of different stakeholders.

- ▶ The “big picture” provided by a DM can help uncover complex sustainability network effects and behavioral patterns (cf. systemic effects on the automotive market).
- ▶ When relevant for multiple stakeholders, the same concern can possibly have different sustainability effects (cf. data sharing being potentially more critical for providers than users).

IV. REFLECTION: OBSERVATIONS AND RESEARCH DIRECTIONS

In addition to the lessons presented in Section III, we could draw the following general observations:

- Thanks to its simplicity, DMs can uncover interesting phenomena that help informed design decision making further. For example, the effects between socio-technical flexibility and work efficiency in Figure 2b reveal a so-called network effect [19] where work efficiency enabled by smart work services has the potential to further enforce social flexibility. This, in turn, can have an important systemic impact on modern society (and for this reason, in the DM the associated concern is placed across the two impact dimensions). By uncovering this network effect, architects can design for achieving this potential impact, and put in place measures to observe if (when implemented) the architecture does realize the designed-for sustainability goals.
- DMs have been conceived for designing software architectures addressing sustainability concerns. However, they are not “software-specific” and, as regularly observed by our industrial partners, they could be extended to the notion of enterprise architecture [8] and used to uncover the potential networks of stakeholders linked to the mapped sustainability concerns. In doing so, they can facilitate the creation of innovative sustainability business models. Moreover, using DMs as the central view for both software- and enterprise architecting would help addressing the still-open “business-IT alignment” problem.

We have also identified the following promising directions for future research:

- DMs provide views that exclusively focus on the sustainability concerns pertaining a certain software archi-

ecture design as a whole. As such, they neglect the relation between a certain sustainability concern and the architecture elements that address it. As described by the IEEE/IEC/ISO 42010 Standard [12], an architecture view (and its associated viewpoint) frames one or more concerns. For instance, it should illustrate *how* a certain concern (e.g. IMPACT OF IT FAILURE), is addressed by (a set of) architecture elements. This is a difficult problem, namely how to map a complex network of sustainability concerns on an equally complex structure of architecture elements. In spite of its complexity, this mapping is necessary for both top-down architecture design decision making based on the foreseen sustainability effects, and bottom-up monitoring of the actual sustainability effects achieved by the implemented architecture (cf. Lesson #1). A possible research direction could be to define a viewpoint in terms of three types of model kinds: one defining the architecture view (or views), one defining the DM, and one defining the mapping between elements of the architecture views and elements of the DMs.

- DMs emphasize the fact that concerns are an essential driver supporting design decision making. However, the notion of *concern space* is still ill understood (cf. Hilliard [9]). While DMs help in modeling, interrelating, and analyzing sustainability concerns, they do not help embracing the fact that not all concerns belong in full to either the problem space or the solution space. An example of cross-cutting concern is maintainability, which emerges from the fact that we have solutions (solution space) that we have to maintain (problem space). A first research step is to investigate past research efforts in e.g. design space exploration [21] and framing elements of the design space [23]).
- There is a big difference, and much confusion, between long-term sustainability and short-term impact: while DMs aim to help understanding the first, they are often used to capture the second. The confusion is quite the dilemma: sustainability is intrinsically addressing a balancing act that can be observed only in a rather long period of time. For example, it is yet to be seen if smart work solutions will have a sustainable socio-technical, environmental and economic impact, where a.o. work effectiveness should balance the lack of social interaction; costs should balance revenues; and ecologic impacts should preserve the environment. While the DM notation can capture these balancing networks of sustainability concerns, we often limit our reasoning to shorter-term impacts, like the reduced need for office space (and associated economic savings) enabled by the flexibility of e.g. home-working. We argue that (similar to the notion of reference architecture), DMs framing recurrent problems should be reused as *reference DMs*, refined and enriched to gain better understanding of the sustainability implications. Only in this way we can mature from short-term impacts to long-term sustainability.

V. RELATED WORK

This work stems from our early interpretation of sustainability as a software quality property [15]. From there, we take an architecture design perspective, while the work of Becker et al. [4] takes a requirements engineering perspective. As such they use illustrations similar to DMs, but add a fifth sustainability dimension (that of the individuals - especially relevant for requirements engineering) and are more informal to “... serve as a visual aid to support interactive collaboration among stakeholders” (cf. [4]).

There are many more research works in the field of software engineering and sustainability. However, most zoom into specific software sustainability aspects. For example, good references include the work of: Cai et al. [5], which uses the notion of design rules to detect flows at the software architecture level over extended periods of time [technical sustainability]; Hindle [11] relating the direct impact of software change on energy consumption, and Li et al. [17] presenting practices that indirectly (i.e. if adopted by developers) can help reducing the energy consumption of mobile applications [environmental sustainability]; Widdicks et al. [27] study socio-technical sustainability at the individual level, while Tamburri et al. [22] do so at the level of development teams and whole organizations [social sustainability]. Our work is orthogonal to the ones above, as decision maps can be used to frame the concerns of each of the sustainability aspects they address, and provide a higher level picture on their interdependencies.

VI. CONCLUSION AND FUTURE WORK

This paper presents the notion of “decision maps” framing the architecture design concerns around the four dimensions of technical, economic, social and environmental sustainability. Decision maps are exemplified in a number of illustrative examples extracted from industrial projects, and accompanied by our lessons learned and general reflections.

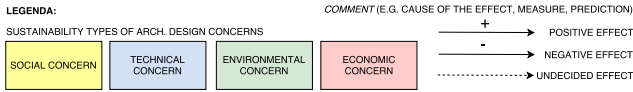
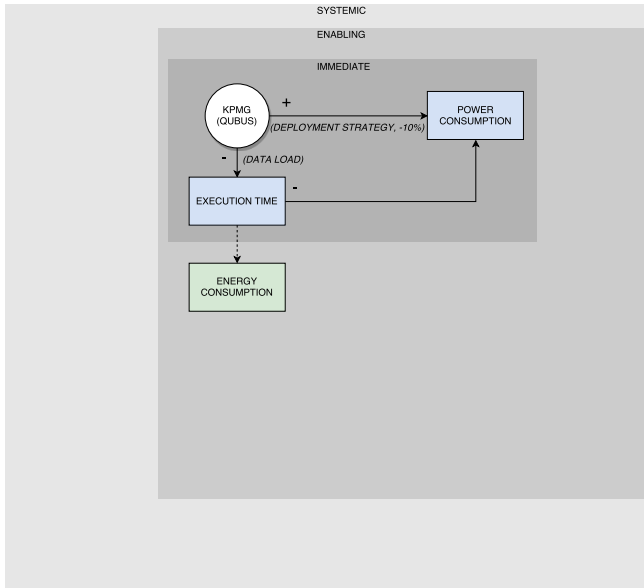
As future work, we are planning to follow-up our lessons learned. In particular, regarding Lesson #1, we plan to extend the DM notation to associate concerns with metrics, then used on implemented architectures to gather measures. To this aim we also plan to use the decision maps for dashboarding and visual analytics by (i) extending the sustainability quality characteristics resulting from [6] into reusable concerns; (ii) sharing these concerns in a number of case studies in collaboration with our industrial partners to identify suitable metrics; (iii) gathering data (measures); and (iv) applying AI techniques for reasoning and learning.

ACKNOWLEDGMENTS

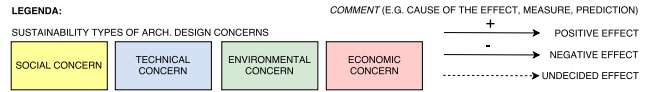
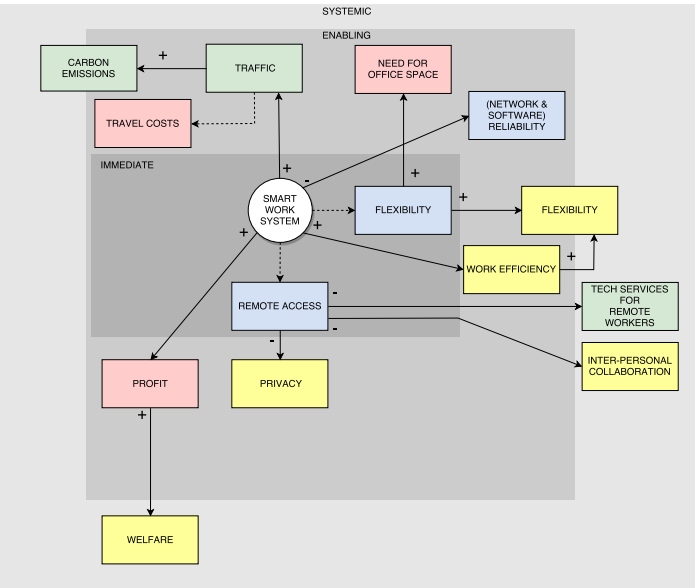
We would like to thank the City of Amsterdam, the Amsterdam Smart City initiative, and KPMG to borrow their projects for this research. Part of the work has been carried out in the context of the RVO GreenServe project. Thanks go also to the many students of the Vrije Universiteit Amsterdam that helped inspiring this research, and Wouter-Paul Trienekens (CGI Group) for his valuable feedback.

REFERENCES

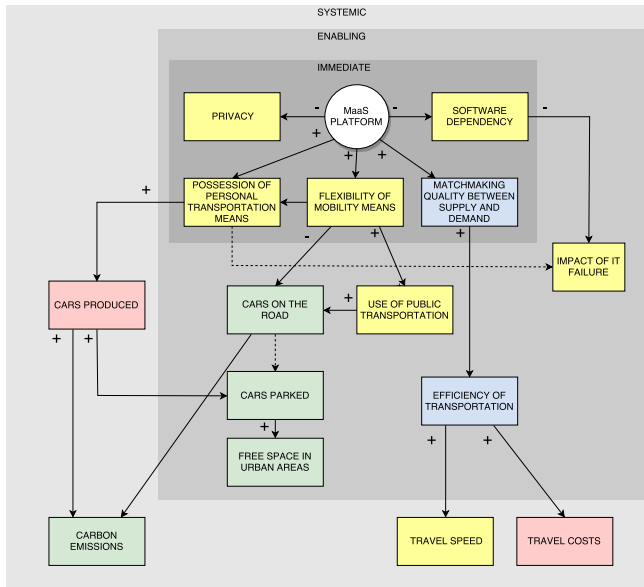
- [1] Amsterdam smart city. <https://amsterdamsmartcity.com/>, 2010. Accessed: 2018-5-23.
- [2] F. Alizadeh Moghaddam, G. Procaccianti, G. A. Lewis, and P. Lago. Empirical validation of cyber-foraging architectural tactics for surrogate provisioning. *The Journal of systems and software*, 138(April):37–51, Apr. 2018.
- [3] P. Avgeriou, M. Stal, and R. Hilliard. Architecture sustainability [guest editors' introduction]. *Software, IEEE*, 30(6):40–44, 2013.
- [4] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. Venters. Requirements: The key to sustainability. *IEEE Software*, 33(1):56–65, Jan. 2016.
- [5] Y. Cai, L. Xiao, R. Kazman, R. Mo, and Q. Feng. Design rule spaces: A new model for representing and analyzing software architecture. *IEEE Transactions on Software Engineering*, (1):1–1.
- [6] N. Condori-Fernandez and P. Lago. Characterizing the contribution of quality requirements to software sustainability. *Journal of Systems and Software*, 137:289–305, 3 2018.
- [7] S. Espana and P. Lago. Software sustainability assessment (SoSA) exercise report. Technical report, Vrije Universiteit Amsterdam, 2016.
- [8] D. Greefhorst and E. Proper. *Architecture Principles: The Cornerstones of Enterprise Architecture*. Springer Science & Business Media, Apr. 2011.
- [9] R. Hilliard. Invited Lecture on “Architecture Description: Standards and Open Research Challenges”, International Software Architecture PhD School (ISAPS), Apr. 2018.
- [10] L. M. Hilty, P. Arnfalk, L. Erdmann, J. Goodman, M. Lehmann, and P. A. Wäger. The relevance of information and communication technologies for environmental sustainability – a prospective simulation study. *Environmental Modelling & Software*, 21(11):1618–1629, Nov. 2006.
- [11] A. Hindle. Green mining: A methodology of relating software change and configuration to power consumption. *Empirical Softw. Engg.*, 20(2):374–409, Apr. 2015.
- [12] International Standardization Organization (ISO). ISO/IEC/IEEE 42010:2011 - Systems and Software Engineering – Architecture Description. Technical report, 2011.
- [13] ISO/IEC TC /SC 7/WG 42. ISO/IEC/IEEE 42030:2017 - Enterprise, systems and software – Architecture evaluation framework. Technical report, International Organization for Standardization (ISO), Oct. 2017.
- [14] P. Lago. SoSA: A Software Sustainability Assessment method. <http://goo.gl/HuY6tf>, Jan. 2016.
- [15] P. Lago, S. A. Kocak, I. Crnkovic, and B. Penzenstadler. Framing sustainability as a property of software quality. *Communications of the ACM*, 58(10):70–78, 2015.
- [16] P. Lago and B. Penzenstadler. Editorial: Reality check for software engineering for sustainability—pragmatism required. *Journal of Software: Evolution and Process*, 29(2), Feb. 2017.
- [17] D. Li and W. G. J. Halfond. An investigation into energy-saving programming practices for android smartphone app development. In *Proceedings of the 3rd International Workshop on Green and Sustainable Software (GREENS)*. ACM, 2014.
- [18] T. Niggebrugge, S. Vos, and P. Lago. *The Sustainability of Mobility as a Service Solutions Evaluated through the Software Sustainability Assessment Method*. VU Technical Report, Jan. 2018.
- [19] G. G. Parker, M. W. Van Alstyne, and S. P. Choudary. *Platform Revolution: How Networked Markets Are Transforming the Economy and How to Make Them Work for You*. W.W. Norton & Company, 2016.
- [20] E. R. Poort and H. van Vliet. RCDA: Architecting as a risk- and cost management discipline. *The Journal of systems and software*, 85(9):1995–2013, Sept. 2012.
- [21] M. Razavian, A. Tang, R. Capilla, and others. In two minds: how reflections influence software design thinking. *Journal of Software Maintenance and Evolution: Research and Practice*, 2016.
- [22] D. A. Tamburri, P. Kruchten, P. Lago, and H. v. Vliet. Social debt in software engineering: insights from industry. *Journal of Internet Services and Applications*, 6(1):10, May 2015.
- [23] A. Tang, T. Antony, and M. F. Lau. Software architecture review by association. *The Journal of systems and software*, 88(1):87–101, 2014.
- [24] United Nations World Commission on Environment and Development. Report: Our Common Future. In *United Nations Conference on Environment and Development*, 1987.
- [25] R. Verdecchia, G. Procaccianti, I. Malavolta, P. Lago, and J. Koedijk. Estimating energy impact of software releases and deployment strategies: The KPMG case study. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 257–266. IEEE CS, Nov. 2017.
- [26] R. Verdecchia, R. A. Saez, G. Procaccianti, and P. Lago. Empirical evaluation of the energy impact of refactoring code smells. In *International Conference on ICT for Sustainability, ICT4S*, 2018.
- [27] K. Widdicks, T. Ringenson, D. Pargman, V. Kuppusamy, and P. Lago. Undesigning the internet: An exploratory study of reducing everyday internet connectivity. In B. Penzenstadler, S. Easterbrook, C. Venters, and S. I. Ahmed, editors, *International Conference on Information and Communication Technology for Sustainability (ICT4S)*, volume 52 of *EPiC Series in Computing*, pages 384–397. EasyChair, 2018.



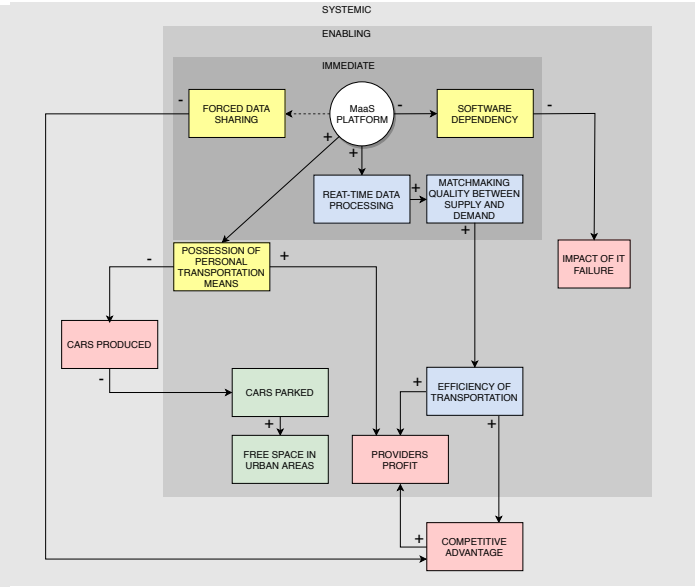
(a) KPMG Qubus



(b) Smart Work Service



(c) Mobility as a Service - User Perspective



(d) Mobility as a Service - Provider Perspective

Fig. 2: Decision Maps