Using the Doughnut Model to Support Sustainable Quality Requirements in iStar

Jennifer Horkoff[1] and Tong Li[2]

[1] Chalmers | University of Gothenburg, Sweden, [2] Technical University of Beijing, China
jennifer.horkoff@gu.se, litong@bjut.edu.cn

Abstract. Sustainability has become a critical topic, not only from an environmental perspective, but in the creation of software systems which are themselves sustainable. The sustainability literature has made use of the doughnut model from economics to understand the delicate balance needed for sustainable development. This model emphasizes the notion of having just the right amount of a resource (e.g., food, water) and not too much, else negative consequences (water depletion, starvation) may be felt by others in the ecosystem. Although iStar has covered well the notion of trade-offs between qualities (e.g. security vs. usability, performance vs. maintainability), the implicit aim of the work is always to maximize qualities. In this work we aim for "just enough" quality by applying the doughnut economic model to quality requirements in iStar. Overall, we propose a visually appealing model which emphasizes a sustainable balance between qualities.

Keywords: sustainability, quality requirements, quality analysis

1 Introduction

Sustainability has become a critical topic, not only from the perspective of environmental or natural sustainability, but in creating software systems which are themselves sustainable [1]. Sustainability literature has made use of the doughnut model from economics to understand the delicate balance between desired aspects for inclusive and sustainable economic development [2]. This model emphasizes the notion of having just the right amount of a resource (e.g., food, water) and not too much, else negative consequences (water depletion, starvation) may be felt by others in the ecosystem. Keeping in mind the importance of sustainable software system development, we believe that this mindset can apply well to the area of non-functional requirements (NFRs) and software qualities.

Many methods exist to model and reason over the achievement of NFRs and qualities (e.g., the NFR Framework [3], iStar [4], and similar frameworks). Although this work has covered well the notion of trade-offs between qualities (e.g. security vs. usability, performance vs. maintainability), the implicit aims of the work is always to maximize the achievable qualities. Although previous work deals with satificing or satisfaction of qualities, as far as we are aware, existing work on requirements qualities does not consider the possibility of over-saturation of satisfaction, having too much, or more than is needed of a particular quality, which may have adverse effects on other qualities (e.g., too much usability is costly, too much performance hinders modifiabilty). To create systems which are sustainable, we should focus on “just enough” quality in various dimensions, in line with agile thinking and the realities of business.

In this work we apply the doughnut economic model to qualities in iStar. We evaluate our idea through application to historical goal model examples, focusing on qualities, considering changes to existing reasoning techniques. By aiming to achieve just enough of various qualities, we argue that systems are easier to construct and maintain, and are therefore more likely to be successful and sustainable. Although other models for sustainable software development have been proposed (e.g., [1, 5]) they do not specifically focus sustainability-minded ways to visualize general quality requirements.

In this paper, we briefly introduce the doughnut model in Sec. 2, then present our ideas via an example in Sec. 3. We discuss issues such as semantics and reasoning, including the notion of over-saturation in Sec. 4. We conclude and describe future plans in Sec. 5.

2 The Doughnut Economic Model

The doughnut model combines together both planetary boundaries (climate change, land use) with social boundaries (income, education) into one holistic view of sustainable development [2] (see Fig. 1). In this model, the inner boundary refers to social foundations, while the outer boundary forms the envi-

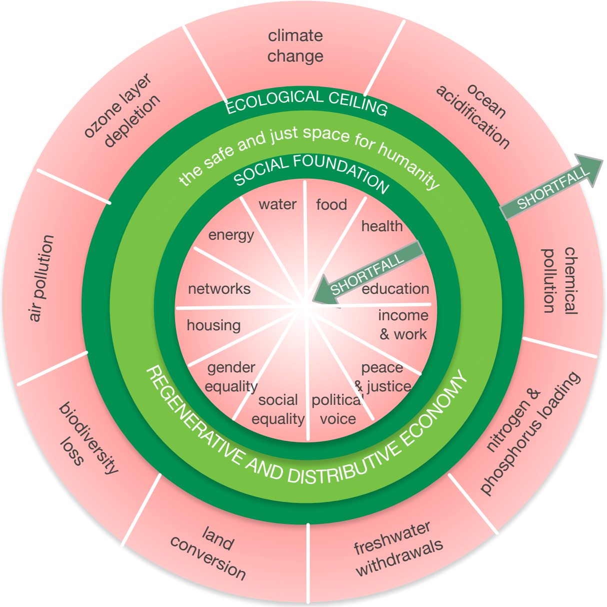


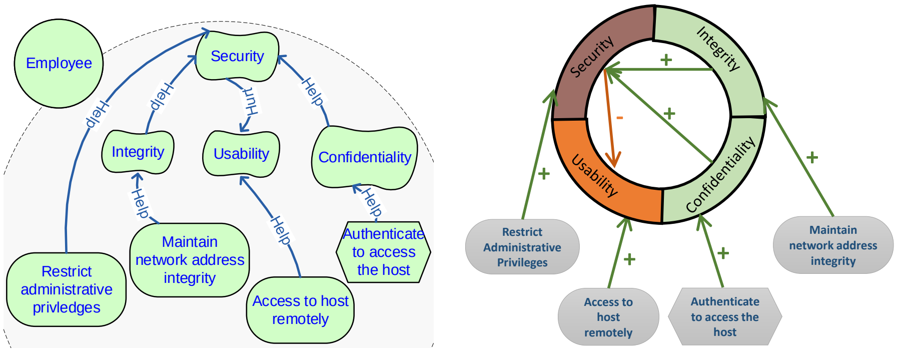
Fig. 1: The Doughnut Economic Model, reproduced from [2]

ronmental dimensions. The idea is to find a safe space for humanity that balances all of these environmental and social factors. In our work, we are inspired by this model, but adapt a simpler version to better suit quality requirements, keeping the general principle of balance.

3 The Doughnut Model applied to Qualities in iStar

In order to further motivate the need for the new visualization and to demonstrate how it would look and work in practice, we present two examples. We start with a simple example from [6] focusing on security trade-offs, recreated in Fig. 2a. We focus the top part of the original model, with the quality goals (softgoals) and the functional elements (tasks and goals) which directly impact qualities. To illustrate what a trade-off may look like we add a hurt link from security to usability. We also add an extra security function to demonstrate potential over saturation. We redraw this extended model using a visualization inspired by the doughnut model, in Fig. 2b. Here we show the four qualities from the original model as labels along the desirable inner part of the "doughnut". We shade the doughnut segments representing the qualities different shades of green, orange, and brown to reflect their level of satisfaction, for example, security has three incoming help links, so is potentially over saturated (brown), while usability is hurt (orange). We discuss reasoning further in Sec. 4. Specific colors and visual elements may have to be adjusted based on future usability studies.

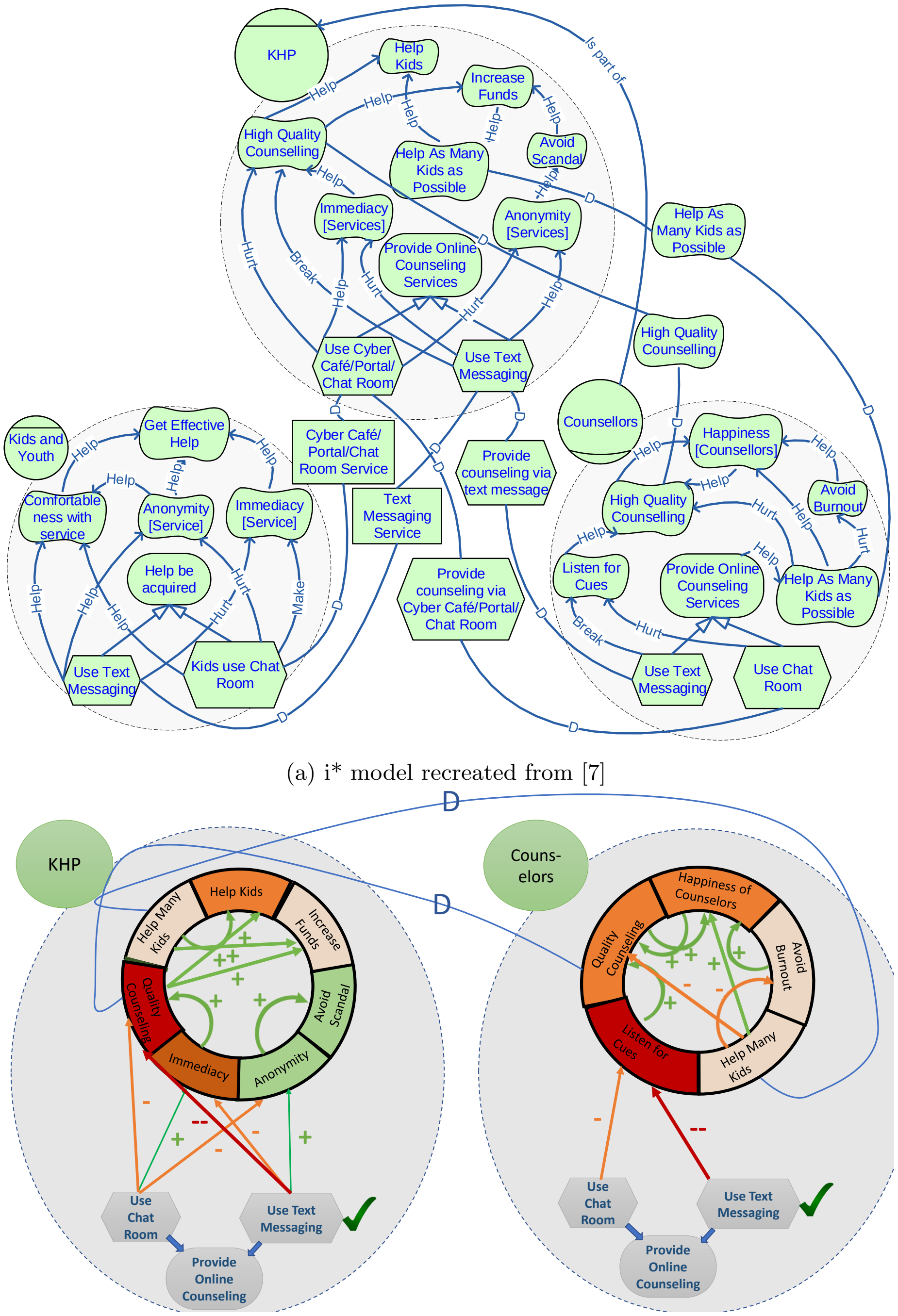
In the second example, we recreate two of the three actors in Fig. 3a, again focusing on qualities. Here we can see the doughnut model beginning to scale to more qualities, in this case seven and five. Due to the increasing number of qualities in the doughnut, we can no longer use straight internal contribution links as in the first example, here we use curved green and red arrows for help and hurt contributions within the doughnut and straight links from functional elements to quality. Again, we indicate positive/negative via color, and severity by link thickness (to be evaluated and improved in future studies). Here we start to explore doughnuts in iStar actors with typical dependencies. We see the


Employee
Security
Help
Help
Hurt
Help
Integrity
Usability
Confidentiality
Help
Help
Help
Restrict administrative privledges
Maintain network address integrity
Access to host remotely
Authenticate to access the host
Security
Integrity
Usability
Confidentiality
+
+
-
+
+
+
+
+
Restrict Administrative Privileges
Access to host remotely
Authenticate to access the host
Maintain network address integrity

(a) Excerpt of an i* model recreated from [6] with additional links and elements

(b) 2a captured using Doughnut model

Fig. 2: Doughnut Visualization applied to part of i* model from [6]



(a) i* model recreated from [7]

(b) Two Actors from 3a captured Using the Doughnut syntax

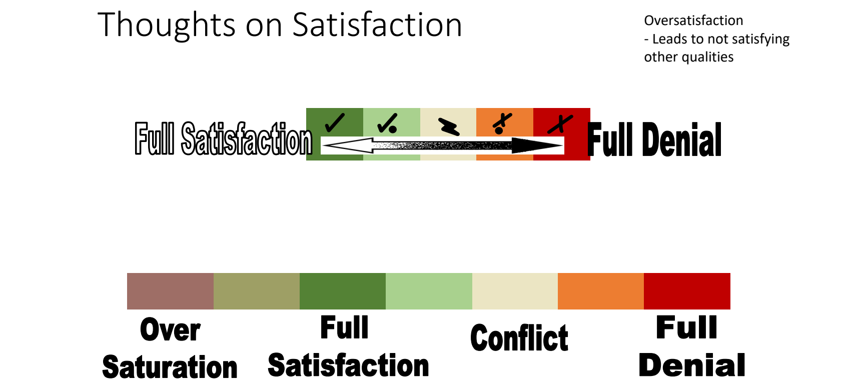
Fig. 3: Doughnut Visualization applied to part of i* model from [7]

common visual clutter, but there is also a clear distinction between quality and function in the model.

4 Semantics and Reasoning

Existing descriptions of NFR or iStar/i* semantics, supporting goal reasoning, can be adjusted to account for our proposal. Traditionally, satisfaction is thought of on a scale from fully denied to fully satisfied, as is shown in Fig. 4a, with some variation in the names, colors, and format of the labels. In this work, we introduce the notion of *Over-Saturation*, to show that going beyond satisfaction (or satisficing) is both possible and undesirable (see left side of Fig. 4b). We defined over-saturation in this case as: satisfying a quality to a degree that is more than is necessary to achieve a desirable state. This definition is vague by construction, as satisfaction and over-saturation depend on the quality and context. For example, for security, satisfaction may mean that the majority of users are happy with the level of security provided by a system. Here even 'majority' will depend on the size of the user base, if there are only five major customers, one customer being dissatisfied with security is significant. However, if there are 1000 customers, it may be acceptable that 90% of customers are satisfied with the system security. One can satisfy the additional 10% by adding more security features, but this may negatively affect cost, usability, or performance in such a way that the overall trade-off is negative. In such a case, adding more security interventions may lead to over-saturation.

Existing goal model reasoning approaches such as reasoning described in the NFR Framework [3] or iStar/i* [7] can be adjusted to account for the new concept of over-saturation. This would mean the introduction of new labels or new color codes. Previously, the goal model community has struggled with how to determine whether something is fully or partially satisfied or denied (how much is enough?), with different interpretations of goal models leading to different procedures and semantics [8]. The question of when a quality moves from satisfied
to
ple
rec



Thoughts on Satisfaction

Oversatisfaction
- Leads to not satisfying other qualities

(b) New Proposal for Quality Satisfaction

Fig. 4: Satisfaction of Qualities: Old vs. New

goal is partially satisfied, satisfied, partially saturated, or over-saturated. For other procedures, e.g., [9] where 'promotion' of incoming contribution values is not accounted for (e.g., any number of incoming partially satisfied values results in partially satisfied), extending to support over-saturation is more challenging. Generally, one needs either a way to 'count' or quantify incoming positive evidence, along with thresholds, to determine when a quality is too satisfied; or this decision must be passed to users, relying on domain knowledge and expertise.

5 Conclusions

We have proposed an alternative way to visualize, think about, and reason over quality requirements, including the notion of quality over-saturation. Through examples, we have begun to show practical feasibility, and we have discussed how the new concepts could influence existing goal reasoning techniques.

Future work will consider views over the model, e.g., incrementally showing the Doughnut model elements, contribution links, contributing elements, etc. Tooling should be adapted, ideally automatically transform existing i* or iStar models into the new notation. We are working on an evaluate plan involving surveys, prototypes and user studies with both iStar experts and novices. We welcome others to participate in the use and development of these ideas.

Acknowledgements

Supported by Vetenskapsrådet project "Non-Functional Requirements for Machine Learning: Facilitating Continuous Quality Awareness (iNFoRM)".

References

1. S. S. Mahmoud and I. Ahmad, "A green model for sustainable software engineering," *International Journal of Software Engineering and Its Applications*, vol. 7, no. 4, pp. 55–74, 2013.
2. K. Raworth, "A safe and just space for humanity: can we live within the doughnut," *Oxfam Policy and Practice: Climate Change and Resilience*, vol. 8, no. 1, pp. 1–26, 2012.
3. L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering.* Springer Science & Business Media, 2012, vol. 5.
4. F. Dalpiaz, X. Franch, and J. Horkoff, "istar 2.0 language guide," *arXiv preprint arXiv:1605.07767*, 2016.
5. A. Raturi, B. Penzenstadler, B. Tomlinson, and D. Richardson, "Developing a sustainability non-functional requirements framework," in *Proceedings of the 3rd International Workshop on Green and Sustainable Software*, 2014, pp. 1–8.
6. G. Elahi and E. Yu, "Modeling and analysis of security trade-offs–a goal oriented approach," *Data & Knowledge Engineering*, vol. 68, no. 7, pp. 579–598, 2009.
7. J. Horkoff and E. Yu, "Interactive goal model analysis for early requirements engineering," *Requirements Engineering*, vol. 21, no. 1, pp. 29–61, 2016.
8. ——, "Comparison and evaluation of goal-oriented satisfaction analysis techniques," *Requirements Engineering*, vol. 18, no. 3, pp. 199–222, 2013.
9. P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Reasoning with goal models," in *International Conference on Conceptual Modeling.* Springer, 2002, pp. 167–181.