A Proposed Recommender System for Eliciting Software Sustainability Requirements

Kristin Roher, Debra Richardson
Bren School of Information and Computer Science
University of California, Irvine
Irvine, CA 92617, USA
{kroher, djr}@ics.uci.edu

Abstract – **Sustainability is not considered sufficiently in developing modern software systems. In spite of the looming threats of global climate change and environmental degradation [1], software companies are more concerned with product "time-to-market" than long-term product impacts. The research goal of this project is to overcome the barriers of incorporating sustainability into the software engineering process through the use of a recommender system to be used during requirements engineering. This system will recommend the kinds of sustainability requirements that should be considered in a given system, based on application domain, deployment locale, etc, and in so doing will lessen the workload of eliciting appropriate sustainability requirements. This research builds on an ongoing research project on Software Engineering for Sustainability.**

I. Motivation

Software-intensive systems have led to profound advances in human civilization, but at the same time they have contributed significantly to the exploitation of the Earth's resources. Moreover, software-intensive systems are deeply engaged with many different aspects of life and day-to-day activities in the industrialized world. As such, they provide a powerful leverage point for enabling sustainability concerns to be brought to bear across a wide range of domains. Developers of these systems, however, may lack a comprehensive understanding of how to integrate environmental sustainability into their software development processes [2].

The software engineering research community has realized the need to transition toward sustainability, as evidenced by efforts of the International Conference on Software Engineering (ICSE) in the past four years. The theme of ICSE 2012 was "Sustainable Software for a Sustainable World" and a related workshop was held on "Green and Sustainable Software." On the political side, the UN Millennium Development Goals and the related action plan, Agenda 21, include objectives to reverse the loss of environmental resources and employ sustainable development towards conservation and management of resources [1, 3].

It is clearly time to address the effects that software systems have on our environment and its sustainability. There are two primary reasons why such a change has not generally occurred in software development processes. For one, there is often an intellectual barrier that must be crossed; the interdisciplinary nature of sustainability has made it challenging for software developers to handle its complexity. Secondly, the extra effort and required techniques needed to consider sustainability could be more expensive up front. Even though the long-term benefits (such as aiding in the effort to decrease global climate disruption) are abundant, software companies often place more emphasis on time-to-market than on long-term product value. This paper proposes a recommender system that will assist in requirements elicitation by recommending the kinds of sustainability requirements that should be considered for a system, based on application domain, location, an organization's goals, etc. The proposed recommender system will serve as a requirements elicitation technique that aims to overcome these issues by supporting software engineers in integrating sustainability.

This paper begins by providing background on recommender systems as well as some ongoing preliminary work at the University of California, Irvine (UCIrvine) in section II. Section III describes the proposed recommender system, while Section IV provides an initial plan for evaluation. Some limitations of the approach are mentioned in section V, with concluding comments in section VI.

II. Background

A. Recommender Systems

Recommendations are often relied upon in every day life. For example, individuals rely on a recommendation from their friends on a book to read or movie to watch, and in the business realm, job recruiters rely on recommendation letters in the hiring process. Recommender systems assist and augment this natural social process to ask for advice on a given topic [4]. Recommendations made by such systems can help users navigate through large information spaces and/or provide suggestions for items to be of use to a user (such as

Amazon, Netflix, Pandora, etc.). They often guide users who do not have sufficient background to evaluate the vast number of alternatives.

The software engineering community has expressed a growing interest in the use of recommender systems. This is evidenced by the efforts of the International Workshop on Recommender Systems for Software Engineering. In addition, academic institutions (including UCIrvine) have begun offering courses on recommender systems.

Current approaches used in recommender systems include:

- *Collaborative-filtering* systems make recommendations based on the ratings or implicit behavior, such as purchase history or search patterns, of other users with similar tastes [5].
- *Content-filtering* systems make recommendations based on algorithms derived from machine learning and information retrieval to determine similar items to those that a user has expressed interest in or is currently viewing [6].
- *Context-aware* systems use either of the above approaches but also take into account environmental factors, such as time, location, or proximity of others [7].
- *Community-based* systems base recommendations on the preferences of the user's friends or colleagues [8].
- *Hybrid* systems combine the above methods with each other or potentially combined with a domain specific method [9].

In this paper, we propose a recommender system that aids developers in the elicitation of sustainability requirements for software systems. The proposed system will take a hybrid approach using mainly a context-aware approach along with content-filtering algorithms. The system will need to be aware of contextual items, such as environmental factors and project domain, as well as make recommendations based on what the user is currently viewing or preferences specified in search criteria.

B. Preliminary Work

UCIrvine's Software Engineering for Sustainability (SE4S) research group provides the groundwork for the proposed research. The group is working towards a methodology for developing software-intensive systems that *meet the functional needs of users while reducing the environmental impacts brought about by those systems* [10]. This research is focused on how software engineering can support environmental sustainability in the wide variety of domains in which software is deployed, whereas the aim of many sustainable software efforts has been just to discover how software engineering can contribute to improving the sustainability of the software itself and not its broader, potentially indirect impacts.

SE4S aims to overcome challenges similar to those that were realized in software engineering for safety and security [11, 12]. Software development practices were not equipped to deal with emerging critical concerns for safety and security and were updated accordingly. SE4S aims to make development practices for sustainability just as prevalent as those for safety and security are today, in particular by considering sustainability as a first class quality attribute.

One of the primary topics of the SE4S project is examining how to support sustainability in requirements engineering. This involves developing an adequate set of models, methods, and tools to elicit, document and analyze sustainability requirements for software systems.

III. Proposed Project

The research proposed here aims to tackle one of the challenges faced in the SE4S project, which is to promote the adoption of the SE4S approach by increasing stakeholder commitment to sustainability and also by reducing the costs and effort associated with that adoption. To accomplish this, the proposed recommender system lessens the workload of eliciting appropriate sustainability requirements by reducing the knowledge barriers associated with said elicitation. The system addresses issues of sustainability early in software development, during the requirements engineering phase. One reason for this is that problems arise when incorporating sustainability into existing systems due to the complexity involved with modifying systems to meet new requirements. Another reason is that the requirements engineering phase is the point at which developers have the most control over eventual system behavior – that is to say, requirements engineering lays the foundation for system development – and therefore is the most appropriate phase of the software life cycle to consider environmental sustainability.

We introduce the term archetype to refer to the format in which sustainability recommendations will be provided. This term originates from Philosophy [13] and is currently used in the field of Biomedical Informatics to provide a formalized Electronic Health Record (EHR) structure [14]. Because most software engineering projects will require a slightly different instantiation of the requirements, it would not be helpful to recommend specific detailed requirements. Instead the system will recommend *sustainability requirement archetypes*, which provide *a template from which requirements of the same kind are copied or on which they are based*. Using archetypes will enable a wider range of projects and project domains to be considered for each recommended sustainability aspect.

Users of the system will consist of stakeholders of a software-intensive system development project. This will include system and software engineers working on eliciting, analyzing and documenting system requirements, most of whom will have some background in requirements engineering but may have very little detailed knowledge of sustainability. Other stakeholders – such as system clients, end-users, and domain experts – are likely to be inexperienced in requirements engineering. Thus, it is important that the sustainability requirement archetypes that are recommended for consideration in the system under development be easy to understand and tailored to the domain at hand.

Users will be able to use the recommender system in two modes. The most basic use involves performing a generic request for recommendations, which will be the typical interaction by a user with no previous experience (although this mode is open to anyone). When entering a generic search, the user will be asked to specify a project domain and expected deployment locale. If the recommender system does not recognize the domain, it will ask the user to rate known project domains, based on how closely related they are to their project. The user will then be presented with several sustainability requirement archetypes to consider. A second mode of use is typical for more frequent or experienced users, who will create a profile that allows them to save recommended archetypes relevant to the types of projects they work on. Users with a profile can save projects, which can be private (for individual use), group (for use by a team), or public (for other users to learn from). Thus, profiled users will not only be able to use the system to build new requirements but also to input their own archetypes for public or personal use. In addition, profiled users will be able to input specific organizational goals that may influence recommendations.

A typical usage scenario can, for example, take place after the elicitation of an initial set of requirements:

1. User logs into their profile.
2. User selects to search based on application domain.
3. User selects a project domain (for example, e-commerce).
4. If that domain can be further narrowed, the user is asked to narrow the scope by selecting from a list. At this stage, the user is able to select as many of the options they see fit (for example, an e-commerce system used for shopping will usually contain both a database of reviews and other product information, as well as a shopping cart and online buying system).
5. The user will then be presented with a list of requirement archetypes to consider. (Continuing the above example, archetypes may include the following: "Product reviews should contain sustainability ratings, such as amount of packaging used and whether or not the packaging was reusable or recyclable," and "The shopper should be able to select packaging options when using the online checkout system").
6. The user can add all or some of the archetypes to a specific project and tailor them further during requirements elicitation.

The proposed system will also be used in re-evaluating existing software systems. It will provide options for the user to specify whether they would like to implement a new system, completely re-develop an existing system, or incorporate sustainability into an existing system without restructuring the entire project. The first two will most likely provide the same results, but the third will be a much more complex problem space. This will be further explored through a user study mentioned below.

IV. Proposed Evaluation

A. Case Studies

This section describes several exemplars that we are using as case studies in the general context of the SE4S project, and which we will also use to evaluate the proposed recommender system. The first case study will analyze ways in which sustainability can be incorporated into pre-designed and existing systems. The AquaLush irrigation system, for which the requirements documents are publicly available [15], will be used as a baseline for comparison and evaluation. Rather than modify the system implementation, we will explore ways to integrate sustainability requirements into the existing architecture.

Industrial partners will then be asked to participate in a case study that will aid them in integrating sustainability into an existing software system. The recommender system will not be used in this process, as the objective of this phase of research is to discover ways to integrate sustainability and not to evaluate the system. However, the sustainability requirement archetypes written for this case study will be entered into the recommender system so future users performing requirements elicitation can use them.

B. User Studies

Before subjecting users to the recommender system, its performance and success in giving useful recommendations will be evaluated. Sustainability requirement archetypes that the recommender should give for several different application domains (e.g. consumer electronics, e-commerce systems, video games, business applications, personal or handheld systems, etc.) will be determined. A test will then be performed on the search feature to ensure that for each domain the system produces the correct recommendations.

The first user study will consist of students using the tool, as part of developing a requirements document for the Requirements Engineering course offered at UCIrvine by Professor Debra Richardson. Based on questionnaires filled out by these student users, the usability of the interface and their overall satisfaction with the system will be evaluated.

Stakeholder willingness to follow the sustainability recommendations will be evaluated using questionnaires and interviews given to professional software engineers. They will be asked to use the recommender system in a software project setting. Data regarding the recommended sustainability archetypes that are implemented in their final project will be recorded.

The usability of the system will be measured by investigating how long it takes a user to discover a specific requirement archetype. Twenty users will be instructed on how to use the recommender system and asked to search for requirement archetypes that Amazon.com could use to make their e-commerce system more sustainable. Users will be observed in an interview lasting 10 minutes. The time it takes each user to discover five relevant recommendations will be recorded.

A study to assess the usability of the system for inexperienced users will also be performed. Twenty users will be asked to perform the same search and given the same criteria as the above-mentioned study on Amazon.com, however, the users here will not be instructed on how to use the system. This will allow for comparison to be done between the two studies.

Some domain specific questions that will be answered during our case studies to aid in articulating new archetypes for the recommender system database are listed below:

1. Was disposal of materials considered?
2. Was the time a system is left idly running considered?
3. Was system energy usage while idle considered?
4. Were sustainable material resources considered?
5. Was ability to lessen energy intensive calculations considered?
6. Does system consider sharing energy usage?
7. Were ways in which CO2 emissions could be reduced considered?
8. Was the system required to shut down at all possible times to conserve energy?
9. Was possible reuse of materials considered?

V. Limitations

One foreseeable limitation of this system is dealing with trust. Trust is a major issue when it comes to developing any recommender system. Literature suggests that users are "more likely to accept recommendations from credible sources and therefore, the credibility of the recommender system is vital to increasing the likelihood of recommendation acceptance" [9]. This system will have to rely on word of mouth, the reputations of supporting researchers, and publications that show the success of use case and user studies. In addition, the database will need to be seeded with a strong initial set of requirement archetypes before releasing the recommender system to encourage stakeholder use.

Although an evaluation plan for the system has been discussed, recommender system evaluation is not yet a structured analysis, nor has a generic metric been developed.

VI. Conclusion

This research project will advance stakeholders' ability to produce software systems that have less negative impact on our environment. In addition, it will increase their general knowledge of sustainability and related types of requirements to consider. Once implemented, the proposed recommender system will be broadly distributed as a free software program. By providing a tool to support this level of analysis, the authors will contribute to the broader goal of supporting the transition to sustainability across many different sectors within the industrialized world.

The overarching SE4S project, focused on incorporating sustainability into software development, will provide an avenue for creating awareness of the myriad of sustainability issues present in software systems. Software engineers will be provided the opportunity to support the UN's development goals for sustainable development – that is, by meeting "the needs of the present without compromising the ability of future generations to meet their own needs" [1]. They will also be able to play a key role in reducing the environmental concerns that threaten today's society.

References

[1] United Nations World Commission on Environment and Development, "Report of the World Commission on Environment and Development: Our Common Future." United Nations, 1987.

[2] B. Penzenstadler, "Towards a Definition of Sustainability in and for Software Engineering," in *Proceedings of* the *28th Annual ACM Symposium on Applied Computing*, 2013.

[3] United Nations, "Agenda 21." United Nations Conference on Environment and Development (UNCED), Jan-1992.

[4] P. Resnick and H. Varian, "Recommender Systems," *Communications of the ACM*, Vol. 40, No. 3, 1997, pp. 56-58.

[5] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," In *The Adaptive Web*, Springer, 2007, pp. 291-324. Print.

[6] P. Lops, M. Gemmis, and G. Semeraro, "Content-based Recommender Systems," In: F. Ricci et al, *Recommender Systems Handbook*, Springer, 2011, pp. 73-105. Print.

[7] G. Adomavicius and A. Tuzhilin, "Context-aware Recommender Systems," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, New York, NY, USA, 2008, pp. 335-336.

[8] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*, Springer, 2011, pp. 73-105. Print.

[9] R. Burke. "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User‐Adapted Interaction*, vol. 12, no. 4, 2002, pp. 331‐370.

[10] B. Penzenstadler, B. Tomlinson, D. Richardson, "Support Environmental Sustainability by Requirements Engineering," in *Proceedings of the International Workshop on Requirements Engineering for Sustainable Systems*, 2012.

[11] P. T. Devanbu and S. Stubblebine, "Software Engineering for Security: A Roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, New York, NY, USA, 2000, pp. 227–239.

[12] R. Lutz, "Software Engineering for Safety: A Roadmap," in *Proceedings of the Conference on The Future of Software Engineering,* New York, NY, USA, 2000, pp. 213-226.

[13] A. Stevens, *Archetype: A Natural History of the Self.* Routledge & Kegan Paul, 1982.

[14] R. Chen, G.Klein, E. Sundvall, D. Karlsson and H. Åhlfeldt, "Archetype-based conversion of EHR content models: pilot experience with a regional EHR system", in *BMC Medical Informatics and Decision Making*, 2009.

[15] C. Fox, *Introduction to Software Engineering Design: Processes, Principles and Patterns with UML2.* Addison Wesley, 2006.