Towards a Systematic Review and Classification of Collective Intelligence Systems

ANGELIKA MUSIL, JUERGEN MUSIL and STEFAN BIFFL, TU Wien, Austria
DANNY WEYNS, KU Leuven, Belgium

1. INTRODUCTION

Collective intelligence systems (CIS) access and harness the collective knowledge and work of connected people by providing a web-based environment for a user community to share, distribute and retrieve topic-specific information in an efficient way. A CIS, as *socio-technical system*, acts as a mediator of interactions among a user community and thereby provides effective, bottom-up collaboration and communication capabilities that rely on user-generated content, facilitating the aggregation and distribution of knowledge in a coordinated way. In the past, software architecture research has neglected to deepen the understanding and methodological support for architecting in CIS domain. Today, research often focuses on the engineering of technical aspects and implementations. Although research fields like human computation [Ahn 2005], collective intelligence [Malone and Bernstein 2015], crowdsourcing [Howe 2006], social machines [Miorandi et al. 2014] and social computing [Parameswaran and Whinston 2007] take into account a wider perspective by investigating the principles and synergetic effects of networked human groups and computing systems [Quinn and Bederson 2011], they do not particularly address aspects that arise from a software architecture point of view, which is although critical to design well-tailored systems. Thus the software architect needs a complete understanding about (1) the set of features all kinds of CIS have in common, and (2) existing system variants. Without this knowledge it is difficult for software architecture researchers and practitioners alike to predict the effects of design decisions on the system's capabilities and behavior.

Therefore, a systematic investigation of underlying models and mechanisms for computational support of mediated social interaction and human cognitive processes is highly relevant [Omicini and Contucci 2013] to provide a consolidated systematic knowledge base of architectural commonalities and the variations in CIS. To address these needs, this paper presents a systematic review of collective intelligence systems and reports preliminary results of a pilot study.

The remainder of the paper is structured as follows: In section 2 we present the study design of a systematic system review, in section 3 we report preliminary results of a pilot, and finally section 4 concludes and outlines paths for future work.

2. STUDY DESIGN

To better understand commonalities and in particular significant variabilities among identified key elements of CIS architectures, we plan to conduct a *Systematic System Review (SSR)* of real-world operating CIS in a wide scope. By applying a Systematic Literature Review-like approach [Kitchenham and Charters 2007], this empirically-grounded investigation focuses on the extensive survey and review of existing variants among key architecture-significant functions and features based on previously identified basic concepts, principles and characteristics of software architectures that all CIS have in common in order to explore different CIS families with an altered feature set. A focused investigation of 100+ CIS will allow to gather and evaluate available evidence from a large system collection concerning a set of characteristics in a systematic way. In addition, it will allow to identify existing CIS variations with respect to the architecture basis pattern of CIS [Musil et al. 2015], how these variations are affected by underlying system elements, processes and design decisions, and to elaborate on the strengths and weaknesses of each variation in order to better assess its potential application areas. Therefore, existing CIS solutions in various application contexts and domains are identified and closely investigated. Based on the provided empirical evidence, we plan to develop a systematic system classification model for the CIS domain.

The research method complies with a well defined and strict sequence of methodological steps, according to a previously developed protocol. This protocol follows an adapted version of the guidelines from Kitchenham and Charters [2007]. The SSR aims to address the following research questions: (1) CIS variabilities: What are architecture-significant variants among key system functions and features in CIS? (2) CIS classification: How can CIS be classified based on identified commonalities and variabilities?

In this study, we will identify a collection of potential system candidates based on our own knowledge and searches from different sources, such as the Alexa web traffic rankings[1], Wikipedia, digital libraries of scientific work, and domain experts from research and industry. The unfiltered search results need to be analyzed regarding defined inclusion and carefully assessed regarding exclusion criteria to select system candidates for their actual relevance to answer the research questions. Once the list of systems to be reviewed is finalized, a reviewer will be randomly chosen who will investigate an assigned system instance in detail and accurately record extracted and monitored information from the systems under review in a form. The draft of predefined data extraction forms with criteria allows to survey each system objectively and thus reduces bias. The data are collected by examining the input to the system and the output of the system from a user's perspective. By this means we can derive information about a system's functionalities, features, capabilities, data structures, workflows, functional organization, and organizational structures. The following synthesis and analysis of the extracted data aim to identify existing variations of key functions and features across different CIS. To better understand the effects of a variation selection, it is important to record observed variances in the interactions between human and system as well as in the system's behavior and environment. In addition, the identified variabilities would allow a clustering of similar CIS. Based on the identified commonalities and variabilities we will focus on the development of dimensions to classify CIS. The goal of such a classification is to provide an overview of the space of existing system sub-families, their underlying key features that these groupings share, and the visualization of associations between them. Further, it allows the documentation of identified deviations from the basis system pattern leading to more differentiated and specific pattern variants. The classification helps software architects, who are not familiar with the CIS domain, to get a working understanding of the key features of different classes of CIS and their relationships.

3. PRELIMINARY RESULTS AND DISCUSSION

We conducted an initial pilot study where we reviewed 20 different CIS in total[2]. We used web-traffic rankings from Alexa to identify potentially interesting and dominating platforms. We looked at interaction sequences and workflows that are supported by the systems, and also at artifacts that have been created as a result of these interactions (e.g. notification / digest emails). In addition, we also reviewed available documentation of each system such as user guides and technical documentation.

[1] http://www.alexa.com/ (last visited at 02/07/2016)
[2] An older pilot with a smaller sample was reported by Musil et al. [2014].

In the end we identified six key features of CIS which are:

(1) Ability of any actor to add new content to domain items,
(2) Ability of any actor to contribute content to parts of domain items of an other actor, thus change its state,
(3) Ability of actors to create system-internal links to connect domain items,
(4) Dissemination of state changes of selected domain items to actors, thus traceable for all actors,
(5) User-driven recommender system,
(6) Support for tracking of usage behavior of a single actor.

First results indicate three groups of capabilities, which are (1) domain item manipulation, (2) dissemination and (3) monitoring and analysis. Domain item manipulation includes features 1-3 and comprises the creation of new items, the (partial) manipulation of domain items by other actors, and the support of linking items together, using system-internal links. Preliminary results indicate that the support of item linkability might be of particular relevance of CIS architectures. Another important feature is the creation of new items by external actors, which is also common to other established social web applications like internet forums, mailing lists, and version control systems. Thereby relevant is the authorship of the item which is stored in a particular data structure that acts as a proxy for the ownership relationship between the actor and the particular domain item. The ownership relationship defines who is the owner of an item and thus who has extensive control to decide (1) to which extent other actors are able to contribute to the item, and (2) if contributions comply to predefined quality requirements. The group of dissemination consists of feature 4-5 and focuses on the distribution of changes to the actor base to keep them aware about updates of relevant domain items and ongoing actor activities in order to stimulate subsequent contributions of the actor to the platform. The third group is monitoring and analysis and consists of feature 6, which focuses on traceability of actor activities and item manipulation activities. Results generated from monitoring and analysis are important input to the dissemination feature group. Preliminary results show that the inspected features have been consistently found in the groups of social networking services (Facebook, Twitter), wikis (Wikipedia, Confluence), media/content sharing platforms (YouTube, Flickr), marketplaces (eBay), review and recommendation sharing platforms (Yelp, TripAdvisor), and knowledge markets (Stack Overflow).

4. CONCLUSION AND FUTURE WORK

The systematic system review presented in this paper is an important step towards gaining new knowledge about commonalities and variations in collective intelligence systems. Although the preliminary results are promising, future work is needed to conduct the survey with a large system sample. In addition to gaining a deeper understanding of CIS, we expect that the systematic system review approach has also the side-effect of being potentially useful in the systematic investigation of systems in other domains.

ACKNOWLEDGMENT

This work was supported by the Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy and Science, the Austrian National Foundation for Research, Technology and Development, and TU Wien research funds.

REFERENCES

Luis Von Ahn. 2005. *Human Computation*. Ph.D. Dissertation. Carnegie Mellon University, Pittsburgh. http://ra.adm.cs.cmu.edu/anon/usr/ftp/2005/CMU-CS-05-193.pdf

Jeff Howe. 2006. The Rise of Crowdsourcing. *Wired* 14, 6 (jun 2006). http://www.wired.com/wired/archive/14.06/crowds.html

Barbara A. Kitchenham and Stuart Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report Version 2.3. EBSE Technical Report, EBSE-2007-01, Software Engineering Group, School of Computer Science and Mathematics, Keele University, UK and Department of Computer Science, University of Durham, UK. https://www.cs.auckland.ac.nz/~norsaremah/2007GuidelinesforperformingSLRinSEv2.3.pdf

Thomas W. Malone and Michael S. Bernstein (Eds.). 2015. *Handbook of Collective Intelligence*. MIT Press.

Daniele Miorandi, Vincenzo Maltese, Michael Rovatsos, Anton Nijholt, and James Stewart (Eds.). 2014. *Social Collective Intelligence: Combining the Powers of Humans and Machines to Build a Smarter Society*. Springer International Publishing.

Juergen Musil, Angelika Musil, and Stefan Biffl. 2014. Towards a Coordination-Centric Architecture Metamodel for Social Web Applications. In *Proc. of the 8th European Conference on Software Architecture (ECSA '14) (LNCS 8627)*, Paris Avgeriou and Uwe Zdun (Eds.), Vol. 8627. Springer International Publishing, 106–113.

Juergen Musil, Angelika Musil, and Stefan Biffl. 2015. SIS: An Architecture Pattern for Collective Intelligence Systems. In *Proc. of the 20th European Conference on Pattern Languages of Programs (EuroPLoP '15)*. ACM, 20:1–20:12.

Andrea Omicini and Pierluigi Contucci. 2013. Complexity and Interaction: Blurring Borders between Physical, Computational, and Social Systems. Preliminary Notes. In *Proc. of the 5th International Conference on Computational Collective Intelligence Technologies and Applications (ICCCI '13) (LNCS)*, Costin Badica, Ngoc Thanh Nguyen, and Marius Brezovan (Eds.), Vol. 8083. Springer Berlin Heidelberg, 1–10.

Manoj Parameswaran and Andrew B. Whinston. 2007. Social Computing: An Overview. *Communications of the Association for Information Systems (CAIS)* 19, 37 (2007), 762–780.

Alexander J. Quinn and Benjamin B. Bederson. 2011. Human Computation: A Survey and Taxonomy of a Growing Field. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, 1403–1412.