Software Architecture Principles of Self-Organizational Collective Intelligence Systems

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

1. INTRODUCTION

Collective intelligence systems (CIS) like wikis, social networking services and content sharing platforms have dramatically improved knowledge creation processes with more effective information aggregation and dissemination to benefit human collaboration, peer-production and self-organization [Bigham et al. 2015]. Despite the wide adoption of CIS, there remains a lack of consolidated systematic knowledge of the architectural principles and practices [Convertino et al. 2010]. From our experience with industry partners, we learned that software architects lack guidance to design CIS for the application context of individual organizations. Thus software architects resort to reproduce designs from similar successful CIS by "trial and error" or "clone and own", but without understanding their underlying principles, mechanisms and rationales. Therefore, software architects are severely impacted in the anticipation of unintentional side effects that are caused by their architectural design decisions on the system's core capabilities.

To address the problems of engineering CIS, this paper contributes results on an architecture pattern and an architecture framework for collective intelligence systems. In particular, two research questions should be addressed: (RQ1) What are the most important underlying architectural principles of collective intelligence systems? (RQ2) How can we codify (capture, document, structure, organize) these architectural principles to make them useful for engineering collective intelligence systems? In order to answer these research questions we followed a three phase approach. First, we collected knowledge from various sources, and second, we synthesized the knowledge and defined the pattern as well as the architecture framework for realizing CIS. Third, we evaluated the framework in two industry cases where CIS have been designed and implemented using our framework.

The remainder of this paper is structured as follows: Section 2 describes the architecture pattern and the architecture framework, section 3 reports results from the multi-phase research method, and finally section 4 draws conclusions and outlines future research.

2. ARCHITECTURAL PRINCIPLES OF COLLECTIVE INTELLIGENCE SYSTEMS

The consolidated knowledge we acquired in the Collect phase of our research allowed us to identify commonly occurring principles of architectural structures in CIS and to describe them in form of an architecture pattern. Building upon these architectural principles and the pattern, we developed and documented a novel architecture framework for realizing CIS which follows the ISO/IEC/IEEE 42010 standard for architecture descriptions [ISO/IEC/IEEE 42010 2011]. For a detailed description of these works we refer the interested reader to [Musil et al. 2015a] and [Musil et al. 2015b].

2.1 Stigmergic Information System Architecture Pattern

Based on the identified CIS key architectural principles, the contributed *Stigmergic Information System (SIS) architecture pattern* is an attempt to incorporate the essence of these systems in a minimal system description including the common elements and processes of a CIS. Central to the SIS pattern is the *stigmergic process* which facilitates *self-organizational, environment-mediated coordination* [Heylighen 2015]. Stigmergy is a nature-inspired coordination mechanism that was originally used to describe self-organizational, environment-mediated task coordination of social insects [Bonabeau et al. 1999; Zambonelli 2015]. In the context of CIS, the stigmergic process creates a *perpetual, positive feedback loop* between a human actor base and the reactive coordination environment. This feedback loop consists of two alternating process phases: (1) the *aggregation phase* where actors are able to create/modify user-generated content stored in the CI artifacts, and (2) the *dissemination phase* where the CIS uses active and passive dissemination mechanisms based on the contributed content to effectively share information, to promote awareness among the actors about activities of others as well as to eventually trigger reactions by the actor base. This continuous flow of actor contributions within the system environment enables the emergence of collective intelligence. The pattern describes a system architecture of a hybrid human-computer system where: (1) *human actors* collectively create and share knowledge by independently performing *activities* on so-called *CI artifacts* as well as *linking* them, and (2) thereby enables the bottom-up building of a virtual *artifact network* (3) that is managed by a *reactive/adaptive computing infrastructure* which enforces different *rules* for content analysis, filtering and dissemination of knowledge among the actor base. The pattern does not address the technical specifics of the implementation of a CIS, since depending on the architectural concerns there may be multiple ways to implement the pattern within a system-of-interest.

2.2 Architecture Framework for Collective Intelligence Systems

The *architecture framework for collective intelligence systems (CIS-AF)* is developed as a methodology and support for software architects to efficiently describe the core elements of a CIS architecture, which are documented in the SIS pattern, without being limited in its technical implementation. The framework defines a set of three complementary architecture viewpoints for designing new CIS solutions that cover the essential concerns of stakeholders.

1. CI Context Viewpoint addresses the *usefulness* and *perpetuality* concerns of software architects, platform owners and actors related to the basic design of CIS-specific capabilities. It defines three model kinds, *As-Is Workflow*, *Stigmergic Coordination*, and *To-Be Workflow*, that support capturing relevant architectural design decisions to achieve the essential bottom-up information aggregation, management and distribution capabilities for hard-to-access dispersed knowledge and information.

2. CI Technical Realization Viewpoint addresses the *data aggregation*, *knowledge dissemination* and *interactivity* concerns of software architects, platform owners, platform builders and actors, and provides a more detailed perspective on the realization of the CIS and its specific capabilities. It defines three model kinds, *Artifact Definition*, *Aggregation*, and *Dissemination*, to capture the details about the collective knowledge and the structure of the CI artifacts, the aggregation of data and the stigmergy-based dissemination of knowledge. The models show the relevant architectural information that is essential to guide the concrete implementation of a new CIS.

3. CI Operation Viewpoint addresses the *kickstart* and *monitoring* concerns of managers and analysts related to the startup of the CIS operation. It defines two model kinds, *Initial Content Acquisition* and *CI Analytics*, to identify initial content and actor groups and specify indicators to measure CIS aggregation and dissemination performance. The models show the relevant architectural information that is essential to guide a successful startup of the perpetual feedback loop of a new CIS.

3. RESEARCH APPROACH & EVALUATION

We applied an iterative research approach in 3 phases. In the first phase (collect phase) we identified the characteristic core elements and processes of CIS, key stakeholders and their architecture-related concerns, and we elicited model needs. This knowledge was acquired by conducting (1) a survey of existing CIS, (2) focus workshops with architects and business people, (3) semi-structured interviews with developers, and by developing (4) a pilot CIS. In the second phase (synthesize phase) we consolidated the insights and input we collected in the first phase and formalized the architectural knowledge using established methodologies as architecture pattern and architecture framework. In the third phase (evaluation phase) we performed a qualitative evaluation of the framework in two industrial cases.

3.1 Collect Phase

To identify the underlying architectural principles of CIS, we performed a survey of existing CIS. In total, we identified 180 potentially interesting and popular CIS and analyzed 30 of them in depth. In addition, we organized three focus workshops with stakeholders from industry and different domains that had an interest in introducing CIS in their organizations. Main findings include the identification of key concerns, including usefulness of CIS, aggregation and dissemination of knowledge, perpetuality, kick-start of the CIS, and monitoring the system in operation. Furthermore, we conducted 10 semi-structured interviews with software developers, which revealed important issues of developers regarding understanding CIS principles and how to implement them. Identified challenges are the lack of understanding of the stigmergic feedback loop process and its implementation as the key challenge for the development of CIS. Finally, we developed a pilot CIS to cross-check the collected knowledge. We learned that it is essential to identify the process improvements and to consider the feedback mechanism of CIS from the outset, as this mechanism is the central factor of the benefit and success of a CIS. We also learned that aggregation and dissemination should be considered as first-class citizens and requires analysis, a workflow, and well-defined stimuli. Bootstrapping a CIS is a dynamic process that takes time and thus building initial content and monitoring system behavior after deployment is highly important.

3.2 Evaluation Phase

We performed a qualitative evaluation of the CIS-AF in two industry cases where CIS have been designed and implemented using the framework. In each case we observed a team of two architects over a period of 12 months to observe how they use the framework to create an architecture for a CIS. These observations include (1) regular communication with status reports, experiences and feedback, (2) stakeholder workshops, and (3) regular evaluations of the created designs. The CIS in the first case is a software reuse platform and the second case is the Feature Deliberatorium, which enables the collective feature reviewing and consolidation in an organization-internal software ecosystem. Results of these two cases showed that the framework effectively supports stakeholders with capturing their CIS-specific concerns and establishing CIS is their organizations. In particular, the evaluation demonstrated that the framework offers a shared vocabulary of CIS concepts to the stakeholders, it guides them to systematically apply the stigmergic principles of CIS, and it supports them with kickstarting the CIS in their organizations. For a detailed description we refer the interested reader to [Musil et al. 2015b].

4. CONCLUSION AND FUTURE WORK

The presented architecture pattern and architecture framework provide a systematic approach to guide the design of CIS and to effectively educate architects about basic elements and specifics of CIS. It is important to note that the presented approach is implementation-agnostic and so it does not prescribe a particular technology or platform. Future research aims to provide tool support for the CIS-AF as well as to extend the framework towards concerns like user motivation/engagement, privacy, qualitative growth of content, and CIS lifecycle and evolution.

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

Jeffrey P. Bigham, Michael S. Bernstein, and Eytan Adar. 2015. Human-Computer Interaction and Collective Intelligence. In *Handbook of Collective Intelligence*, Thomas W. Malone and Michael S. Bernstein (Eds.). MIT Press, 57–83.

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.

Gregorio Convertino, Antonietta Grasso, Joan DiMicco, Giorgio De Michelis, and Ed H. Chi. 2010. Collective Intelligence In Organizations: Toward a Research Agenda. In *Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW '10)*. ACM, 613–614.

Francis Heylighen. 2015. Stigmergy as a Universal Coordination Mechanism: components, varieties and application. In *Human Stigmergy: Theoretical Developments and New Applications*, T. Lewis and L. Marsh (Eds.). Springer. Available: http://pespmc1.vub.ac.be/papers/stigmergy-varieties.pdf (last visited at 02/07/2016).

ISO/IEC/IEEE 42010. 2011. *Systems and Software Engineering - Architecture Description*. http://www.iso-architecture.org/42010/

Juergen Musil, Angelika Musil, and Stefan Biffl. 2015a. 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.

Juergen Musil, Angelika Musil, Danny Weyns, and Stefan Biffl. 2015b. An Architecture Framework for Collective Intelligence Systems. In *Proc. of the 12th Working IEEE/IFIP Conference on Software Architecture (WICSA '15)*. IEEE, 21–30.

Franco Zambonelli. 2015. Engineering self-organizing urban superorganisms. *Engineering Applications of Artificial Intelligence* 41 (2015), 325–332.