Towards a Coordination-Centric Architecture Metamodel for Social Web Applications

Juergen Musil[1], Angelika Musil[1], and Stefan Biffl[2]

CDL-Flex, Institute of Software Technology and Interactive Systems,
Vienna University of Technology,
Favoritenstrasse 9/188, 1040 Vienna, Austria
[1]{jmusil,angelika}@computer.org
[2]stefan.biffl@tuwien.ac.at

Abstract. Social web applications like wikis, social networks, and crowdsourcing markets have provided people with new dynamic forms of communication and collaboration. Although communities have widely adopted these systems, the methodological support for their architecting is still at the beginning. Since social web applications are mediation environments for human interaction, environment-based coordination models like stigmergy have increased in relevance. Therefore, we propose the concept of a Stigmergic Information System (SIS) architecture metamodel, which embeds a stigmergy-like coordination model. The metamodel defines key system elements and organizes a system into four layers: agent, artifact data, analysis & control and workflow. The metamodel should support the systematic investigation of common architecture elements, their relations and interdependencies, and future approaches for the description and modeling of social web applications. In this work we introduce the SIS architecture metamodel and evaluate the metamodel's validity with preliminary results from a pilot survey on groupware systems.

Keywords: Architecture Metamodel, Collective Intelligence, Coordination, Social Web Application, Stigmergic Information System, Stigmergy

1 Introduction

Over the last decade, new forms of online collaboration platforms like wikis, social networks and crowdsourcing markets have enabled individuals to communicate and work together on problems effectively. While social web applications have been widely adopted in a variety of domains, the understanding and methodological support for architecting and "programming" them on a higher, more abstract, system level is still at an early stage [4], [8]. Social web applications mediate the interaction among their users by realizing a certain coordination model. Thus modifications of the coordination model highly affect a social web application's main regulatory capabilities. Therefore, research which investigates the models and mechanisms for computational support of mediated social interaction and human cognitive processes is highly relevant as well

as approaches, which enable the systematic design and analysis of these socio-technical systems [13], [16].
In this paper we explore the concept of a Stigmergic Information System (SIS) architecture metamodel, which realizes stigmergy-like coordination and self-organization and which also covers common key features of popular social web applications. Stigmergy (from Greek *stigma*: sign, and *ergon*: work) is a nature-inspired coordination mechanism to describe the environment-mediated task coordination of social insects [2]. Stigmergy promotes *awareness* among agents about the activities of other agents, which in turn reinforces their own activities [16]. In computer science, stigmergy is well-known as an effective coordination model, which provides computational systems with bottom-up, environment-mediated coordination capabilities [1], [12], [16]. The SIS metamodel represents a first step towards a common system model on which basis architectures for social web applications can be designed. Furthermore, the metamodel should also assist in the identification of design patterns, thus support architecture decision making and tradeoff considerations.
The remainder of this paper summarizes related work in section 2 and the research question and approach in 3. The architecture metamodel concept is detailed in section 4 and section 5 discusses preliminary results from the pilot survey. Finally, section 6 concludes and outlines future work.

2 Related Work

This section presents an overview on architecting coordination in social web applications and the stigmergy model for self-organizational, environment-mediated coordination.
The challenge of architecting social web applications is well known: In 2001, Tiwana and Bush [17] presented with the KNOWeb architecture one of the first approaches, which uses positive feedback mechanisms to deliberately reinforce the social/knowledge exchange in distributed virtual communities. Girgensohn and Lee [7] described their experiences from designing two "social interaction web sites" for two social groups. Similar to Tiwana and Bush, they concluded that in order to retain user engagement (1) the role of the social software system as a merely supportive infrastructure is not sufficient, and (2) mechanisms to maintain a continuous influx of new user contributions are needed. In recent time, Dorn and Taylor [4] presented a human Architecture Description Language hADL to describe collaboration structures and patterns in social web applications. Minder and Bernstein [8] focused on human computation and propose with CrowdLang a programming framework for interaction mechanisms and the design of human computation systems.
Software architectures are known, besides coordination languages, to be the primary means to embed coordination models [3]. When using a computational system to coordinate a multi-agent system (MAS) through stigmergy, the concepts of environment and artifact are essential [14], [16]. Weyns et al. [19] noted on the environment that it *"mediates both the interaction among agents and*

the access to resources". The *artifact* is used as a coordination medium, as an environment abstraction, through which the agents communicate. Advantages of environment-based coordination approaches are that processes can be decoupled in space and time as well as that producer and consumer can be anonymous [15]. For social web applications, stigmergy is of particular relevance, since the interaction between the human agents is predominantly mediated/regulated by the software infrastructure [13]. Parunak [18] surveyed stigmergic computational systems, which are used to coordinate human interactions. So far, some types of social web applications (social networking services, wikis) have been identified as stigmergic systems [13], [16], [18].

3 Research Question and Approach

The research question of this work is to **explore the possibility of a hypothetical metamodel with a built-in coordination mechanism, which is capable to cover common key features of dominant social web application types**.

We follow best-practice processes from software architecture discovery and reconstruction (SAR) to derive and validate a conceptual architecture metamodel. The metamodel should support the research for a future architecture viewpoint in order to assist software architects in the description and modeling of social web applications. We have chosen a hybrid bottom-up and top-down process as described by Ducasse and Pollet [5], which follows a metamodel focus like the CacOphoNy approach introduced by Favre [6]. Favre's approach has been deemed promising by Ducsasse and Pollet as it focusses on different abstraction levels horizontally and vertically.

We proceed in three phases: (1) design of a hypothetical architecture metamodel based on literature and experience from industry, (2) derivation of a catalogue of key features using the metamodel and formal concept analysis method, and (3) top-down exploration of the architecture hypothesis' validity in an initial pilot survey by mapping model constructs to features from systems from the field. Insights from the pilot should support the design of a following large-scale system survey.

4 The Stigmergic Information System (SIS) Architecture Metamodel

This section presents the Stigmergic Information System (SIS) architecture metamodel. An initial description of the SIS approach has been presented in [9], where the overall system concept and its key areas have been outlined. Further a simplified subset of metamodel elements has been described in [10]. This work extends previous research by contributing (a) a coherent, hypothetical architecture metamodel, and (b) a set of key features, with which systems can be tested for compliance with the proposed metamodel. A detailed description of the metamodel and its elements can be found online in a technical report [11].

The SIS metamodel is organized in four layers: I. agent layer, II. artifact data layer, III. analysis & control layer and IV. workflow layer (see Fig. 1). Human agents in layer I provide a continuous stream of information, whereby layer II and III form the computational coordination infrastructure, which maintains and enforces the workflows from layer IV.

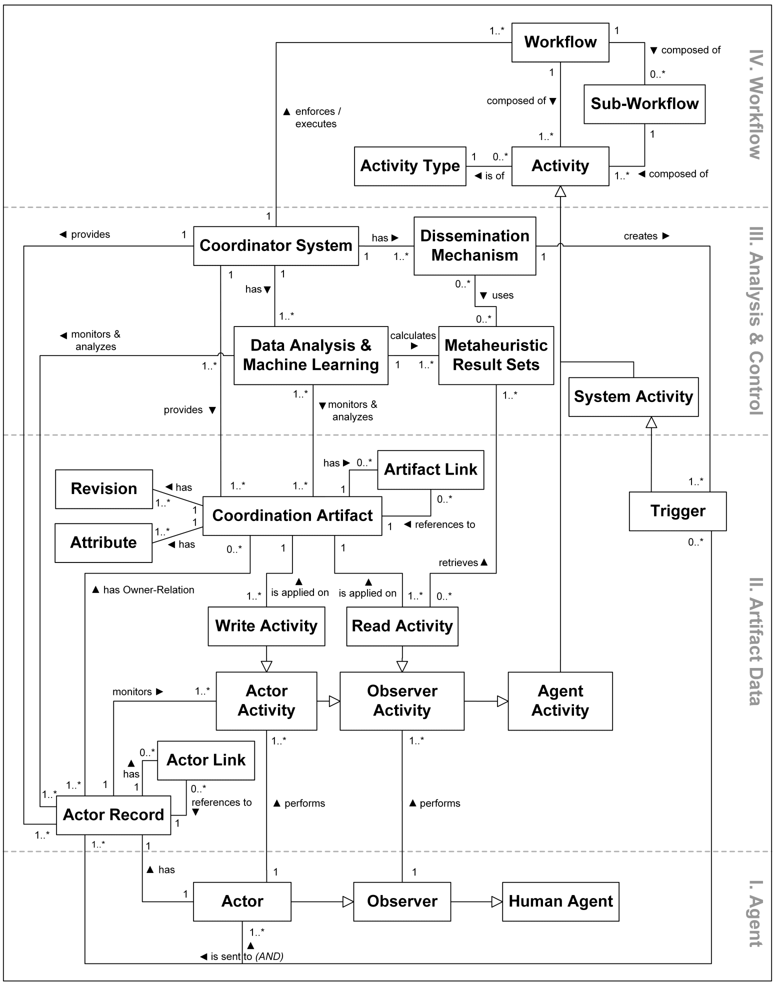

Fig. 1. UML class diagram of the hypothetical architecture metamodel for Stigmergic Information Systems.

I. Agent Layer: The agent layer encompasses types of *human agents*, who interact with the system and are an active component in a SIS. Human agents are divided into *observers*, who have read-only access to the artifact content, and *actors*, who can also create artifacts and modify their content. Typically the actor role requires an agent to sign in with some sort of user account in the system.

II. Artifact Data Layer: The artifact data layer is the first coordination tier and consists of the coordination artifacts and the actor records. A *coordination artifact* (CA) is a characteristic tuple of *attributes*, which is the same for all CAs within a SIS. The coordination artifacts are the passive components in a SIS and store actor contributions, whereby actors can only modify the values of the attributes, but not the attribute configuration of the tuple itself (e.g., a wiki user can edit an article page, but she cannot modify the article page's data model). Also, CAs can be linked by actors via *artifact links*, which can be direct via uni-/bi-directional links or indirect by joins of tags or categories. Each actor has her own *actor record* (AR) that logs an actor's activities within the SIS. Activities, logged by the AR, are for example all artifact activities, logins, page views, and clicks on trace links in notification messages.

III. Analysis & Control Layer: The analysis & control layer is the second coordination tier and hosts the coordinator system and the subsystems for data analysis and machine learning. Different to typical computational stigmergic systems, where the active component is represented exclusively by autonomous agents interacting through a passive environment, a SIS has with the *coordinator system* an additional adaptive component, that reacts to changes in the CAs. The information from the data analysis is the basis for the *machine learning* subsystem, which uses dissemination mechanisms to create *stimuli/trigger* for the actor base, based on artifact activities and according to defined workflows from level IV. *Dissemination mechanisms* make the agents, in particular actors, *aware* about ongoing activities in the artifact network and motivate them to contribute to an artifact, whereby a contribution of one actor should trigger contributions of other actors and so on. It can be discriminated between pull-based and push-based mechanisms.

IV. Workflow Layer: The workflow layer is the third coordination tier and defines the rules to orchestrate the layers below. Workflows are defined by the SIS platform provider and composed of at least one *activity* performed by an agent or the system. The workflow layer is conceptually responsible for maintaining the perpetual feedback loop between agent base (layer I) and coordination infrastructure (layer II + III) and to improve SIS utility for the agent base.

5 Preliminary Results and Discussion

To evaluate the SIS metamodel's validity and scope, we conducted a pilot survey of 14 space/time-asynchronous groupware systems. Where possible, dominant systems with high Alexa[1] web-traffic rankings have been chosen. We examine

[1] `www.alexa.com` (last visited at 06/18/2014)

6 characteristic features, which we derived from the metamodel and a concept lattice using formal concept analysis method. Features 1-4 focus on capabilities of the individual coordination artifact and features 5-6 address data analytics and tracking capabilities on the system level. The following key features of a SIS have been identified:

1. Can any actor add a new coordination artifact?[2]
2. Can any actor contribute to parts of the coordination artifact of an other actor, thus change its state?
3. Are actors able to create system-internal links to connect coordination artifacts?[3]
4. Are state changes of selected artifacts traceable for all actors and/or forwarded to them (via dissemination mechanisms)?
5. Does the system have a user-driven recommender system?
6. Does the system keep track about the usage behavior of a single actor?[4]

A system has to meet all 6 features in order to comply with the SIS metamodel. The pilot results (see Table 1) show that the inspected features have been consistently found in the groups of social networking services, wikis, media sharing, marketplace, review and recommendation sharing, crowdsourcing and knowledge markets. All of the compliant systems are instances of modern social web applications. A feature, which should be observed in more detail in future research and which is unique to SIS-conform social web applications, is the linkability of artifacts using system-internal links. Although the terms 'folksonomies', tagging, and social graph have become buzz words in the last decade of web applications, preliminary results indicate that this feature is indeed pivotal. Another important feature is the creation of new artifacts by external users, which is also common to other established groupware systems like internet forums, mailing lists, version control systems and BitTorrent trackers.

6 Conclusions and Future Work

This work introduced a hypothetical architecture metamodel of social web applications, which embeds a stigmergy-like coordination model. The SIS metamodel defines key elements and their relations and organizes a system in the four layers of agent, artifact data, analysis & control and workflow. In a pilot survey we explored the metamodel's validity for various types of groupware and social web applications with a set of 6 characteristic key features derived from the metamodel. Results of the pilot study indicate that the metamodel is capable of describing certain types of social web applications and substantiate the hypothesis that a coordination-centric perspective like the SIS metamodel has the

[2] Access restrictions (password wall, pay wall, etc.) are not an exclusion criteria as long as they affect all actors in the same way.

[3] Examples are the friend-relationship in Facebook or Wikilinks in Wikipedia (`http://en.wikipedia.org/wiki/Help:Link#Wikilinks`) (last visited at 06/18/2014).

[4] Client-side tracking of usage behavior via cookies is not sufficient.

Table 1. Features mapped to representative groupware instances.

Category	Example (SW-Tech)	1. Art	2. Ctb	3. Lnk	4. Dsm	5. Rmd	6. Trk
Internet Forum	forums.debian.net (phpBB)	✓	✓		✓		✓
Mailing List	Apache Software Foundation Mailing Lists (ezmlm)	✓	✓		✓		✓
Blog	mashable.com (WordPress)		✓		✓	✓	
Version Control System	git.kernel.org (Git)	✓					✓
CMS	huffingtonpost.com (Moveable Type)		✓		✓	✓	
BitTorrent Tracker	OpenBitTorrent.com (Opentracker)	✓					
Online Booking System	expedia.com		✓		✓	✓	
Social Networking Service	Facebook	✓	✓	✓	✓	✓	✓
Wiki	Wikipedia	✓	✓	✓	✓	✓	✓
Media/Content Sharing	YouTube	✓	✓	✓	✓	✓	✓
Marketplace	eBay	✓	✓	✓	✓	✓	✓
Review & Recommendation Sharing	Yelp	✓	✓	✓	✓	✓	✓
Crowdsourcing	InnoCentive	✓	✓	✓	✓	✓	✓
Knowledge Market	Stack Exchange	✓	✓	✓	✓	✓	✓

potential to provide a wider and more detailed viewpoint of the system.
For future work the following steps are planned: (1) Interviews with software architects to get feedback on the metamodel for soundness and further refinement. (2) A quantitative, comprehensive survey of systems from the field to evaluate the metamodel's validity and the identified key features, as well as to investigate commonalities and variations in features. (3) An architecture analysis of a representative social web application to map metamodel elements to system elements. Though it takes extensive validation in multiple steps to conclusively evaluate a metamodel that covers such a broad field, we see it as a promising architectural research agenda in the time of socio-technical platforms and networked societies.

Acknowledgments. This work was supported by the Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development, Austria.

References

1. Babaoglu, O. et al.: Design patterns from biology for distributed computing. ACM Trans. Autonomous and Adaptive Systems 1(1), 26–66 (2006)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
3. Ciancarini, P.: Coordination Models and Languages as Software Integrators. ACM Computing Surveys 28(2), 300–302 (1996)

4. Dorn, C., Taylor, R. N.: ZIB Structure Prediction Pipeline: Architecture-Driven Modeling of Adaptive Collaboration Structures in Large-Scale Social Web Applications. In: Wang, X.S., Cruz, I., Delis, A., Huang, G. (eds.) Web Information Systems Engineering - WISE 2012. LNCS, vol. 7651, pp. 143–156. Springer, Heidelberg (2012)
5. Ducasse, S., Pollet, D.: Software Architecture Reconstruction: A Process-Oriented Taxonomy. IEEE Trans. Software Engineering 35(4), 573–591 (2009)
6. Favre, J.-M.: CacOphoNy: Metamodel-Driven Architecture Recovery. In: Proc. 11th Working Conf. on Reverse Engineering (WCRE '04), pp. 204–213. IEEE CS (2004)
7. Girgensohn, A., Lee, A.: Making Web Sites Be Places for Social Interaction. In: Proc. ACM Conf. on Computer Supported Cooperative Work (CSCW '02), pp. 136–145. ACM (2002)
8. Minder, P., Bernstein, A.: CrowdLang: A Programming Language for the Systematic Exploration of Human Computation Systems. In: Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., Guéret, C. (eds.) Social Informatics. LNCS, vol. 7710, pp. 124–137. Springer, Heidelberg (2012)
9. Musil, J., Musil, A., Winkler, D., Biffl, S.: A First Account on Stigmergic Information Systems and Their Impact on Platform Development. In: Proc. WICSA/ECSA 2012 Companion Volume (WICSA/ECSA '12), pp. 69–73. ACM (2012)
10. Musil, J., Musil, A., Biffl, S.: Elements of Software Ecosystem Early-Stage Design for Collective Intelligence Systems. In: Proc. Int'l Workshop on Ecosystem Architectures (WEA '13), pp. 21–25. ACM (2013)
11. Musil, J., Musil, A., Biffl, S.: Stigmergic Information Systems - Part 1: An Architecture Metamodel for Collective Intelligence Systems. Technical report, IFS-CDL 14-40, Vienna University of Technology (August 2014) `http://qse.ifs.tuwien.ac.at/publication/IFS-CDL-14-40.pdf`
12. Omicini, A.: Nature-Inspired Coordination Models: Current Status and Future Trends. ISRN Software Engineering 2013 (2013)
13. Omicini, A., Contucci, P.: Complexity and Interaction: Blurring Borders between Physical, Computational, and Social Systems. Preliminary Notes. In: Bădică, C., Nguyen, N.T., Brezovan, M. (eds.) Computational Collective Intelligence. Technologies and Applications. LNCS, vol. 8083, pp. 1–10. Springer, Heidelberg (2013)
14. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A Meta-model for Multi-agent Systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
15. Papadopoulos, G.A., Arbab, F.: Coordination Models and Languages. Advances in Computers 46, 329–400 (1998)
16. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive Stigmergy: Towards a Framework Based on Agents and Artifacts. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) Environments for Multi-Agent Systems III. LNCS, vol. 4389, pp. 124–140. Springer, Heidelberg (2007)
17. Tiwana, A., Bush, A.: A social exchange architecture for distributed Web communities. Journal of Knowledge Management 5(3), 242–249 (2001)
18. Van Dyke Parunak, H.: A Survey of Environments and Mechanisms for Human-Human Stigmergy. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) Environments for Multi-Agent Systems II. LNCS, vol. 3830, pp. 163–186. Springer, Heidelberg (2006)
19. Weyns, D., Omicini, A., Odell, J.: Environment as a first class abstraction in multiagent systems. Autonomous Agents and Multi-Agent Systems 14(1), 5–30 (2007)