SIS: An Architecture Pattern for Collective Intelligence Systems

JUERGEN MUSIL, ANGELIKA MUSIL and STEFAN BIFFL,
Institute of Software Technology and Interactive Systems, Vienna University of Technology

Collective intelligence systems (CIS) like social networking services, wikis and social media platforms have influenced the way how people create knowledge and share information. An important aspect of these systems is how they mediate interaction and coordination among their user base. In this paper we report the STIGMERGIC INFORMATION SYSTEM (SIS) architecture pattern that can be used by software architects to describe a wide range of CIS on a systemic level. The SIS pattern builds upon the nature-inspired coordination mechanism of stigmergy, which enables indirect communication of agents via traces in the environment. The SIS pattern describes a hybrid human-computer system where humans collectively create and share knowledge and thereby a virtual artifact network is growing that is managed by a reactive/adaptive computing infrastructure. The SIS pattern aims to support software architects to take into account the system design of CIS from an implementation-agnostic perspective.

Categories and Subject Descriptors: D.2.11 [**Software Engineering**]: Software Architectures—*Patterns*; *Domain-specific architectures*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Current awareness systems*; *Information networks*

General Terms: Design, Human factors

Additional Key Words and Phrases: Awareness, collective intelligence, coordination, social web application, software architecture pattern, stigmergic information system, stigmergy

ACM Reference Format:

Musil, J., Musil, A. and Biffl, S. 2015. SIS: An Architecture Pattern for Collective Intelligence Systems. jn V, EuroPLoP '15: Proceedings of the 20th European Conference on Pattern Languages of Programs, Article 20 (July 2015), 12 pages.

1. INTRODUCTION

Since more than a decade, collective intelligence systems (CIS) like social networking services, wikis and social media platforms have become an important type of software systems. Initially the advent of CIS began as social web applications during the Web 2.0 phase with the movement of "architectures of participation" [O'Reilly 2007] and associated terms like crowdsourcing, folksonomies and "the wisdom of crowds" [Surowiecki 2005]. A key characteristic of CIS is that they allow users the collective, bottom-up creation and modification of a shared information network, which is focused around a certain topic. CIS experienced world-wide adoption and diversification in a

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.
Author's address: J. Musil, A. Musil, S. Biffl, Institute of Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstrasse 9/188, 1040 Vienna, Austria; email: {jmusil, angelika}@computer.org, stefan.biffl@tuwien.ac.at

EuroPLoP '15, July 08 - 12, 2015, Kaufbeuren, Germany

ACM 978-1-4503-3847-9/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2855321.2855342

variety of application domains, their ranking is among today's top web traffic generators.

In the current state, there is a variety of perspectives from which CIS are investigated: Computer-supported collaborative work (CSCW) classifies CIS based on the CSCW matrix [Baecker et al. 1995] as time- and space-independent groupware, which focuses on communication and coordination. Researchers from CSCW and multi-agent systems (MAS) also explore CIS as socio-technical systems, which are hybrid systems where "active components are mostly represented by humans, whereas interaction is almost-totally regulated by the software infrastructure" [Omicini and Contucci 2013]. In addition, coordination and MAS researchers see the aspect of indirect, environment-mediated interaction between the agents as a central characteristic. System dynamics in CIS are researched in physics, network science and computational social sciences, where a focus is on complex networks and the investigation of the dynamics of these networks. In recent years, CIS are also under investigation from the viewpoint of collective intelligence (CI), which is a phenomenon that is researched in several fields like sociology, biology, political science and economics for decades. Pierre Lévy's [1997] often cited definition describes CI as *"a form of universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills. [...] No one knows everything, everyone knows something."*.

Though, despite the wide adoption of CIS in today's life, there still remains a lack of consolidated systematic knowledge of their architectural principles and practices. From our experience with industry partners, we learned that due to a lack of CIS-specific knowledge, software architects often resort to imprudently reproducing designs of existing CIS without understanding their underlying principles, mechanisms and rationales. Subsequently, software architects are severely impacted in their ability to anticipate unintentional effects caused by their architectural design decisions on the system's main capabilities and to reason about the completeness of its basal functionalities.

In this work, we contribute the STIGMERGIC INFORMATION SYSTEM (SIS) architecture pattern, which describes a system architecture that includes the common elements and processes of CIS on a systemic level. With this pattern software architects are able to efficiently describe core elements of a CIS architecture without being limited in its technical implementation.

Central to the STIGMERGIC INFORMATION SYSTEM architecture pattern is the nature-inspired coordination mechanism of stigmergy (from Greek *stigma*: sign, and *ergon*: work) which was originally used to describe self-organizational, environment-mediated task coordination of social insects [Bonabeau et al. 1999]. Stigmergy is *"indirect communication among individuals where the trace left in the environment by an action stimulates subsequent actions, by the same or other individuals. By reinforcing each other, the actions aggregate coherent collective knowledge to the benefit of the users"* [Musil et al. 2015]. In the field of MAS, STIGMERGY is also regarded as a design pattern for creating complex, self-organizational behavior. A thorough account of STIGMERGY and related nature-inspired patterns has been provided by Omicini [2013] as well as by Juziuk et al. [2014] in their systematic literature review on MAS design patterns.

The SIS pattern is grounded on surveying existing CIS, the development of a pilot CIS and two cases where CIS architectures have been created with industry partners. For the survey we used Alexa[1] to identify popular CIS, whereby we collected 180 CIS from which we analyzed 30 of them in detail. Based on the analysis results, we derived six key features, that are described in section 5. To systematically architect CIS based on the SIS-pattern we also developed an architecture framework which follows ISO/IEC/IEEE 42010 standard for architecture descriptions [2011]. For a detailed description of the architecture framework and the two industry cases we refer the

[1] Alexa Internet Inc. provides commercial web traffic analytics and global rankings: `http://www.alexa.com/` (last accessed: 6/15/2015)

interested reader to [Musil et al. 2015].

It is important to consider that the presented SIS pattern is not meant to address the technical specifics of the implementation of CIS, since depending on the architectural concerns there may be multiple ways to implement the SIS pattern within a system-of-interest. This work represents a reworked and extended version of the SIS pattern that was presented at the MiniPLoP writers' workshop [Harrison et al. 2015], which was co-located with ECSA 2014, and includes feedback from the workshop's participants.

The remainder of the work is structured as follows: Section 2 introduces basic characteristics of CIS. Section 3 describes the SIS pattern in pattern writing form. Section 4 discusses related patterns and section 5 describes an approach to identify the SIS pattern in the field. Finally, section 6 draws conclusions and details future work.

2. COLLECTIVE INTELLIGENCE SYSTEM CHARACTERISTICS

Collective intelligence systems (CIS) enable IT-mediated collective intelligence [Malone and Bernstein 2015] and are a particular family of socio-technical systems. A main CIS characteristic is their capability for *"bottom-up information sharing and knowledge aggregation by combining the strength of computing systems (data processing, workflow coordination) with the cognitive capabilities of human groups (abstract thinking, pattern recognition)"* [Musil et al. 2015]. An important aspect is that high-level, system-wide behavior of a CIS is *"influenced by low-level rules, encapsulated by the coordination infrastructure that comprises artifacts that store the shared content and define the rules of interaction and coordination, and local activities of the individual users"* [Musil et al. 2015]. A driving force of emergence in CIS is the stigmergic process [Heylighen 2015], which promotes *awareness* among agents about the activities of other agents, which in turn reinforces their own activities [Ricci et al. 2007]. In computer science, stigmergy is well-known as an effective coordination model that provides computational systems with bottom-up, environment-mediated coordination capabilities [Babaoglu et al. 2006; Omicini 2013; Ricci et al. 2007; Parunak 2006]. For socio-technical systems, stigmergy is of particular relevance since the interaction between the human agents is predominantly mediated/regulated by the software infrastructure [Omicini and Contucci 2013; Heylighen 2015].

The stigmergic process creates a perpetual, positive feedback loop between a human actor basis and a reactive coordination environment. This feedback loop consists of two alternating process phases (see Figure 1): First in the aggregation phase, actors are able to create and share new artifacts, but also modify, comment and review artifacts created by other actors and thus contribute new content. Second in the dissemination phase, the CIS uses active and passive dissemination mechanisms (e.g. personalized recommendations, notification messages) based on the content contributed by others, to make actors aware about artifact changes and overall environment activity, and to trigger a reaction by the actor base so that phase one starts over again. The interdependence between aggregation (collection of content/knowledge) and dissemination (making others aware about content/knowledge/activity) results in a ongoing, self-organizational coordination cycle.

3. STIGMERGIC INFORMATION SYSTEM PATTERN

This section describes the STIGMERGIC INFORMATION SYSTEM (SIS) architecture pattern according to pattern writing form [Meszaros and Doble 1998]. We illustrate the pattern with the MediaWiki[2] system as a running example. MediaWiki is an open source wiki system, which is widely-used and initially also powered the Wikipedia platform. A wiki is a social web application that allows its users the cooperative creation, modification and linking

[2] `http://www.mediawiki.org/` (last accessed: 6/15/2015)

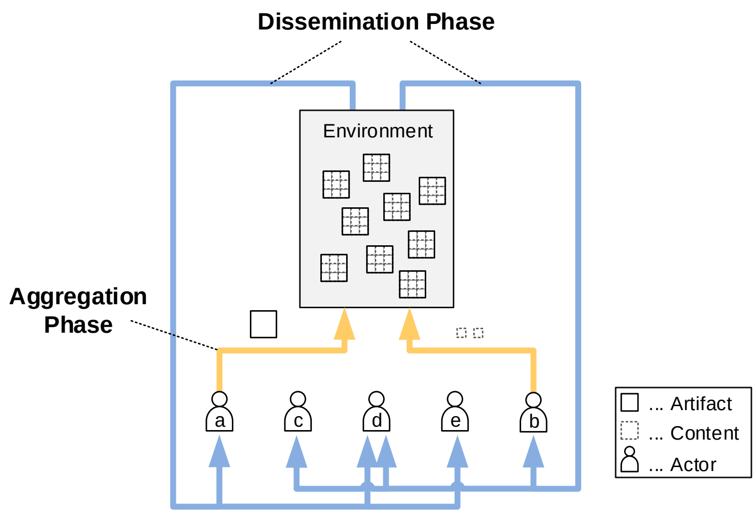

Fig. 1. Stigmergic process with aggregation and dissemination phases (adapted from [Musil et al. 2015]).

of article pages.

3.1 Context

People organize themselves in collectives like groups, organizations, communities and societies for their common or mutual benefit. If people share certain knowledge and information, the collective gets more effective and efficient, which vice versa benefits each individual.

3.2 Problem

The problem is a lack of structured coordination to share and retrieve the knowledge and information between human agents by using a software system. Knowledge and information are dispersed among individuals and thus are difficult to access on a collective level. Each individual is situated in a specific, local context and thus has low global awareness about available, remote information from other individuals, also with respect to its context, status, relevance and sources.

3.3 Forces

The problem is affected by the following forces:

—F1: Varying quality of shared information due to inconsistent structuring and poorly defined information aggregation and integration processes.

—F2: Users are burdened by information overload, since more information is available than a user can process and actually needs.

—F3: High effort for the user to retrieve relevant information due to low discoverability of potentially useful information.

—F4: Timeliness of information transfer depends solely on the individual, who has access to the requested information.

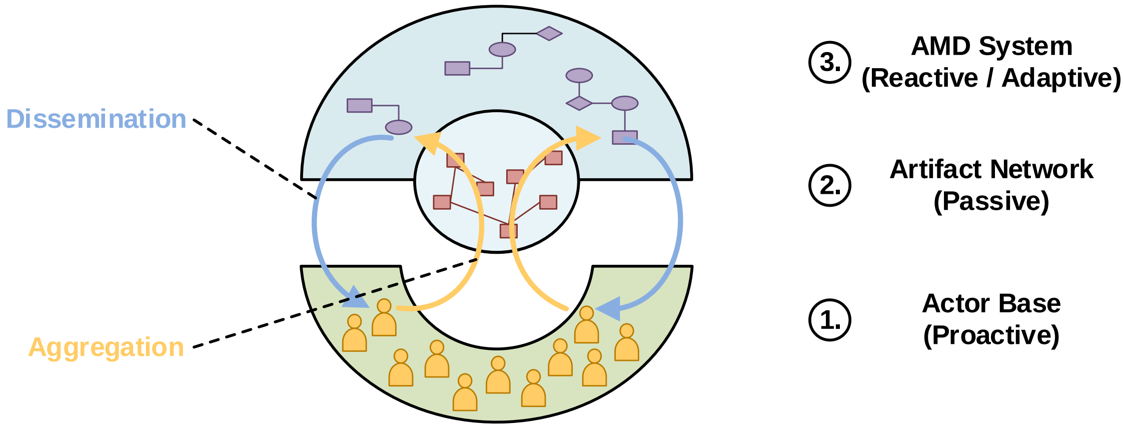

Fig. 2. Simplified SIS pattern with stigmergy cycle (adapted from [Musil et al. 2012]).

—F5: Users face a lack of awareness about activities performed by other users on shared content.

3.4 Solution

Therefore we propose the STIGMERGIC INFORMATION SYSTEM (SIS) architecture pattern: A SIS describes a system architecture that facilitates the bottom-up building of an artifact network by allowing its actors to create/modify user-generated content stored in artifacts and thereby effectively accumulate and share information among each other [Musil et al. 2012]. A SIS consists of three main components which are illustrated in Figure 2: (1) an actor base, represented by a group of human agents, as proactive component, (2) a single, homogeneous, CI artifact network as a passive component, and (3) a computational analysis, management and dissemination (AMD) system as a reactive/adaptive component. The continuous flow of actor contributions within the system environment enables the emergence of collective intelligence that allows the individual to benefit for own purposes. The SIS pattern facilitates the design of a socio-technical multi-agent system architecture which mediates human interaction and provides support for distributed cognitive processes.

This SIS-based architecture design enables a CIS to provide emergent, bottom-up, self-organizational knowledge transfer and coordination capabilities to human groups and organizations. The CIS architecture creates a perpetual feedback loop between actor base (1) and coordination infrastructure (2, 3), by instrumenting the actors' contributions to stimulate a subsequent reaction by other actors, causing a stigmergy-like process. Figure 3 illustrates a detailed metamodel, that underlies the SIS pattern, describing key elements and their relations.

1. Actor Base. The actor base layer consists of *human actors*, who independently perform *activities* on the *CI artifacts*. In a MediaWiki, the actor base comprises all editors, who are typically all users, who have created an account at the wiki.

2. Artifact Network. The artifact network layer consists of *CI artifacts* which store topic-specific content generated by the actors. CI artifacts are manipulated by *actor activities* which resemble different types of create, read, update and delete operations. An important activity is the linking of artifacts using artifact links. *Artifact links* are links that actors can set between artifacts, leading to the emergent creation of an *artifact network* which is shared among the total actor base. Each performed activity is tracked in an *actor record*, whereby each actor has her own actor record. The actor record has two main purposes: First, it logs the complete actor activities of an individual actor which allows the system to build knowledge about its actors and to provide advanced services like recommendations and shared interests. Second, the actor record acts as a proxy for the *ownership relationship* between the actor and the CI artifacts. The ownership relationship defines who is the owner of an artifact and thus has extensive control to decide (1) to which extent other actors are able to contribute to the CI artifact, and (2) if contributions comply to predefined qualitative requirements. In a MediaWiki, the CI artifact is the article page and artifact links are represented by so-called internal links. The actor record is stored in the logging table and can be accessed via the Special:Log page by querying for a particular performer. Typically, all editors have equal ownership rights to all article pages in a MediaWiki, which allows an editor the extensive manipulation of articles created by other editors.

3. AMD System. The analysis, management and dissemination (AMD) system is a reactive/adaptive computational system which enforces different *dissemination rules*. Based on these rules, *filtered content* is generated from CI artifact content and actor records. The filtered content is used by generated *triggers*, also defined by the dissemination rules, that are sent to individual actors in order to propagate changes of CI artifacts as well as to stimulate subsequent actor activities. Such triggers make 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.

In a Media Wiki, several types of events are defined that trigger a short notification message for the individual editor to inform her about recent activities, e.g. a new message on the user's talk page, linking of the user page to any other page (mentions), linking to an article page that the user has created, and when a page the user has created is reviewed.

3.5 Examples

Facebook[3]*.* The CI artifact in Facebook is the user profile. The actor base comprises people that can be described with the organizational structures and processes of a society. Each actor is the owner of only one profile. Users primarily contribute to their own profile pages, but they can also contribute directly to information shared by other users or their profile pages. Artifact links are represented by the friend relationship between two profiles and are defined by the actors. Facebook generates different kinds of triggers, for example the personalized News Feed presenting recent user activities, system pop-up notifications about friends' real time interactions with the user, and activity alert notifications via email.

Stack Overflow[4]*.* The CI artifact in Stack Overflow is the user question. The community of people interested in a wide range of topics in computer programming forms the actor base. Each actor can be the owner of several questions. Users primarily contribute to Stack Overflow by creating new questions or sharing their knowledge in form of answers to existing questions. Artifact links are represented by tags that actors can assign to questions. Actors automatically subscribe to their own created questions and to preferred tags, and thus receive triggers in form of email notification messages about recent activities. Alternatively, real-time updates are provided on the platform.

YouTube[5]*.* The actor base of the media-sharing platform YouTube covers both private individuals and professional organizations of a society. The CI artifact is a user-generated video which can primarily be uploaded, shared, discovered and watched by the actor base. Each actor can be the owner of several videos. By editing a video the actor is allowed to define tags and categories of the artifact which represents the artifact links. Actors can subscribe to YouTube channels they like in order to receive notifications about ongoing activities to keep

[3] `http://www.facebook.com/` (last accessed: 6/15/2015)
[4] `http://www.stackoverflow.com/` (last accessed: 6/15/2015)
[5] `http://www.youtube.com/` (last accessed: 6/15/2015)

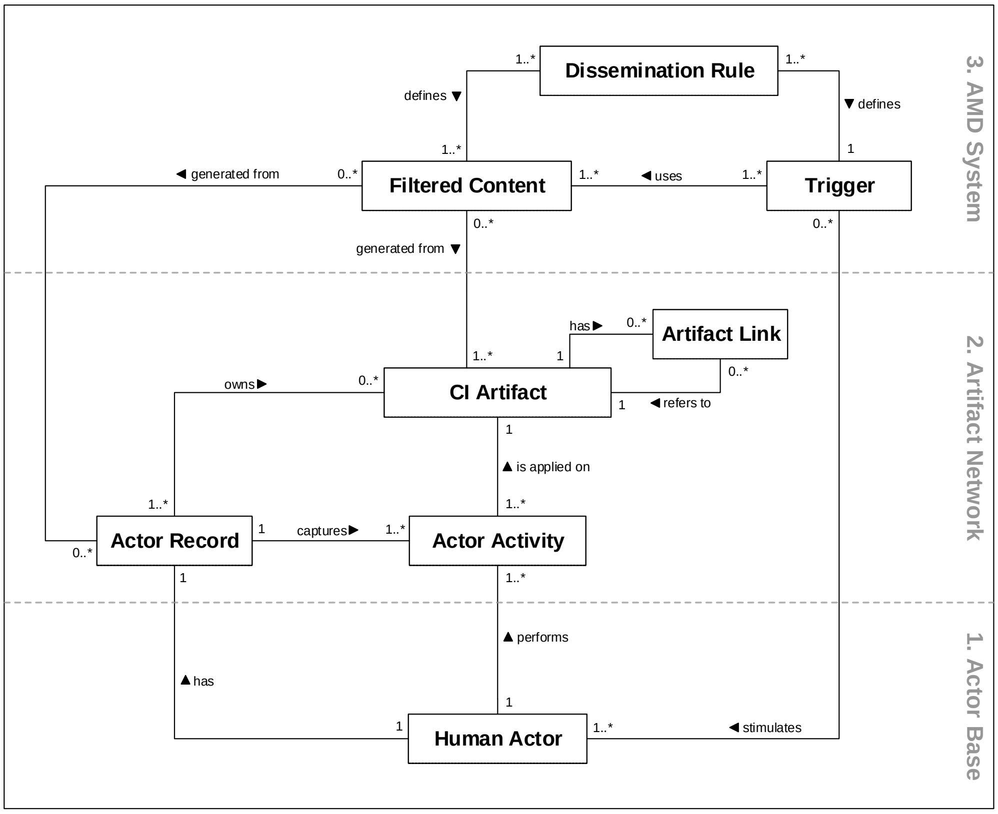

Fig. 3. Detailed SIS pattern as simplified UML class diagram (refined from the metamodel in [Musil et al. 2014]).

the subscribers up to date. This personalized emails are sent regularly to the actors and are an example of a generated trigger. Another trigger example are the recommendations for channels and videos the actor may be interested in presented in the feed on the platform homepage.

Thingiverse[6]*.* The actors participating in MakerBot's Thingiverse are members of a 3D design community who are interested in creating, experimenting, and printing useful, innovative or funny 3D models of things. These printable 3D designs of things form the network of CI artifacts. Each actor can be the owner of several shared 3D designs. Primary actor activities in Thingiverse comprise the sharing of new created or remixed 3D model designs with the community and discovering and printing of existing designs. Actors can assign tags/categories to 3D model designs, but they can also add artifacts to collections, which generates different kinds of artifact links.

3.6 Example Scenario: Document-based Knowledge Sharing

An example scenario is the sharing of knowledge by means of individual digital documents by a group of users.

[6] `http://www.thingiverse.com/` (last accessed: 6/15/2015)

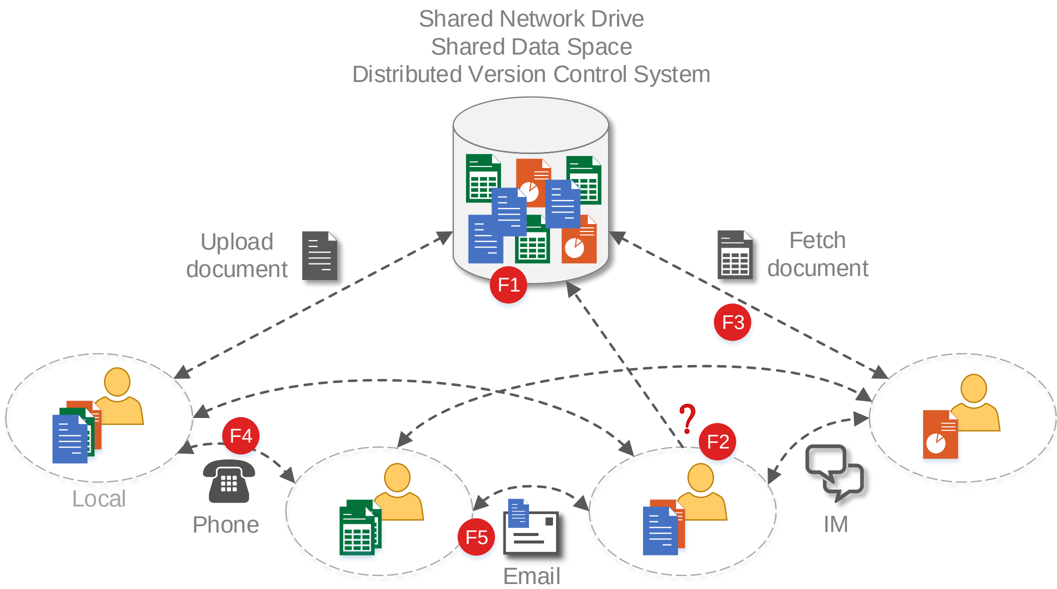

Fig. 4. Overview of the user's challenges before the support provided by a wiki.

Initial case (Figure 4): Content (text, figures, tables) is stored in individual digital documents, e.g. text files. Each user has her own local space with her personal set of documents. A subset of these documents are shared with other users via a shared space like a network drive or a version control system (VCS). A user can also modify a document retrieved from the shared space. This setup causes several limitations: In order to make meaningful modifications or use one of the shared documents, a user needs additional information which is gathered from other users via various communication channels, like email, instant messaging (IM) or phone. The same channels are also used when a user intends to inform herself about document changes that happened in her absence (less a problem in a VCS) or where to find a certain shared document. The last point may also lead to unintended duplication of a document in the shared space or the ad hoc sharing via email. In this scenario the shared space gets less manageable the more documents are in it and the more users are involved, whereby the effort and complexity of sharing knowledge for each user also increases.

SIS-based case (Figure 5): In the SIS-based case, a Wiki-type CIS has taken the place of the shared space. Other than in the initial case, the users can now share content either by creating a new article or by modifying an existing one. Activities of the user are tracked in an actor record (AR)-like log, as is any modification of any article. Each submitted article modification creates a new revision of the article. This improves the traceability of modifications by other users and enables them to undo changes. Articles can be linked together by users via Wikilinks and categories, creating a network of related articles which improves content discoverability. Also each article artifact has a talk page where users can discuss the content of the individual article. To improve awareness of changes during a user's absence, the system sends personalized notification emails.

3.7 Consequences

The SIS pattern has the following benefits:

—The created system architecture realizes a stigmergic coordination model which generates an ongoing feedback loop between information aggregation and dissemination phases.

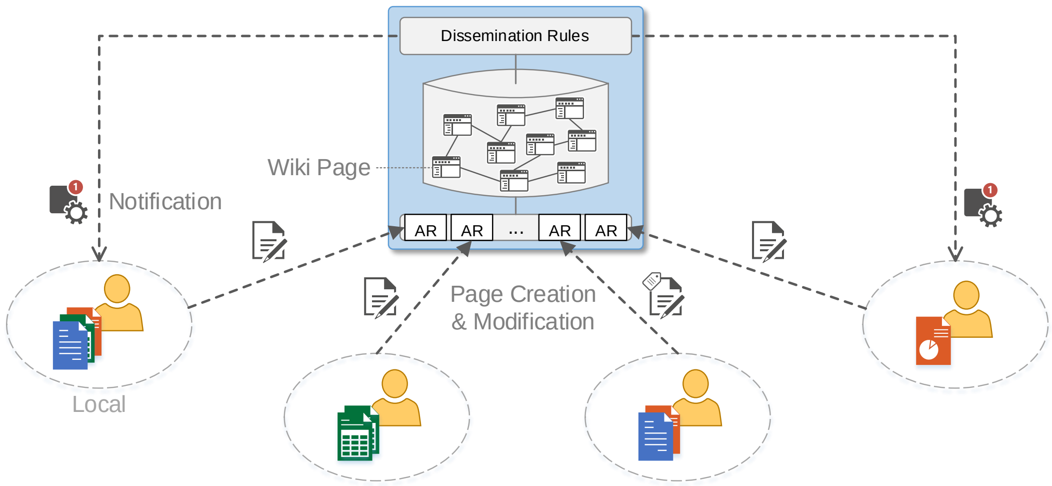

Fig. 5. Overview of the user's workflow supported by a wiki.

—The artifact network and the AMD system mediate sharing and retrieving of content and the interactions between individual actors. (counters force 1)

—Division of labour, since an artifact evolves by incremental contributions and modifications of different actors, and each actor contributes information that is available in her own sphere of expertise. (counters force 2)

—The artifacts predetermine how and what information can be shared between the actors. (counters force 1)

—The AMD system autonomously filters relevant/trending content and ongoing activities, and diffuses them effectively and continuously among the actor base, thus improving the awareness of the individual user. (counters forces 3, 4, 5)

—The actor record enables traceability of actor activities performed in the artifact network. (counters force 5)

The SIS pattern has the following disadvantages:

—A CIS may have low or detrimental effects, if there is a mismatch between the system design (type and structure of shared content, activities) and the organizational workflows it ought to support, causing slow-downs of processes, obstructions of workflow steps and duplication of information.

—It increases the need for organizational rules to guide collaboration on the platform.

—Improved transparency through bypassing of established organizational hierarchies and informal information brokers, which may cause opposition of the system's adoption among the user base.

—The created system is driven by actor activities, thus it is critical to maintain a certain activity level of the actor base so that enough new content is available for the system to build new triggers.

—Initial building of an artifact network (if there is no preseed of content) and actor base takes time.

—The design and tuning of incentive mechanisms to maintain/increase actor engagement with the system requires additional expertise from domains like psychology or interaction design.

3.8 Known Uses

The SIS pattern can be found in a wide array of social web applications and collaborative platforms: Facebook, Google+, Twitter, Jira, Confluence, YouTube, Stack Overflow, Wikipedia, MediaWiki, Wikia, Foursquare, Instagram, Bugzilla, LinkedIn, Xing, EBay, InnoCentive, Mendeley, Researchgate, TripAdvisor, Thingiverse, Yelp, Flickr.

4. RELATED PATTERNS

The SIS architecture pattern can be organized within the multi-agent system pattern space. In this section we discuss the SIS pattern with respect to the BLACKBOARD [Erman et al. 1980] and REFLECTIVE BLACKBOARD [Silva et al. 2003] pattern regarding their commonalities and differences.

Like in a REFLECTIVE BLACKBOARD system, there is a separation of concerns regarding application data and control data. The base level of the REFLECTIVE BLACKBOARD corresponds to the artifact network and actor levels, whereby the meta-level maps to the AMD system level. Knowledge sources in a BLACKBOARD system are independent software agents / modules, whereby actors in a SIS-based system are typically human agents. The SIS and BLACKBOARD patterns rely on a shared space-like structure to store the collective contributions of the individual agents, although the structure of this shared space is different. In a SIS-based system the agents collectively build a network of artifacts, whereby each artifact can be modified by its creator and with varying extent by other actors. Like a BLACKBOARD, a single artifact accumulates additive contributions from multiple agents in order to create the most comprehensive solution set. Other than the control shell in the BLACKBOARD system, the AMD system in a SIS-based system does not actively control the flow of problem-solving activities. Actors in the SIS are self-determined and decide at their own discretion if, when and to what extent they contribute to the system. Therefore the AMD system can only try to indirectly influence its actor base by notifying and recommending them filtered, personalized information about artifacts, contributions and activities of other actors. The reason of actor autonomy may also lead to the likely case that if there are no agents willing or available to contribute, a SIS-based system does not yield any actor-generated solution data at all, which is a less probable scenario in a BLACKBOARD style-like system.

5. IDENTIFYING THE SIS PATTERN IN THE FIELD

To check if a software platform realizes the SIS pattern, the following features need to be positively identified [Musil et al. 2014]:

(1) Can any actor add a new CI artifact?[7]
(2) Can any actor contribute to parts of the CI artifact of an other actor, thus change its state?
(3) Are actors able to create system-internal links to connect coordination artifacts?[8]
(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?[9]

[7]Access restrictions (password wall, pay wall, etc.) are not an exclusion criteria as long as they affect all actors in the same way.
[8]Examples are the friend-relationship in Facebook or Wikilinks in Wikipedia (`http://en.wikipedia.org/wiki/Help:Link#Wikilinks`) (last visited at 6/15/2015).
[9]Client-side tracking of usage behavior via cookies is not sufficient.

6. CONCLUSION

In this work we illustrated the STIGMERGIC INFORMATION SYSTEM architecture pattern, that should support software architects to consider relevant elements and processes in a system architecture design of CIS, and explained it using real-world examples and scenarios. We showed that this pattern is capable to describe CIS on a systemic level.

Future work encompasses multiple lanes of research: One lane is to use the provided SIS pattern as a base from which more specific system patterns are derived that cover sub-families of CIS based on variations of architecture significant elements in the base pattern. This family of patterns can inform a pattern catalogue and subsequently a pattern language to guide architectural decision making in the CIS domain. A second research direction are empirical studies for CIS architecting, both by strengthen the validity of the identified models and patterns, and their effectiveness to design and document CIS architectures in practice.

ACKNOWLEDGMENT

We thank our shepherd Christian Kohls and the participants of the writers' workshop group at EuroPLoP 2015 as well as the participants of the MiniPLoP event, which was co-located with ECSA 2014, for their constructive and helpful feedback and comments to improve the paper.

REFERENCES

BABAOGLU, O., CANRIGHT, G., DEUTSCH, A., CARO, G. A. D., DUCATELLE, F., GAMBARDELLA, L. M., GANGULY, N., JELASITY, M., MONTEMANNI, R., MONTRESOR, A., AND URNES, T. 2006. Design patterns from biology for distributed computing. *ACM Transactions on Autonomous and Adaptive Systems 1,* 1, 26–66.

BAECKER, R. M., GRUDIN, J., BUXTON, W., AND GREENBERG, S. 1995. *Readings in Human-Computer Interaction: Toward the Year 2000* 2nd Ed. Morgan-Kaufmann.

BONABEAU, E., DORIGO, M., AND THERAULAZ, G. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.

ERMAN, L. D., HAYES-ROTH, F., LESSER, V. R., AND REDDY, D. R. 1980. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Computing Surveys 12,* 2, 213–253.

HARRISON, N. B., VAN HEESCH, U., SOBERNIG, S., SOMMERLAD, P., FILIPCZYK, M., FÜLLEBORN, A., MUSIL, A., AND MUSIL, J. 2015. Software Architecture Patterns: Reflection and Advances. *ACM SIGSOFT Software Engineering Notes 40,* 1, 30–34.

HEYLIGHEN, F. 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.

ISO/IEC/IEEE 42010. 2011. *Systems and Software Engineering - Architecture Description*.

JUZIUK, J., WEYNS, D., AND HOLVOET, T. 2014. Design Patterns for Multi-agent Systems: A Systematic Literature Review. In *Agent-Oriented Software Engineering*, O. Shehory and A. Sturm, Eds. Springer Berlin Heidelberg, 79–99.

LÉVY, P. 1997. *Collective Intelligence: Mankind's Emerging World in Cyberspace*. Perseus Books.

MALONE, T. W. AND BERNSTEIN, M. S., Eds. 2015. *Handbook of Collective Intelligence.* MIT Press.

MESZAROS, G. AND DOBLE, J. 1998. A pattern language for pattern writing. *Pattern languages of program design 3*, 529–574.

MUSIL, J., MUSIL, A., AND BIFFL, S. 2014. Towards a Coordination-Centric Architecture Metamodel for Social Web Applications. In *Proc. of the 8th European Conference on Software Architecture (ECSA '14)*, P. Avgeriou and U. Zdun, Eds. LNCS Series, vol. 8627. Springer International Publishing, 106–113.

MUSIL, J., MUSIL, A., WEYNS, D., AND BIFFL, S. 2015. An Architecture Framework for Collective Intelligence Systems. In *Proc. of the 12th Working IEEE/IFIP Conference on Software Architecture (WICSA '15)*. IEEE, 21–30.

MUSIL, J., MUSIL, A., WINKLER, D., AND BIFFL, S. 2012. A First Account on Stigmergic Information Systems and Their Impact on Platform Development. In *Proc. of the WICSA/ECSA 2012 Companion Volume (WICSA/ECSA '12)*. ACM, 69–73.

OMICINI, A. 2013. Nature-Inspired Coordination Models: Current Status and Future Trends. *ISRN Software Engineering 2013*.

OMICINI, A. AND CONTUCCI, P. 2013. Complexity and Interaction: Blurring Borders between Physical, Computational, and Social Systems. Preliminary Notes. In *Proc. of the 5th Int'l Conf. on Computational Collective Intelligence Technologies and Applications (ICCCI '13)*. LNCS Series, vol. 8083. Springer Berlin Heidelberg, 1–10.

O'REILLY, T. 2007. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Communications & strategies 65*, 17–37.

PARUNAK, H. V. D. 2006. A Survey of Environments and Mechanisms for Human-Human Stigmergy. In *Environments for Multi-Agent Systems II*, D. Weyns, H. V. D. Parunak, and F. Michel, Eds. LNCS Series, vol. 3830. Springer Berlin Heidelberg, 163–186.

RICCI, A., OMICINI, A., VIROLI, M., GARDELLI, L., AND OLIVA, E. 2007. Cognitive Stigmergy: Towards a Framework Based on Agents and Artifacts. In *Proc. of the 3rd Int'l Conf. on Environments for Multi-Agent Systems III (E4MAS '06)*. LNCS Series, vol. 4389. Springer Berlin Heidelberg, 124–140.

SILVA, O., GARCIA, A., AND LUCENA, C. 2003. The Reflective Blackboard Pattern: Architecting Large Multi-agent Systems. In *Software Engineering for Large-scale Multi-agent Systems*, A. Garcia, C. Lucena, F. Zambonelli, A. Omicini, and J. Castro, Eds. LNCS Series, vol. 2603. Springer Berlin Heidelberg, 73–93.

SUROWIECKI, J. 2005. *The Wisdom of Crowds*. Abacus.

EuroPLoP '15, July 08 - 12, 2015, Kaufbeuren, Germany

ACM 978-1-4503-3847-9/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2855321.2855342