Technical Report

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Stigmergic Information Systems: A Conceptual Framework for Architecting Social Web Applications

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

Christian Doppler Laboratory CDL-Flex, Institute of Software Technology and Interactive Systems, Vienna University of Technology
Favoritenstrasse 9-11/188, AT 1040 Vienna, Austria
[1]{juergen.musil, stefan.biffl}@tuwien.ac.at
[2]angelika.musil@qse.ifs.tuwien.ac.at

Technical Report No. IFS-CDL 14-25
Issued: April 2014

Stigmergic Information Systems: A Conceptual Framework for Architecting Social Web Applications

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

Christian Doppler Laboratory for
Software Engineering Integration for Flexible Automation Systems,
Institute of Software Technology and Interactive Systems,
Vienna University of Technology,
Favoritenstrasse 9/188, 1040 Vienna, Austria
[1]{juergen.musil,stefan.biffl}@tuwien.ac.at
[2]angelika.musil@qse.ifs.tuwien.ac.at

Abstract. Wikis, social networking services 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 architecting them is still at the beginning. Therefore we propose the Stigmergic Information System (SIS) approach describing a novel architectural meta-model to facilitate implementation-independent system design for social web applications and the analysis of the system's behavior to identify next design desicions for improvement. The SIS approach organizes a system into four layers: agent, artifact data, analysis & control and workflow. We evaluated the SIS approach's feasibility with (1) a prototypical implementation of a glossary system which was designed based on the meta-model and (2) the analysis of system critical elements and dynamics to identify limitations in the system behavior.

Keywords: Collective Intelligence, Coordination, Knowledge Management, Social Web, Socio-Technical System, Software Architecture, Stigmergic Information System, Stigmergy

1 Introduction

Over the last decade, new forms of online collaboration platforms like wikis, social networks, crowdsourcing markets and social media platforms have enabled individuals to communicate and work together on problems in an effectiveness, which has not been envisioned at the end of the last century. These types of social web applications can also be referred as *socio-technical systems*, which have been defined by Omicini [21] from a coordination perspective as a particular class of social systems *"where the active components are mostly represented by humans, whereas interaction is almost-totally regulated by the software infrastructure"*. Key characteristics of socio-technical systems are that they enable bottom-up collaboration and combine the strengths of computing systems (effective data processing) with the cognitive capabilities of groups of people (abstract thinking, pattern recognition). Today, research investigating the principles and synergetic effects of networked human groups and computing systems runs under various names [24] like crowdsourcing, social computing and human computation. 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. In this paper we present the technology-independent Stigmergic Information System (SIS) approach, which describes a novel architectural meta-model to provide software architects guidance for the system design of social web applications with capabilities for bottom-up knowledge transfer and coordination of collaborative activities among human groups. Also, the approach enables the analysis of system behavior parameters and significant metrics derived from the meta-model elements to assess the system's behavior and identify next design desicions for improvement. We evaluate the SIS approach's feasibility with (1) a prototypical implementation of a glossary system for scientific theory building which was systematically designed using the SIS meta-model, and (2) the analysis of the glossary's system behavior with respect to the meta-model's critical elements. Results show a successful mapping of the meta-model elements to the glossary system elements and a comprehensible system performance evaluation whose identified limitations can be used as input for futher improving the system's design.

The remainder of this paper is structured as follows: Section 2 summarizes related work and the research challenges are introduced in section 3. We present the Stigmergic Information System architecture approach describing the meta-model and characteristic process in section 4. In section 5 we present the evaluation of the approach and discuss the results in section 6. Finally, section 7 concludes and outlines future work.

2 Related Work

This section presents an overview of related work on the topics of coordination models, environment-mediated stigmergic coordination and architecting coordination in social web applications.

2.1 Coordination Models

Coordination models are described by Gelernter et al. [11] as *"the glue that binds separate activites in an ensemble"* and by Omicini [21] as basal to define *"the abstractions and the computational models for ruling the interaction space in computational systems"*. Ciancarini [5] identifies coordination entities, coordination media and coordination laws as the constituents of a coordination model for computational systems. *Coordination entities* are the entities that are being coordinated, like processes, threads, agents or humans. The *coordination media* enables communication among the entities, and serves as means for manipulations among the whole entity base. Examples of coordination media can be simple constructs like semaphores or monitors or complex constructs like tuple spaces [10] and blackboards [7]. Finally, *coordination laws* describe rules, constraints and mechanisms how entities are coordinated by the means of the coordination media. The duality between coordination medium and laws has also been described by Schmidt and Simone [27] in the context of Computer-Supported Cooperative Work (CSCW) using similar concepts of coordinative artifact and coordinative protocol. Ciancarini [5] notes that coordination models are either embedded in software architectures or coordination languages. Mordinyi and Kühn [16] distinguish coordination models in formal and conceptual frameworks. Formal frameworks express coordination systems by means of formal notations and rules, whereby conceptual frameworks assist system architects and developers with abstraction mechanisms in managing interactions. Papadopoulos et al. [23] surveyed coordination models and identified control-driven or data-driven as the two major approaches. In *control-driven* coordination (e.g. in languages like WS-BEPL[1]), the manipulated data is of no concern to other processes due to a separation of coordination and computation. Whereby in *data-driven* coordination *"the state of the computation at any moment in time is defined in terms of both the values of the data being received or sent and the actual configuration of the coordinated components"* [23]. A popular data-driven approach is using the concept of a *shared dataspace* [26] as coordination medium, which is a common, content-addressable data structure [26, 23] (like LINDA tuple spaces [10] or JavaSpaces[2]). The coordination entities communicate indirectly among each other by manipulating the shared dataspace. Manipulation can be either posting information into or removing/copying information from the space [26]. Advantages of environment-based coordination approaches, like shared dataspaces, are that processes can be decoupled in space and time, as well as that producer and consumer can be anonymous [23]. A special form of environment-based coordination, stigmergy, should be discussed more in details.

2.2 Mediated Interaction Through Stigmergic Coordination

Stigmergy (from Greek *stigma*: sign, and *ergon*: work) is a coordination mechanism introduced by Grassé [13] to describe the environment-mediated task coordination of social insects. Therefore, stigmergy enables not only environment-mediated coordination and communication, it possesses also a positive feedback mechanism [3, p.14][4], so that activity causes more activity. In detail, stigmergy promotes *awareness* among agents about the activities of other agents, which in turn reinforces their own activities [25]. The process behavior of stigmergy is *emergent*, meaning that certain system properties exist on a high-level, but

[1] `http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html` (All URLs referenced in this paper have been last visited at 02/17/2014.)

[2] `http://www.oracle.com/technetwork/articles/javase/javaspaces-140665.html`

not on a low-level and vice versa [2]. For stigmergy, this means that high-level, system-wide behavior is influenced by low-level rules, encapsulated by the artifacts and the environment, and *local* activities. There is no explicit coordination control [8], and the agents are independent and choose autonomously which activities they perform [9, 22, 8]. In computer science, stigmergy is well-known as an effective, nature-inspired coordination model, which provides computational systems with bottom-up, environment-mediated coordination capabilities [9, 28, 1, 25, 20]. When using a computational system to coordinate a multi-agent systems (MAS) through stigmergy, the concepts of environment and artifact are essential [28, 25, 22]. Weyns et al. [31] notes 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. Extensive discussions of coordination artifacts from a MAS perspective can be found in [22, 8] and from a CSCW perspective in [27]. Omicini et al. [22] provide a particular perspective on agents and artifacts in their agents & artifacts (A&A) meta-model for MAS. In this approach (1) agents are pro-active components, which autonomously execute activities inside an environment, whereby (2) artifacts are *"passive components which are cooperatively or competitively constructed, manipulated and shared by the agents to support their activites"* [22]. Additionally there are workspaces, which represent local environments in which agents can interact with artifacts [22].
For social web applications, stigmergy is of particular relevance, since the interaction between the human agents is predominantly mediated/regulated by the software infrastructure [21]. Susi et al. [28] provides a conclusive description of using stigmergy to support human cognitive processes and the usage of artifacts as mechanism to mediate emergent human collective behaviour. Ricci et al. [25] adapted Susi's work towards a theory of *cognitive stigmergy* for MAS, which proposes the dual usage of artifacts as means (1) to enable emergent coordination processes and (2) to share and represent high-level knowledge for cognitive agents, like humans. In their work they identified the recurring stigmergic mechanisms of diffusion, aggregation, selection and ordering [25]. Parunak [30] surveyed stigmergic computational systems, which are used to coordinatne human interactions. So far, some types of social web applications (social networking services, wikis) have been identified as stigmergic systems [30, 25, 8, 21].

2.3 Architecting Coordination in Social Web Applications

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 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. The challenge of architecting social web applications is a well known: In 2001, Tiwana and Bush [29] 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 [12] described their experiences from designing two "social interaction web sites" for two different 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 [6] present a human Architecture Description Language hADL to describe collaboration structures and patterns in social web applications. Minder and Bernstein [15] focus on human computation and propose with CrowdLang a programming framework for interaction mechanisms and the design of human computation systems. Another workflow-based approach, which focuses on architecting human-based service clouds, is presented by Hong-Linh et al. [14]. Finally, a different perspective is provided by Nebeling et al. [19], who propose a crowdsourced approach for web engineering and design itself by providing a domain-specific crowdsourcing platform and a web-based design environment.
This section has provided some background on why self-organizational coordination models like stigmergy can provide useful coordinative mechanisms for social web applications.

3 Research Challenges

At the moment the lack of easy-to-understand architectural models and design processes presents a gap, which prevents a wide applicability of the theoretical models, like stigmergic coordination, and their benefits on domains like social web applications. In this work, we concentrate on this gap with the following research challenges:

RC1 - Definition of the novel architectural meta-model, called Stigmergic Information System (SIS), to facilitate the practical system design of social web applications with capabilities for bottom-up knowledge transfer and coordination of collaborative activities among human groups.

RC2 - Investigation of the meta-model's feasibility in a real-world scenario and derivation of system behavior parameters and significant metrics from the meta-model elements to evaluate the system's behavior and identify limitations in the system's design.

We evaluate the SIS approach with the design and prototypical implementation of a glossary system which we investigate in two steps: Firstly, we inspect how the individual meta-model elements map to the glossary's system elements. Secondly, we analyze the glossary's system behavior with respect to the meta-model's critical elements. Then the glossary's performance evaluation results are used as a basis to identify current limitations and with respect to the SIS approach to provide guidance for further improving the system's design.

4 The Stigmergic Information System (SIS) Architecture Approach

This section presents the Stigmergic Information System (SIS) architecting approach and describes its meta-model and its characteristic process model. A preliminary outline of the SIS approach has been presented in [18] and [17].
We have defined a Stigmergic Information System [18] as *"a software platform, which facilitates the building of an information network by allowing actors to create/modify network elements and thereby share information among each other. Hence a SIS is a combined communication system and information regulation system. The central principle behind SIS is harnessing collective intelligence by stimulating, aggregating, leveraging, and distributing user contributions"*.
The Stigmergic Information System approach enables the technology-agnostic description and design of a certain type of socio-technical system architecture. The SIS architecture realizes a composite coordination mechanism consisting of a stigmergic, environment-mediated coordination mechanism in form of a single, homogeneous artifact network embedded in a reactive infrastructure, which facilitates diffusion-like information propagation among its agent base. In detail, a SIS consists of (1) human agents as proactive compontents, (2) a single, homogeneous, coordination artifact network as a passive component, and (3) a computational coordinator system as a reactive/proactive component. In this system the coordinative protocol/coordination laws are encapusled in the coordination artifact network and the computational coordinator system. This design enables a social software system to provide advanced emergent, self-organizational knowledge transfer and coordination capabilities to human groups and organizations, as already seen in wikis and social media services.

The formal model of a Stigmergic Information System is described by (1) a meta-model which define basic elements and relations between them on a micro level and (2) a characteristic process describing the system behavior on a macro level.

4.1 SIS Meta-Model

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

I. Agent Layer: The agent layer encompasses types of *human agents*, which 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.

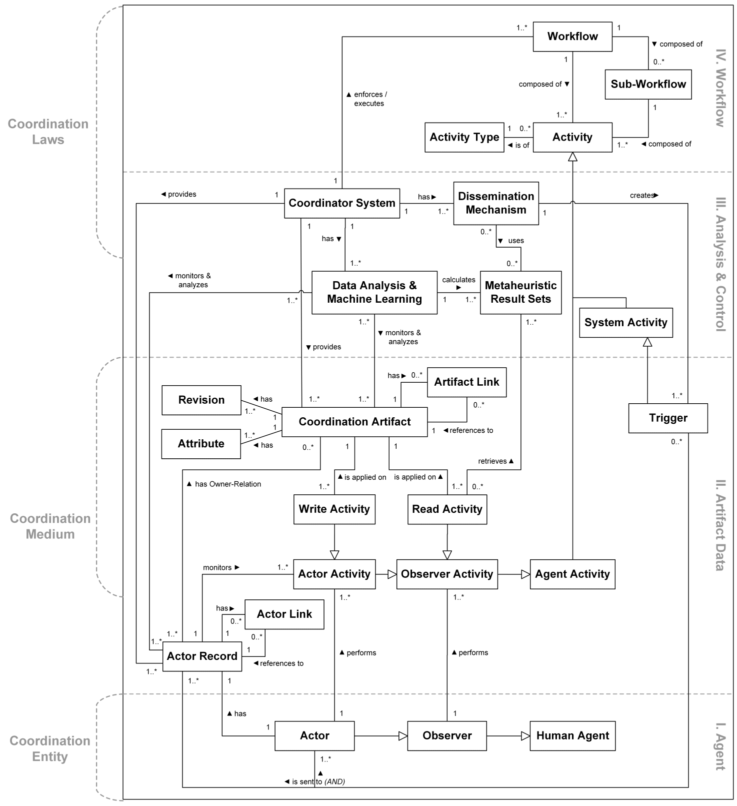

Fig. 1. Architecture meta-model for Stigmergic Information Systems organized in four layers: agent, artifact data, analysis & control, and workflows.

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 can be regarded from their purpose as equal to the artifacts in the A&A meta-model [22]. The CA stores 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). Each performed modification of a CA triggers the creation of a new *revision* entry in the artifact's own history. Versioning is an important artifact functionality, since it enables revisiting actors to learn about activities, which have happened on the artifact in their absence. 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. The circumstance that CAs have inherent linking capability, leads to the creation of a *coordination artifact network*, which is a graph. Each actor has her own *actor record* (AR) that logs an actor's activities within the SIS. Activities, which are logged by the AR are for example all artifact activities, logins, page views and clicks on trace links in notification messages. The AR is of utmost importance for the coordinator system's machine learning subsystem (level III), which uses its data to create and personalize *triggers* that intend to stimulate additional actor contributions.

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 compontent is represented exclusively by autonomous agents interacting through a passive environment, a SIS has with the *coordinator system* an additional reactive/active component. The coordinator system is a computational system, which goal is to influence and mobilize the actor base in order to keep SIS process cycle (fig. 2) running. The coordinator system has a subsystem for *data analysis*, which monitors and processes the content in the CAs and ARs, but also analyzes the of global system behavior (e.g. performance and trends). 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 activity and according to defined workflows from level IV. Typically layer III systems react to changes in the CAs. *Dissemination mechanisms* make the agents, particularly 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 another actors and so on. It can be discriminated between pull-based and push-based mechanisms. *Pull-based*, or passive, dissemination mechanisms rely on the agent to actively retrieve the updates and changes from the system e.g. manual looking at the activity feed or dashboard. *Push-based*, or active, dissemination mechanisms rely on the coordinator system and its subsystems to forward them to the agents in order to make them revisit the platform. A common example is the sending of emails with personalized reports about artifact updates to actors.

IV. Workflow Layer: The workflow layer is the third coordination tier and sets 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. Additionally layer IV provides rules and specifications to maintain and improve quality and quantity of data aggregated by CAs, as well as to increase the actor base, engagement of individual actors and activity level of artifacts.

4.2 Characteristic Process

A SIS has a characteristic process realizing of a perpetual feedback loop between a human actor basis and a reactive coordination infrastructure. The actors modify the content of CAs and the coordination infrastructure makes other actors aware of changes in the CA, which triggers those actors themselves to modify the content of the very same or other CAs. This interdependence between actor basis and coordination infrastructure creates a positive feedback loop with the CAs in its center continuously accreting content from actors.

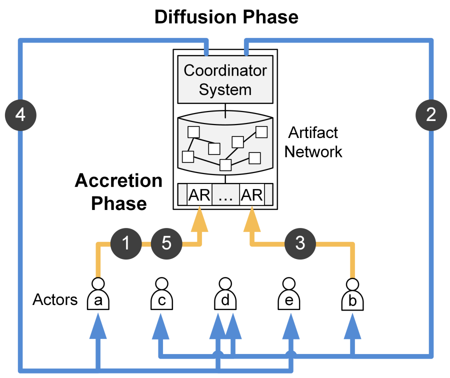

Fig. 2. Characteristic SIS process with alternating accretion phase (yellow) and diffusion phase (blue).

This process has the two phases of accretion and diffusion. Firstly, in the accretion phase the actor base accesses and modifies the CA content through the coordination infrastructure. Following the diffusion phase, where the coordination infrastructure uses active and passive dissemination mechanisms to make its actors aware about CA content changes and overall actor activity in the CA network. In a SIS process there is an interdependence between accretion (collection of content/knowledge) and diffusion (making others aware about content/knowledge/activity) resulting in a perpetual coordination cycle. Figure 2 depicts the characteristic process with accretion and diffusion phase and which consists of the following steps:

1. Actor a performs change action on coordination artifact A.
2. Coordinator system distributes artifact change increment across set of actors {b,d,c}. Actor b receives trigger about/discovers modified artifact A.
3. Actor b performs change action on coordination artifact A or on other artifact.
4. Coordinator system distributes artifact change increment across set of actors {a,d,e}. Actor a receives trigger about/discovers modified artifact.
5. Actor a performs change action on coordination artifact A.

This section introduced the Stigmergic Information System architecting approach and described its meta-model and its process model. The following section evaluates the approach regarding its feasibility in a practical scenario.

5 Evaluation

The feasibility of the SIS approach is demonstrated with a prototype of a social web application: a collaborative, online glossary system and the investigation of it's runtime behavior. The prototype was designed and developed by the authors and evaluated in the context of a research project on software inspection theory building at the Vienna University of Technology.

5.1 Prototype: Glossary System for Software Inspection

In research groups scientists from multiple domains with varying levels of expertise regularly need to collaborate on research projects in groups and communities towards a certain topic. Group members are not aware about what concepts the other members know and what meaning each member associates with each concept. Thus, a glossary is usually used to provide a list of relevant terms with their definitions in the context of the research project as common foundation. But the identification and mutual agreement of context-relevant terms and suitable definitions that fits for all researchers and their proper integration together with their definitions in a structured and consistent format to provide entry completeness and a common foundation still remain a challenge.

Therefore, we provided a collaborative, online glossary that increases awareness among agents about ambiguity of term definitons and ongoing activities, facilitates definition decisions and term agreements and disseminates changes made by others to the researchers in order to create collaboratively a common basis of relevant terms and accepted definitions in a specialized field of knowledge. This solution of a collaborative glossary system has been prototypically implemented as a SIS focused on the field of software inspection.

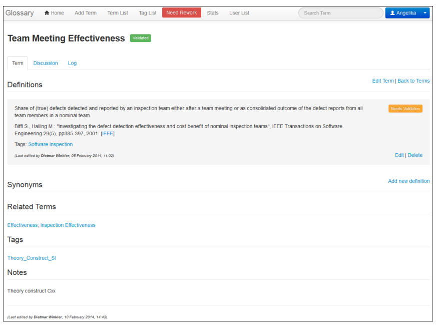

Fig. 3. Screenshot of the current glossary prototype, showing the term *Team Meeting Effectiveness* with its assigned definitions and attributes.

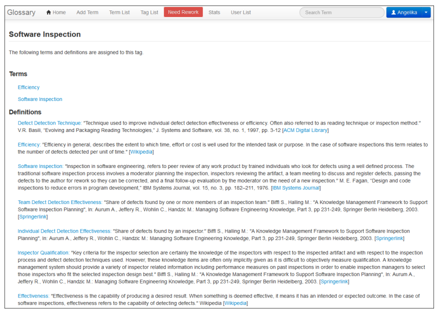

Fig. 4. Screenshot of the glossary prototype, showing the tag *Software Inspection* that links several terms and their definitions.

We started describing the business process of a simple glossary to identify current weaknesses that need to be addressed by the system architecture design and the workflow definitions. During system design the proposed meta-model was used for the analysis and construction of the architecture model. The system protoype was developed using Ruby on Rails and is hosted by the PaaS provider Heroku. The first contributions were made by researchers in Sept. 2013. Since the glossary system continues to be used by the researchers, further system extensions and improvements are implemented continually. Figure 3 and 4 show screenshots of the current prototype version of the glossary[3].

The glossary system prototype is a typical SIS with accretion of terms and appropriate definitions as well as diffusion of term changes across the researchers. In the following the four layers of the glossary system based on the meta-model are described in detail.

Workflow Layer To realize the SIS process and thus maintain the perpetual feedback loop, workflows need to be defined for the glossary system. One example workflow is illustrated in figure 5 and describes the accretion and diffusion phase triggered by an actor activity. It is documented using the Business Process Model and Notation (BPMN) standard[4]. To support a term validation process we introduced quality states: *not validated*, *needs rework*, *needs validation*, *validated*). The transition events between them are defined in different workflows. Triggered by an actor's modification of a term, i.e. the CA, by updating the quality status from "not validated" to "needs rework", a new process cycle starts. Thereby a new actor activity is recorded and the CA is updated according to the actor's changes. The CA modification leads to the creation of a new CA revision. All changes in the CA network are monitored by a data analysis unit in the coordinator system. By applying defined meta heuristics, result sets are calculated and consequently the system activity status is updated to present the results to the agents (dissemination mechanism). Thereby the system passively disseminates the artifact change increment across the actors, triggering further contributions of actors.

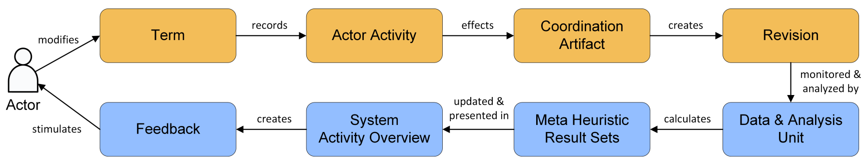

Fig. 5. Example workflow (in BPMN notation) defined for the glossary system, where an actor updating the quality status of a term starts one cycle of the characteristic process with accretion (yellow) and diffusion (blue) phase.

[3] http://glossary-sis.herokuapp.com

[4] http://www.bpmn.org/

Agent Layer The actor base of the glossary consists of researchers collaborating in a research project and only members of this defined group are able to modify CAs after signing in. Observers are all visitors of the glossary application without signing in and have read-only access to the glossary's content.

Artifact Data Layer The glossary's coordination artifact is a term. An overview of the CA's characteristic attribute tuples and the actor record is given in figure 6 where details of the database model are illustrated. A term has at least one definition with a reference of the source, where the definition can be found. Additionally, a term can have a quality status and notes. Terms can be connected via tags (indirect bidirectional linking), synonyms (bidirectional linking) and related terms (unidirectional linking) which leads to the creation of a CA network. After each CA modification a new revision of a term is created in order to keep track of an artifact's changes. The actor record consists of an actor's account information as well as the actor's activities throughout the glossary system.

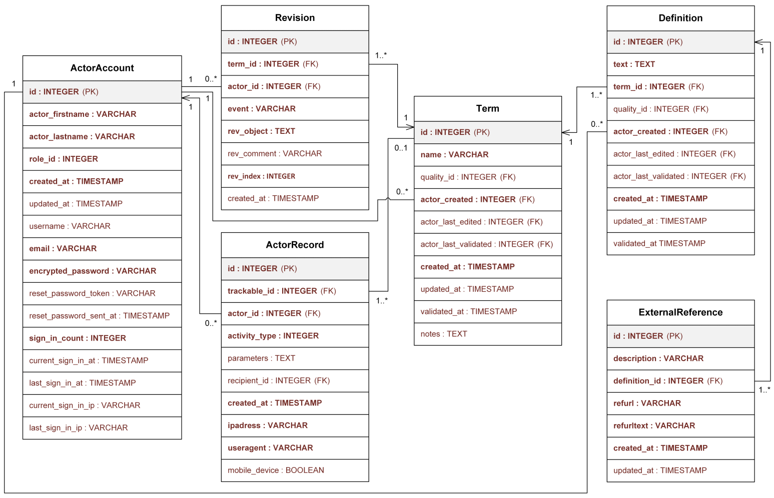

Fig. 6. The glossary's database model (in UML notation) consisting of the CA's main concept *term*, its attributes and the actor record as well as their relations in the glossary system.

Analysis & Control Layer In its current state, the glossary has only pull-based dissemination mechanisms implemented. The actors need to actively browse through the content of the CAs as well as ongoing activities. This immediate feedback of activities and changes is presented by the system in form of timestamps on each term, revision logs, a set of generated lists (term list, tag list, list of CAs that need rework, list of recently changed terms and definitions) and a system activity status overview. To support the term validation process we introduced a form of rating using quality states.

5.2 System Analysis

To get an assessment of the glossary's runtime behavior, a systematic analysis of the critical system elements and parameter is necessary. The results can be used by a software architect as deliberation basis for making further design decisions. The data used for the analysis were collected between Sep. 2013 and Feb. 2014. In this timeframe 7 researchers actively used the glossary system, whereby only 5 of them also edited CAs. By creating terms and linking them via tags (indirect bidirectional), via synonyms (bidirectional) and via related terms (unidirectional) a CA network has been created comprising 82 vertices (CAs) and 786 edges (artifact links). Table 1 presents a summary of general metrics about the system's behavior.

Table 1. Summary of general system metrics about the overall system growth.

General Metrics	Total Number
Actors involved in system activities	7
Actors involved in artifact activities	5
Coordination artifacts (vertices)	82
Bidirectional artifact links (edges)	5
Unidirectional artifact links (edges)	19
Indirect bidirectional artifact links (edges)	762

We identified the actor records, the coordination artifacts and the activities as critical system elements based on the SIS meta-model. Therefore, we investigate the activities performed by all actors and logged in the actor records as well as the activities on the CAs in more detail in order to derive an assessment of the system dynamics. To obtain the latest state of the system's runtime behavior we examine all activities made in the last month (Jan.-Feb. 2014).

Artifact Activities: Artifact activities include all *creates* of a new CA and *edits* of a given CA. The actors performed a total number of 137 editing activities across all CAs during the particular time slice. The resulted distribution of edits per artifact is illustrated in figure 7.

To measure the inequality among values of a frequency distribution the most common metric is the Gini coefficient. A value of 0 represents a complete equality whereas a value of 1 represents a complete inequality. The calculated Gini coefficient of the frequency distribution of all artifact edits is 23.34 % and indicates a low inequality of editing activities among the CAs. The maximum number of artifact edits on one day was 67, half of the total number of artifact edits, which implies highly infrequent artifact editing activities. The maximum number of edits on one single artifact was 7 which shows a low level of actor contributions accretion.

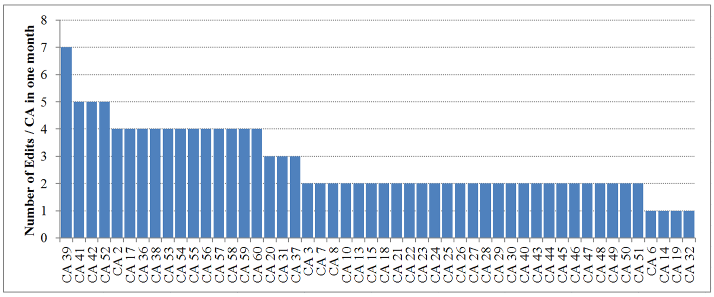

Fig. 7. Frequency distribution of edits per artifact during the defined interval in the glossary system. It also shows the most *wealthy* CA.

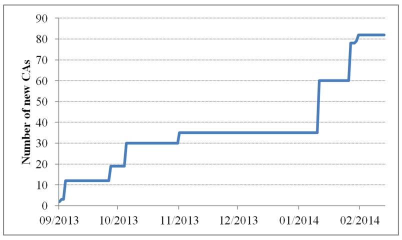

Fig. 8. Growth of coordination artifacts (terms) since Sept. 2013.

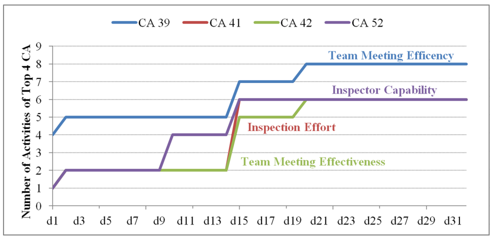

Fig. 9. Accretion of artifact edits of the Top 4 CAs during the defined interval.

Actor Activities: For the examination of actor activities, data from the actor records were extracted. Actor activities include all *artifact creates*, *artifact edits*, *artifact visits*, *tag visits (artifact links)*, *sign in* and *sign out*. The actors performed a total number of 867 activities across the system during the particular time slice. Figure 10 illustrates the infrequently and low engagements of the actors to the glossary system.

To describe the actor engagement we correlate the actor activities on a micro level, i.e. on a per actor basis, with the macro level across all actors in the system. The average number of artifact edits performed by Actor 1-5 (*values: 0.1, 1.8, 0.1, 1.2, 1.1*) is significantly lower compared to the average number of artifact edits performed by all actors (*value: 4.3*) during the defined interval. This implies a current low individual actor engagement in artifact edits accretion, highlighting a need for futher enhancement.

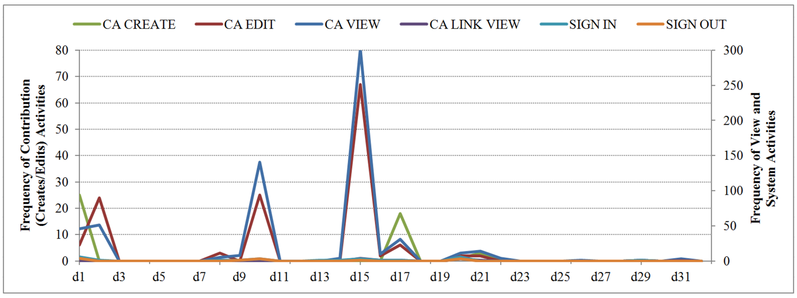

Fig. 10. Overview of all actor activities per day throughout the system in one month, separated in 6 activity types. On the left axis the frequency of all artifact creates and edits are presented, the right axis shows the frequency of all views (artifact views and artifact link views) as well as the system activities (sign in, sign out).

6 Discussion

In the prototype scenario of the glossary, the applicability of the SIS meta-model proved satisfactory for the description of the architecture design. The approach provided design guidance during the development of a new collaborative social web application, which has different features than a common wiki. The tailored structure of the glossary's coordination artifact, allows that the collected terms and associated defintions can be provided to other systems (e.g. via REST API and JSON) without additional processing effort. Also we were able to investigate the system behavior with statistical analysis and identify limitations and implications for the future glossary design in correlation with the SIS meta-model.

Lessons Learned: The glossary in its current form implements functionality in the accretion phase but lacks capability in the diffusion phase. The measured results of low artifact activity level and low individual actor engagement corroborate the role of the diffusion phase to maintain the SIS process cycle and are consistent with findings from previous studies [29, 12]. Therefore it is necessary to implement accretion and diffusion functionalities to equal parts. Although this increases development effort it is essential to create a closed reinforcement loop. In detail we identified the following design steps for improvement: 1) Implementation of push-based dissemination mechanisms, e.g. email notifications and updates about weakly activities. 2) Abilitiy to subscribe to individual terms in order to receive notifications about edits of the term and its definitions. 3) Modification of the current pull-based mechanisms to add generated lists in order to make the actors aware of highly active artifacts and artifacts that need improvement. After diffusion functionality is added to the system, it is also advisable to increase the actor base. Though smaller groups are suitable for piloting prototypes, like in the current case, the correct scaling of the system behavior can only be examined with larger actor bases. Strategies to increase the actor base could be to recruit prospective actors through the researcher, who are already in the glossary (member-gets-member), or to open the glossary for a global community.

Limitations: So far, the presented approach is in an early stage and has been evaluated only in the context of a glossary system. In order to demonstrate wider applicability of the approach implementations of additional application types - new as well as established ones like social networks and crowdsourcing platforms need to be done. Also to show that the meta-model is capable to describe also existing systems from the field, studies with systems already used for research like Wikipedia, Facebook or Twitter need to be conducted. A limitation of the measurement is, that only a limited set of metrics has been applied.

7 Conclusions & Future Work

In this paper introduced the Stigmergic Information System approach as a feasible architecting approach for the implementation-agnostic design of mediated interaction in social web applications. The SIS approach is described by a meta-model that organizes a system in the four layers of (1) agent, (2) artifact data, (2) analysis & control and (4) workflow, and applys a characteristic process consisting of an accretion and diffusion phase. We successfully evaluated the feasibility of the approach with a prototypical design and implementation of a glossary system by illustrating the mapping of the meta-model elements to the system elements. Also we measured the system's runtime behavior and identified limitations, which we analyzed with the introduced meta-model.

In future work we continue focussing on the support for the systematic design and analysis of social software architectures. Next research steps include (1) conducting a comprehensive survey of existing socio-technical systems to investigate their design and characteristic process as well as (2) extending our prototype with push-based dissemination mechanisms to investigate the effects on artifact activity and actor engagement. (3) Also we plan to design and implement more workflows for the research prototype to improve the CA content quality and the measurement of workflow effectiveness.

Acknowledgments. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria.

References

1. Babaoglu, O., et al.: Design patterns from biology for distributed computing. ACM Transactions on Autonomous and Adaptive Systems 1(1), 26–66 (Sep 2006)
2. Bedau, M.A., Humphreys, P.: Emergence: Contemporary Readings in Philosophy and Science. MIT Press (2008)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press (1999)

4. Brun, Y., Serugendo, G., Gacek, C., Giese, H., Kienle, H., Litoiu, M., Müller, H., Pezzè, M., Shaw, M.: Engineering Self-Adaptive Systems through Feedback Loops. In: Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525, pp. 48–70. Springer (2009)
5. Ciancarini, P.: Coordination Models and Languages as Software Integrators. ACM Computing Surveys 28(2), 300–302 (Jun 1996)
6. Dorn, C., Taylor, R.N.: Architecture-Driven Modeling of Adaptive Collaboration Structures in Large-Scale Social Web Applications. In: Proc. of the 13th Int'l Conf. on Web Information Systems Engineering (WISE '12). pp. 143–156. Springer (2012)
7. Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R.: The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. ACM Computing Surveys 12(2), 213–253 (Jun 1980)
8. Esparcia, S., Argente, E.: A Functional Taxonomy for Artifacts. In: Proc. of the 5th Int'l Conf. on Hybrid Artificial Intelligence Systems (HAIS '10). pp. 159–167. Springer (2010)
9. Franklin, S.: Coordination without Communication (1996), `www.msci.memphis.edu/~franklin/coord.html`
10. Gelernter, D.: Generative communication in Linda. ACM Transactions on Programming Languages and Systems 7(1), 80–112 (Jan 1985)
11. Gelernter, D., Carriero, N.: Coordination Languages and Their Significance. Communications of the ACM 35(2), 96 (Feb 1992)
12. Girgensohn, A., Lee, A.: Making Web Sites Be Places for Social Interaction. In: Proc. of the ACM Conf. on Computer Supported Cooperative Work (CSCW '02). pp. 136–145. ACM (2002)
13. Grassé, P.P.: La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie: Essai d'interprétation du comportement des Termites constructeurs. Insectes Sociaux 6(1), 41–80 (1959)
14. Hong-Linh, T., Dustdar, S., Bhattacharya, K.: Conceptualizing and Programming Hybrid Services in the Cloud. Intl. Journal on Cooperative Information Systems 22(4), 1–27 (2013)
15. Minder, P., Bernstein, A.: CrowdLang: A Programming Language for the Systematic Exploration of Human Computation Systems. In: Proc. of the 4th Int'l Conf. on Social Informatics (SocInfo '12). pp. 124–137. Springer (2012)
16. Mordinyi, R., Kühn, E.: Coordination Mechanisms in Complex Software Systems. In: Next Generation Data Technologies for Collective Computational Intelligence, vol. 352, pp. 3–30. Springer (2011)
17. Musil, J., Musil, A., Biffl, S.: Elements of Software Ecosystem Early-Stage Design for Collective Intelligence Systems. In: Proc. of the Int'l Workshop on Ecosystem Architectures (WEA '13). pp. 21–25. ACM (2013)
18. Musil, J., Musil, A., Winkler, D., Biffl, S.: 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). pp. 69–73. ACM (2012)
19. Nebeling, M., Leone, S., Norrie, M.C.: Crowdsourced Web Engineering and Design. In: Proc. of the 12th Int'l Conf. on Web Engineering (ICWE '12). pp. 31–45. Springer (2012)
20. Omicini, A.: Nature-Inspired Coordination Models: Current Status and Future Trends. ISRN Software Engineering 2013 (2013)

21. Omicini, A., Contucci, P.: 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, vol. 8083, pp. 1–10. Springer (2013)
22. 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 (Dec 2008)
23. Papadopoulos, G.A., Arbab, F.: Coordination Models and Languages. Advances in Computers 46, 329–400 (1998)
24. Quinn, A.J., Bederson, B.B.: Human Computation: A Survey and Taxonomy of a Growing Field. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI '11). pp. 1403–1412. ACM (2011)
25. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: 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, vol. 4389, pp. 124–140. Springer (2007)
26. Roman, G.C., Cunningham, H.C.: Mixed Programming Metaphors in a Shared Dataspace Model of Concurrency. IEEE Transactions on Software Engineering 16(12), 1361–1373 (Dec 1990)
27. Schmidt, K., Simone, C.: Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. Computer Supported Cooperative Work (CSCW) 5(2-3), 155–200 (Dec 1996)
28. Susi, T., Ziemke, T.: Social Cognition, Artefacts, and Stigmergy: A Comparative Analysis of Theoretical Frameworks for the Understanding of Artefact-mediated Collaborative Activity. Cognitive Systems Research 2(4), 273–290 (Dec 2001)
29. Tiwana, A., Bush, A.: A social exchange architecture for distributed Web communities. Journal of Knowledge Management 5(3), 242–249 (2001)
30. Van Dyke Parunak, H.: A Survey of Environments and Mechanisms for Human-Human Stigmergy. In: Proc. of the 2nd Int'l Conf. on Environments for Multi-Agent Systems II (E4MAS '05). LNCS, vol. 3830, pp. 163–186. Springer (2006)
31. 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 (Feb 2007)