An Architecture Framework for Collective Intelligence Systems

Juergen Musil*, Angelika Musil*, Danny Weyns[†] and Stefan Biffl*
*CDL-Flex, Institute of Software Technology and Interactive Systems
Vienna University of Technology, Vienna, Austria
Email: {jmusil, angelika}@computer.org, stefan.biffl@tuwien.ac.at
[†]AdaptWise Research Group
Linnaeus University, Växjö, Sweden
Email: danny.weyns@lnu.se

Abstract—Collective intelligence systems (CIS), such as wikis, social networks and content sharing platforms, have dramatically improved knowledge creation and sharing at society level. There is a trend to exploit the stigmergic mechanisms of CIS also at organization/corporate level. However, despite the wide adoption of CIS, there is a lack of consolidated systematic knowledge of the architectural principles and practices that underlie CIS. Software architects lack guidance to design CIS for the application context of individual organizations. To address these challenges, we contribute with an architecture framework for CIS, aligned with ISO/IEC/IEEE 42010. The CIS-AF framework provides guidance for architects to describe key CIS elements and systematically model a CIS that is well-suited for an organization's context and goals. The framework is grounded in an in-depth analysis of existing CIS, workshops and interviews with key stakeholders, and experiences from developing a prototypical CIS. We evaluated the architecture framework in two cases in industry setting where CIS have been designed and implemented using the framework. Results show that the framework effectively supports stakeholders with providing a shared vocabulary of CIS concepts, guiding them to systematically apply the stigmergic principles of CIS, and supporting them with kickstarting CIS in their organizations.

Keywords—*Architecture framework, architecture viewpoint, collective intelligence, coordination, software architecture, stigmergy.*

I. Introduction

Collective intelligence systems (CIS), such as wikis, social networks and content sharing platforms, have dramatically improved knowledge creation with more effective information aggregation and dissemination to benefit human collaboration. Central to CIS is stigmergy, that 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. For example, Wikipedia, created and maintained by users all over the world evolves 24/7 by users that add and modify its pages.

Traditionally, CIS are used at society or community level. There is now a trend to exploit the stigmergic mechanisms of CIS also at organization or corporate level. Examples of such CIS are corporate wikis[1] and corporate social net-

working services[2]. However, despite the wide adoption of CIS, there is a lack of consolidated systematic knowledge of the architectural principles and practices that underlie CIS. Furthermore, software architects lack guidance to design CIS for the application context of individual organizations. Our experiences with industry partners show that architects and developers interested in CIS have to resort on trial and error, or clone and own from similar successful CIS at best.

To address the problems with engineering CIS at organization level, we contribute with an architecture framework for collective intelligence systems (CIS-AF), aligned with the ISO/IEC/IEEE 42010 standard [1]. The standard defines an architecture framework as "the conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders." Applied to our context, the CIS-AF defines the key principles of CIS and provides guidance to architects, owners, developers and users to describe and systematically design CIS that are well-suited for the context and goals of organizations. As such, the focus of the framework is on the realization of CIS at organization level, from inception to operation.

The architecture framework is grounded in an extensive analysis of existing CIS. We looked at well established CIS to derive key elements (stakeholders, processes, components, behavior, etc.) to identify the common foundational principles of CIS. Furthermore, we organized a series of focus workshops with architects and business representatives of organizations with an interest in CIS, and performed semi-structured interviews with developers interested in implementing CIS. These interactions with key stakeholders provided invaluable input to pinpoint the key concerns of CIS and how they can be framed in models and analysis. In addition, we drew from our own experience with developing a non-trivial prototypical CIS.

The CIS-AF comprises three complementary viewpoints together with correspondence rules that express relations across the viewpoints. The context viewpoint describes the conventions to derive architecture views that frame the usefulness and perpetuality[3] concerns of architects, owners and actors that use the system. The technical realization viewpoint describes the conventions to derive architecture views that frame the

---

data aggregation, knowledge dissemination, and interactivity concerns of architects, owners, builders, and actors. Finally, the operation viewpoint describes the conventions to derive architecture views that frame the kickstart and monitoring concerns of system managers and analysts of CIS. It is important to note that the focus of the architecture framework and its viewpoints is on *CIS-specific concerns of the realization of CIS*. To deal with traditional concerns, such as performance, availability or scalability, that may be relevant to the realization of CIS, additional viewpoints or other architectural approaches can be used to support the stakeholders.

We evaluated the architecture framework in two cases in industry setting where CIS have been designed and implemented using the framework. The CIS in the first case is a Reuse Center (RUC), which enhances an IDE with facilities for programmers to suggest and reuse code snippets during the software development process. The RUC was developed in an R&D effort between an Austrian company that develops tools for industrial automation systems and our own research group CDL-Flex at TU Wien. The CIS in the second case is the Feature Deliberatorium (FD), which enhances an industrial software ecosystem to build and customize automation solutions with facilities for supporting collective feature reviewing and consolidation. The FD was developed in a joint R&D project between a large Austrian company and an external research lab at Johannes Kepler University Linz. Results of these two cases show that the framework effectively supports stakeholders with capturing their CIS-specific concerns and establishing CIS is their organizations. In particular, the evaluation demonstrated that the framework offers a shared vocabulary of CIS concepts to the stakeholders, it guides them to systematically apply the stigmergic principles of CIS, and it supports them with kick-starting the CIS in their organizations.

The remainder of this paper is structured as follows. Section II introduces the basics of CIS. In section III we address the research questions and the research methodology we used. Section IV presents an overview of the architecture framework for CIS, comprising three viewpoints with their model kinds and the correspondence rules between the model kinds. The framework's applicability is evaluated in section V with two industrial cases. Section VI discusses related work. Finally, section VII draws conclusions and outlines future work.

II. CIS Characteristics

Collective Intelligence Systems (CIS) enable IT-mediated collective intelligence [2] and belong to the family of socio-technical systems. A socio-technical system is a hybrid system where "the active components are mostly represented by humans, whereas interaction is almost-totally regulated by the software infrastructure" [3]. Key characteristics of CIS are that they enable bottom-up information sharing and knowledge aggregation by combing the strengths of computing systems (data processing, workflow coordination) with the cognitive capabilities of human groups (abstract thinking, pattern recognition) [2]. CIS behavior is emergent, meaning that high-level, system-wide behavior 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 [4].
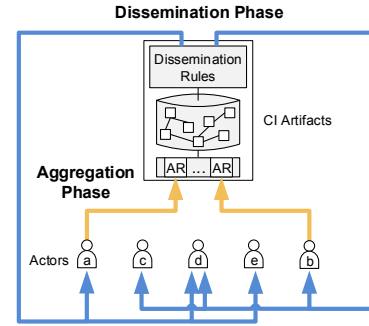

Fig. 1.   Stigmergic CIS process with aggregation and dissemination phases.

A CIS realizes a stigmergic process [5] of a perpetual feedback loop between a human actor basis and a reactive coordination infrastructure [6] as shown in Fig. 1. The actors modify the content of CI artifacts and their behavior is tracked in actor records (AR). The infrastructure makes other actors aware of changes of CI artifacts, which triggers those actors through dissemination rules to modify the content of the very same or other CI artifacts. This interdependence between actor basis and coordination infrastructure creates a positive feedback loop where the CI artifacts are in the center and are continuously accumulating content from actors. The process consists of two essential phases: aggregation and dissemination. Firstly, in the aggregation phase the actors access and modify the CI artifacts' content through the infrastructure. Following the dissemination phase, where the infrastructure uses active and passive dissemination mechanisms to make the actors aware about artifact content changes and overall actor activity in the system. In a CIS there is an interdependence between aggregation (collection of content/data) and dissemination (making others aware about content/knowledge/activity) resulting in a perpetual cycle that enables self-organization.

III. Research Questions and Approach

To tackle the engineering problems with realizing CIS at organizational or corporate level, and based on the related work and current best practices, we identified the following two research questions:

RQ 1 - What are the most important underlying architectural principles of collective intelligence systems?

RQ 2 - How can we codify (capture, document, structure, organize) these architectural principles to make them useful for engineering collective intelligence systems?

To answer these research questions we applied an iterative research approach in three phases. In the first phase, we collected knowledge. In particular, we identified the core elements of CIS, the key stakeholders and their architecture-related concerns, and we elicited model needs. The primary focus of the first phase is on answering RQ1. In the second phase, we synthesized knowledge. Concretely, we defined and documented a novel architecture framework for realizing CIS following the ISO/IEC/IEEE 42010 standard. Based on the inputs from the first phase, the framework comprises architecture viewpoints and correspondence rules that define conventions for the construction and use of architecture views to deal with
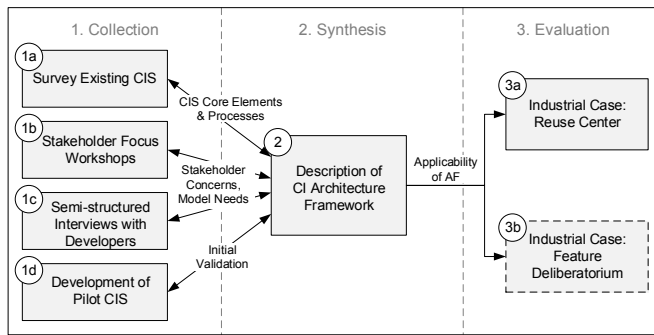
Fig. 2. Design of the multi-phase research method.

the identified CIS stakeholder concerns. The primary focus of the second phase is on answering RQ2. In the third phase, we validated the research results obtained in the previous phases. In particular, we evaluated the framework's applicability and effectiveness with two industrial cases. All material generated during the different phases of the research process is available at the Architecture Framework website [7].

Figure 2 provides a schematic overview of the activities in each phase.

The Collect phase consisted of four parts: (1) a survey of existing CIS, (2) focus workshops with architects and business people, (3) semi-structured interviews with developers, (4) development of a pilot CIS. The Synthesize phase consolidated the input from the Collect phase in form of the Architecture Framework for CIS. The Evaluation phase comprised two cases in an industrial setting: the RUC and the FD. We elaborate on the different steps of the Collect phase. Details of the other two phases are provided in the following sections.

A. Survey of Existing Systems

To identify the underlying principles of CIS, we performed a survey of existing CIS. Concretely, we used the Alexa website[4] to identify potentially interesting and popular CIS. In total, we identified 180 different CIS and analysed 30 of them in depth. Based on the usage interface and available material of this subset we identified six key features of CIS. The features are: the ability of actors to (1) add and (2) change domain items and (3) create links between data items, system support for (4) dissemination of selected state changes to actors and (5) user-driven recommendations, and finally (6) support for tracking of usage behavior of actors. Based on this set of features, we defined a meta-model for CIS in four layers from bottom to top: agent, data items, analysis & control, and workflow. For more information about the survey and the meta-model, we refer the interested reader to the Architecture Framework website and a related publication with preliminary results [6].

B. Stakeholder Focus Workshops

We organized three workshops with stakeholders from industry that had an interest in introducing CIS in there organizations. Each workshop considered a different domain: software development tools, design environments for building

architects, and knowledge management. The overall goal of the workshops was to get insights in how stakeholders perceive CIS and what their concerns are regarding the introduction of CIS in their organization. In each workshop five to eight stakeholders participated, typically domain experts, software architects, and business representatives. The workshops were led by researchers of our team and took between three and five hours each. A workshop started with a presentation of what CIS is about, illustrated with classic examples. Previous examples for the domain of interest are presented. The main part of the workshops was a discussion focusing on the expectations, stakeholder needs, the potential of CIS, impact and risks. A workshop concluded with a wrap up of concrete output. The main findings of the focus workshops were the identification of key concerns, including usefulness of CIS, aggregation and dissemination of knowledge, perpetuality, kick-start of the CIS, and monitoring the system in operation.

C. Semi-Structured Interviews

In addition to the focus workshops, we organized a series of 10 interviews with software developers. The overall goal of the interviews was to probe the concerns of developers with the development of CIS. The interviews were conducted by researchers of our team and took about 40 minutes each. An interview started with introducing a conceptual design of an example system (wiki-type of CIS for games). Subsequently, the interviewer asked the subject a set of guiding questions about how she or he would approach the concrete development of the example CIS, and what main implementation challenges they see. The interviews revealed important issues of developers regarding understanding CIS principles and how to implement them. On the one hand, developers consider the understanding of stigmergic feedback loop process and its implementation as the key challenge for the development of CIS. On the other hand, they tend to encapsulate (hide away) the core CI features in "intelligent" components to address CI needs assuming that integrating these components would be sufficient to address CI needs.

D. Development Pilot CIS

Armed with the knowledge we obtained from the other activities in the Collect phase, we decided to develop a pilot CIS. The main objective was to cross-check the collected knowledge and support the development of the Architecture Framework from first-hand experience. The pilot system is a Collaborative Glossary for Multidisciplinary Engineering that was developed by our research team. This CIS supports collaborating scientists potentially from multiple domains and geographically distributed with establishing and maintaining a shared online glossary of terms along with their definitions. We applied the glossary for the domain of multidisciplinary engineering[5]. From the development of the pilot we learned important CIS aspects. First, the effort stressed the importance of identifying process improvements from the outset. In particular, it is essential to consider the feedback mechanism of CIS from the start, as this mechanism is the central factor of the improvement of a CIS, but it requires a different way of thinking about processes. Second, the pilot learned us that

aggregation and dissemination should be considered as first class citizens. In particular, dissemination is non-trivial and requires analysis, a workflow, and well-defined stimuli. Third, bootstrapping a CIS is a dynamic process that takes time. The pilot learned us the importance of building initial content and monitoring system behavior after deployment.

IV. ARCHITECTURE FRAMEWORK FOR CIS

Based on the insights and knowledge we acquired in the first phase of our research (Collect, see Fig. 2), in the second phase we defined the architecture framework for collective intelligence systems (CIS-AF) aligned with the ISO/IEC/IEEE 42010 standard. The framework defines a set of three architecture viewpoints for building new CIS solutions: CI context viewpoint, CI technical realization viewpoint, and CI operation viewpoint. In addition, the framework comprises a set of correspondence rules that define relations between the different viewpoints. The three viewpoints cover the essential concerns of stakeholders with an interest in the realization of CIS at organization level. The focus of the framework is on CIS specific concerns. As such, architects may use additional architectural approaches, such as additional viewpoints or patterns to deal with other traditional stakeholder concerns.

The viewpoints are structured using the following template:

- Name: set of words to refer to the viewpoint

- Overview: short description of the viewpoint

- Stakeholders: individual, team, organization with an interest in the concerns of the viewpoint

- Concerns: interests in the CIS relevant to stakeholder(s); the viewpoint provides questions for each concern to help stakeholders framing their concerns

- Model kinds: conventions for a type of modelling

- Meta models: core constructs of the model kinds

- Analysis: methods to check, reason about, transform, predict, apply and evaluate architectural results from the views generated from this viewpoint

We start by listing the stakeholders of the architecture framework and highlight their roles. Then we present the three viewpoints. We use the template above to describe each viewpoint, discuss meta-models and analysis in detail, and illustrate them with excerpts of the Collaborative Glossary for Multidisciplinary Engineering (CGME) CIS. Finally, we provide an overview of the correspondence rules.

A. Stakeholders

Stakeholders of the architecture framework include *architect(s)* who design and describe the CIS architecture, *owner(s)* who define the CISs purpose and business goals, *manager(s)* who manage the CIS and operate it to provide the service to the actors, *builder(s)* who develop the CIS, *analyst(s)* who monitor the CIS and perform analysis of its behavior, and *actors* who access and contribute to the CIS. We use the term *users* to refer to all the stakeholders that interact with the system.
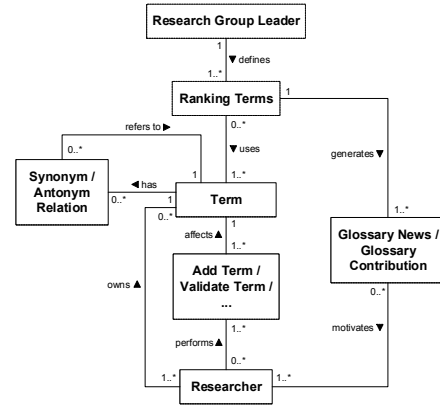

Fig. 3. Stigmergic Coordination Model for the CGME CIS

B. CI Context Viewpoint

Table I gives an overview of the CI context viewpoint.

1) As-Is Workflow Meta Model: Workflow refers to a series of Activities performed by Users with possibly different Roles. The Workflow has a particular Objective in the organization. Activities may be performed on a System. The workflow of the as-is glossary system consists of informal activities, such as adding a term to the glossary, agreeing on the definition of a term in a meeting, etc. The Objective of this system is obtaining a Common Understanding of Terms in the domain of multidisciplinary engineering. The System consists of a shared repository. The Users are Researchers in the Role of Contributor to the glossary and the Administrator that manages the repository.

2) Stigmergic Coordination Meta Model: Central to the meta model are Domain Items that may have Item Links to other items. Actors can perform Interaction Rules to affect Domain Items, while Owners define Management Rules that use the network of Domain Items to generate Stimuli to motivate Actors. Both push and pull mechanisms can be considered for Stimuli. Fig. 3 shows the instantiation of the Stigmergic Coordination Meta Model for the CGME CIS.
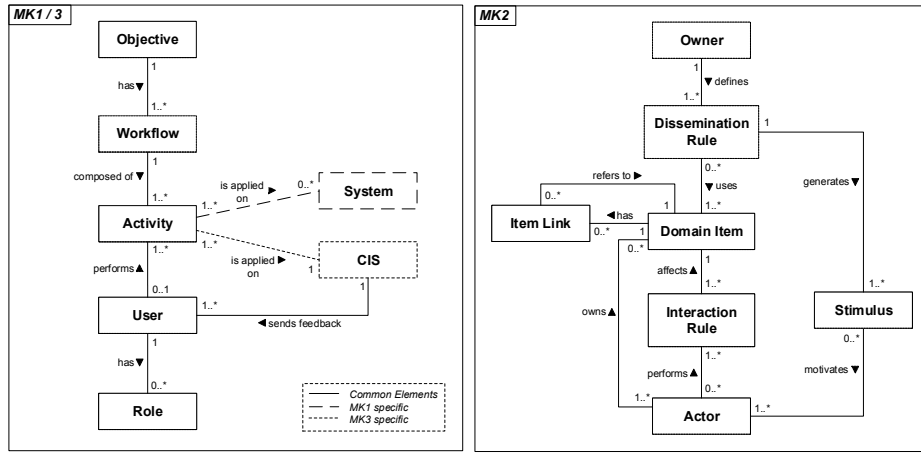
Domain Items for the CGME CIS are Terms of the glossary with a definition. Item Links are Term Links that express relationships between the terms, such as synonyms or antonyms. The CGME CIS has several Interaction Rules, such as Add Term, Link Terms, Validate Term, etc. that are performed by Researchers. CGME CIS has currently one Management Rule, which is Ranking Terms that ranks terms based on their activities by researchers. Ranking Terms generates two types of Stimuli of push type: Glossary News and Glossary Contribution. In particular, these stimuli are distributed by email every week to the researchers.

3) To-Be Workflow Meta Model: This meta model shares the model elements Workflow, Objective, Activity, User, and Role with the As-Is Workflow Meta Model. However, Activities in the To-Be Workflow Meta Model are applied on the CIS. Furthermore, the CIS sends feedback to the User. The CIS in the glossary system is the CGME CIS. Researchers get feedback from the CIS via Glossary News and Glossary Contribution emails (see also the Stigmergic Coordination Meta Model).

TABLE I.   CONTEXT VIEWPOINT FOR CIS

Name	CI Context Viewpoint (VP1)
Overview	The architecture viewpoint deals with the main stakeholder concerns related to the design of CI-specific system capabilities and defines models for the construction of new CIS and capture of design decisions. The models show the relevant architectural information to enable bottom-up, information aggregation, management and distribution capabilities for hard-to-access distributed and dispersed knowledge and information.
Stakeholders	*Architect(s)* who design and describe the CIS architecture. *Owner(s)* who define the CISs purpose and operate it to provide the service to the users. *Actors* who access and contribute to the CIS.
Concerns	*C1 - Usefulness:* How does the process look like that needs to be improved? Does the process have limitations that can be addressed with CIS capabilities? Is there a CIS that can improve a business-critical workflow / process of the organization? What is getting better in the daily routines of actors who use the CIS? *C2 - Perpetuality:* How could the stigmergic process of a planned CIS look like for the application scenario? How is the user involved in the system? Does the core feedback loop of the CIS make sense for the organization's workflow / business process?
Model Kinds	*MK1 - As-Is Workflow (deals with concern C1):* A model that describes the current workflow of interest in the organization with the activities performed by users and the existing system of interest. *MK2 - Stigmergic Coordination (deals with concern C2):* A model that describes the domain items of interest in the organization, the rules to interact with the domain items, and the feedback loop that provides stimuli to users. *MK3 - To-Be Workflow (deals with concern C1):* A model that describes the to-be workflow of interest in the organization with the activities performed by users and the CIS, along with feedback from the CIS to the users.
Meta-Models	

(1) As-Is Workflow / To-Be Workflow meta-model   (2) Stigmergic Coordination meta-model

Key: UML

| **Analyses** | *A1 - Comparative Process Analysis (using MK1 and MK3):* Compare how well the as-is workflow and the to-be workflow realize the objective. The results of this comparison support decision making whether a CIS could be beneficial or not compared to a non-CI system.
A2 - Scenario Analysis (using MK2): Simulate (manually or automatically) the stigmergic feedback loop using different scenarios with sets of actors bases, linked domain items, rules interaction and management rules, and stimuli. The results of the simulation support the understanding of the CI principles and give insights in the conditions for the conceptual feasibility of the CIS in the application context. |

4) Comparative Process Analysis: Comparing how well the objective of the glossary system is realized boils down to comparing the degree to which researchers obtain a common understanding of terms in the domain of multidisciplinary engineering without and with the CGME CIS. Given the ad-hoc approach that used in the as-is workflow, a formal comparison is difficult to achieve. Based on the expected benefits of the stigmergic feedback loop, the team decided to build the CGME CIS.

5) Scenario Analysis: Informal manual simulations with different scenarios of the glossary allowed the stakeholders to reason about the stigmergic feedback loop, quantify the initially required critical mass of users to launch the CGME CIS, and identify the initial parameter settings for ranking terms based on activities and the stimuli. Monitoring activities of the system in operation during the first year confirmed its feasibility in the application context.

C. CI Technical Realization Viewpoint

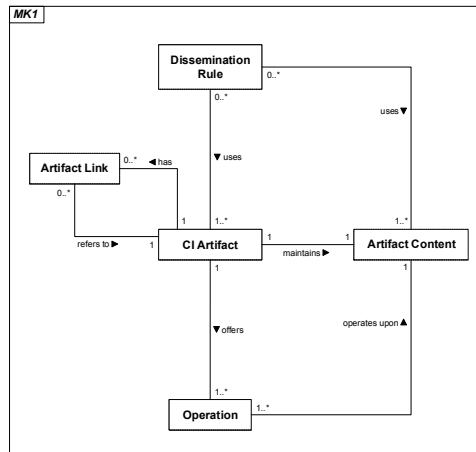Table II shows an overview of the CI technical realization viewpoint.

1) Artifact Definition Meta Model: The CI Artifact represents the central element of a CIS and specifies the Content that actors can manipulate via Operations, and the Dissemination Rules that access Artifact and Content generating feedback to Actors. CI Artifacts are liked through Artifact Links, thus creating a network structure. The CGME CIS CI Artifact contains the Term, its Definitions with a literature reference in text form. The Artifact Links in the CGME CIS are realized using Tags, Synonym- and Related-Term relationships, which are also in text form.

2) Aggregation Meta Model: Actors can perform Activities (Read/Write) to manipulate the CI Artifact using a software Client. Actors may have a Role. The Actor Record logs Actor
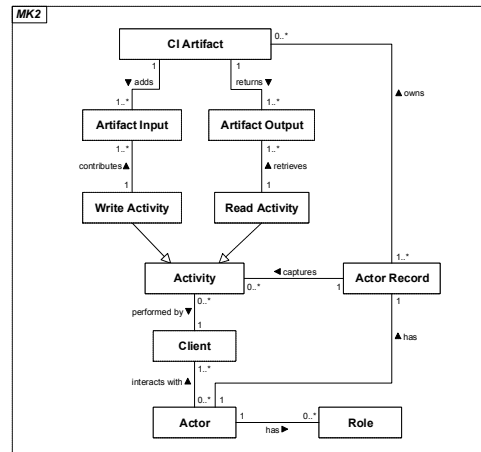
TABLE II.   TECHNICAL REALIZATION VIEWPOINT FOR CIS

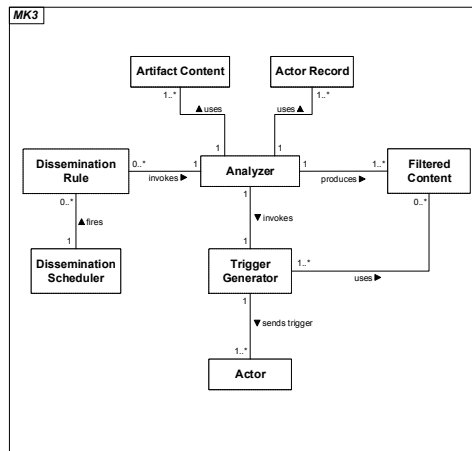| Name	CI Technical Realization Viewpoint (VP2)
Overview	The architecture viewpoint deals with the main stakeholder concerns related to the realization of the CIS and defines models to model collective knowledge, the aggregation of data and the stigmergy-based dissemination of knowledge. The models show the relevant architectural information that is essential to guide the concrete implementation of a new CIS for an organization.
Stakeholders	*Architect(s)* who design and describe the CIS architecture. *Owner(s)* who define the CIS's purpose and operate it to provide the service to the users. *Builder(s)* who implements the CIS. *Actors* who access and contribute to the CIS.
Concerns	*C1 - Data Aggregation:* What kind of data is needed from actors? How is the data collected? How is the data stored? *C2 - Knowledge Dissemination:* What data processing mechanisms are needed to effectively distribute relevant knowledge to the actors? How does the system keep users aware about the status of relevant content? How are dissemination rules executed? *C3 - Interactivity:* What interactions (activities) can actors do with the CIS? How are the actor interactions realized?
Model Kinds	*MK1 - Artifact Definition (deals with concern C1):* A model that describes how the CI artifact is realized and linked, and which operations apply to interact with the artifact content. *MK2 - Aggregation (deals with concerns C1, C3):* A model that describes what activities can be performed by the actors, what kind of data is aggregated from the actors, and to what extent these actor activities are captured. *MK3 - Dissemination (deals with concerns C2, C3):* A model that describes what content and how this content is disseminated and stimulates subsequent actor activities.
Meta-Models	 (1) Artifact Definition meta-model (2) Aggregation meta-model (3) Dissemination meta-model *Key:* UML
Analyses	*A1 - Aggregation Analysis (using MK1 and MK2):* Review if the aggregation activities produce the intended artifact content. The results of this review show inconsistencies between the defined artifact content and the content that is produced by the actor activities. *A2 - Dissemination Analysis (using MK1 and MK3):* Review if the intended dissemination content can be derived from the available content. The results of this review identify inconsistencies between the content that needs to be disseminated and the stored artifact content and actor record data.

Activities in the CIS. Each Actor has a single Actor Record, and each CI Artifact is owned by at least one Actor Record. In the CGME CIS, Researcher can perform CRUD and validation activities for Terms and their Definitions. All Terms in CGME CIS are owned equally by all Actor Records, which track CRUD and validation activities.

3) Dissemination Meta Model: Filtered Content is generated from Artifact Content and Actor Records by the Analyzer following Dissemination Rules fired by a Dissemination Scheduler. The Filtered Content is used by a Trigger Generator to generate Triggers that are distributed to individual Actors. The CGME CIS sends Researchers Weekly Digest Emails about new and updated Terms and Definitions and a Weekly Ranking Email that includes a leaderboard of the most active users.

4) Aggregation Analysis: Comparing the Artifact Definition and Aggregation models allows reviewing inconsistencies in CI Artifact manipulation. For the Terms in the CGME CIS it is important they all contain Definitions and that Definitions can only be validated by Researchers with Validator role.

5) Dissemination Analysis: Review whether the intended dissemination content can be derived from the Artifact Content and the Actor Record. In the CGME CIS dissemination analysis is for example required to check that the information that is presented in the Weekly Digest Email is available in the Actor Record.

D. CI Operation Viewpoint

Table III shows an overview of the CI operation viewpoint.

1) Initial Content Acquisition Meta Model: This meta model describes two important factors to kickstart the feedback loop of a CIS: (1) Sources of Initial Data and its Transformations into CI Artifact's Content, and (2) Initial Actor groups. In the CGME CIS a spreadsheet with a set of Definitions was imported using a function from the glossary's underlying database system. Initial actors were recruited from Lab staff and fellow Researchers.

2) CI Analytics Meta Model: The meta model describes CIS measurement design for Analysts to generate Metrics and Analysis Results according to Measurement Profiles with Probes located in System Components. A metric in the CGME CIS is Term Views by individual Users, which are tracked by Tracker probe at each page load in the Controller component.

3) Minimum Content Quality Analysis: Measure minimum quality criteria of Artifact Content that is migrated from other sources or created by actors. Quality criteria in the CGME CIS are completeness of attribute elements of terms and consistency of links between terms, which were manually checked.

E. Model Correspondence Rules

Table IV gives an overview of relations between the model elements of the three viewpoints. VP1 refers to the Context VP, VP2 to the Technical Realization VP, and VP3 to the Operation VP. The x in MKx refers to the numbering of the respective model kinds in the VP tables. The table does not show equality relations, which apply to all elements with the same name.

In addition, Actor (VP1-MK2, VP2-MK-2), Owner (VP1-MK2), and Analyst (VP3-MK2) specialize User (VP1-MK1/3).

TABLE IV.   OVERVIEW CORRESPONDENCE RULES BETWEEN MODEL KIND ELEMENTS

ID	AD Element	Relation	AD Element
CR1	VP1-MK2 Interaction Rule	refines	VP2-MK2 Activity
CR2	VP2-MK1 Operation	refines	VP1-MK2 Interaction Rule
CR3	VP2-MK1 CI Artifact	realizes	VP1-MK2 Domain Item
CR4	VP1-MK2 Interaction Rule	constraints	VP2-MK2 Activity
CR5	VP1-MK2 Item Link	refines	VP2-MK1 Artifact Link

V. EVALUATION

We performed a qualitative evaluation of the architecture framework in two industry cases. The evaluation of the framework had the following goals:

1) To what extent are the included viewpoints sufficient to describe the core CI-specific concerns for realizing a CIS in an organization context?

2) How are the viewpoints used to produce the architectural design of CIS?

3) How useful and understandable are the viewpoints and models to architects?

For a detailed description of the qualitative evaluation, we refer the reader to [7].

A. Procedure

In each case we took the role of observer of a team of two architects over a period of 12 months to observe how they use the framework to create an architecture for a CIS in an industry setting. Observation (complemented with restricted interactions with the teams) consisted of:

1) Regular visits, typically once a month, where the architects reported on their status and experienced limitations and where we provided feedback on general CI concerns.

2) Attendance of stakeholder workshops, to observe how the architects used the viewpoints to explore concerns, interact with stakeholders, and derive design models.

3) Regular evaluations of the created designs, where we also asked the architects questions about their experiences and difficulties with creating the respective views and models.

B. Industry Cases

We give a brief description of the two cases. For a detailed description and additional material we refer the interested reader to [7].

Case 1: Reuse Center - The Reuse Center (RUC) is a CIS that was developed in a joint R&D project between an Austrian company partner, who is a provider of software development tools for industrial automation systems, and CDL-Flex at TU Wien. The company has 20+ years experience in developing programming tools and platforms for the automation industry. The RUC complements an IDE product that is used for

TABLE III.   OPERATION VIEWPOINT FOR CIS

Name	CI Operation Viewpoint (VP3)
Overview	The architecture viewpoint deals with the main stakeholder concerns related to the startup of the CIS operation and defines models to identify initial content and actor groups and define indicators to measure CIS aggregation and dissemination performance. The models show the relevant architectural information that is essential to guide a successful startup of the perpetual feedback loop of a new CIS for an organization.
Stakeholders	*Manager(s)* who is responsible for overseeing CIS operation. *Analyst(s)* who assess the performance of a CIS.
Concerns	*C1 - Kickstart:* What is the profile of an initial actor group that fits to the CIS? How can existing data be transformed into initial content for the CIS? *C2 - Monitoring:* What properties are needed to monitor to know whether the system is working properly? What metrics are needed therefore? How are these metrics monitored?
Model Kinds	*MK1 - Initial Content Acquisition (deals with concern C1):* A model that describes the sources from which initial content for the artifacts can be migrated and potential groups of initial actors to build up an actor community. *MK2 - CI Analytics (deals with concern C2):* A model that describes relevant metrics for the CIS and probes to capture the data necessary for calculating the metrics.
Meta-Models	 (1) Initial Content Acquisition meta-model (2) CI Analytics meta-model *Key:* UML
Analyses	*A1 - Minimum Content Quality Analysis (using MK1 and MK2):* Determine metrics that measure minimum quality criteria of an artifact's content that is migrated from other sources or created by the actors. The results of this determination support the initial population of the CIS with content.

programming PLCs (Programmable Logic Controllers) with the programming language ST (Structured Text) according to IEC 61131-3 industry standard. The RUC's objective is to introduce CI capabilities into the automation software development process by enabling ad hoc source code reuse through collectively sharing, modifying and reviewing of code snippets within development teams.

Case 2: Feature Deliberatorium - The Feature Deliberatorium (FD) is a CIS that was developed in a joint R&D project between a large Austrian company, who builds and operates industrial automation systems, and a research lab at Johannes Kepler University Linz. The architecture researchers used our AF to independently design a CIS prototype for their industry partner. The company maintains an internal industrial software ecosystem (ISECO) to build and customize automation solutions for their customers. The ISECO consists of the three tiers: platform (top), domain solutions (middle) and applications (bottom). An inherent challenge in this setting is the upward propagation of implemented, new features from the application tier to the domain solution tier due to ineffective sharing, consolidating and tracking of technical knowledge. Propagating application features into the domain solutions for integration into the platform is important to maintain feature consistency between tiers and counteract architecture erosion. The FD's objective is to introduce CI capabilities into the ISECO by supporting collective feature reviewing and consolidation within and across application and domain solution tiers.

C. Results

From the observations and input we received from both cases, we learned that the understanding and competency in CI principles of the stakeholders - architects in particular - gradually increased. We noticed that the framework effectively provides stakeholders with a shared vocabulary of CIS concepts, guides them to systematically apply the stigmergic principles of CIS, and supports them with kickstarting CIS in their organizations. Whereas stakeholders relatively easily grasp the general idea of CIS - the stigmergic mechanism - we noticed that the technical realization of this principle requires guidance, which the architecture framework provided. We highlight experiences with the model kinds we observed.

The Stigmergic Coordination meta model (CI Context VP) was particularly helpful for architects to keep the "big picture" throughout the architecting process, blending the global design concepts of the architectural framework together with the intended solution concept without going into technical details. Fig. 4 shows an excerpt of a simplified stigmergic coordination model for the RUC created in a working session. The model shows the key elements of the CIS, including the users (ActorBase), the interaction mechanisms for the users, the artifact network, the types of analysis and stimuli.

The As-Is and To-Be Workflow meta models (CI Context VP) were actively used by the stakeholders in the starting phases of both projects to identify the potential benefits of introducing CIS. In both cases, the workflows where further refined with concrete activities in the specific domains. Further
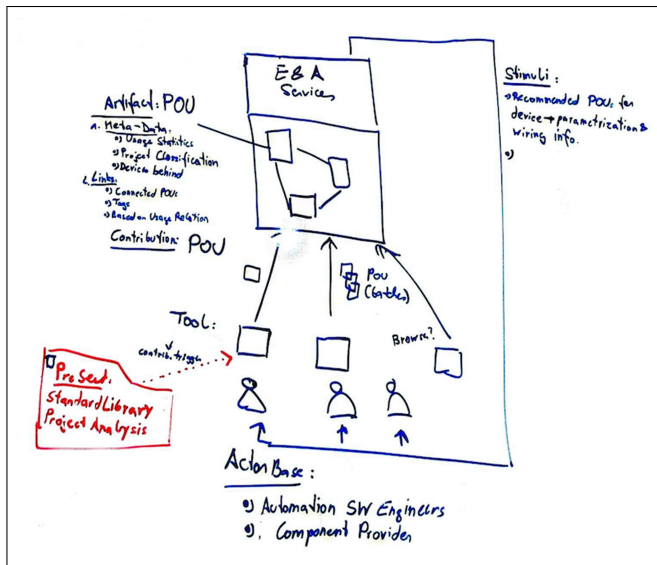
Fig. 4. Sketched instantiation of the Stigmergic Coordination Model Kind during a work session in the RUC case.

on, we observed that architects used the models derived from these model kinds to communicate workflow improvements with the to-be-CIS to other stakeholders, in particular from management/business areas.

We observed that the Artifact Definition meta model (CI Technical Realization VP) took the stakeholders some time to familiarize themselves with the concept and mechanism of the CI artifact. We noticed that getting the artifact "right" with respect to its type and attributes was considered an important success factor in both cases. In the FD case, the artifact definition took longer than in the RUC case, since the architects needed to choose from three artifact candidates, with a significant overlap in attributes.

In both cases, we observed that the Aggregation and Dissemination VPs (CI Technical Realization VP) where less actively used by the architects during the design. In fact, these model kinds turned out to be very useful for re-engineering efforts, after initial deployment, when problems with the efficiency of the CIS were identified. In line with this, we observed that handling the concerns covered by the CI Operation VP was also postponed until after initial deployment, when limitations of the CIS were discovered such as lack of sufficient artifact content and poor working stimuli.

In both cases, the architects extended the architectural description with additional UML diagrams. Architects of both cases including additional component and deployment models. In the FD case, the architects specified additional use case documents, driven by the company's internal documentation practices. As a result of the models derived from the CI Context and CI Technical Realization VP, the architects reported an effort reduction of creating the use case document by about 50%; the content provided by the models of these viewpoints covered the remaining part.

D. Discussion

The cases show that the architecture framework effectively supports stakeholders with creating designs and realizing CIS in their organizations. Although, the architects had the challenge to design an unfamiliar type of system with a framework they have not used before, they let themselves guide by the framework and design principles. Overall, we noticed that the framework offers a mindset and constrains to guide stakeholders to systematically apply the principles of CIS in the realization of their CIS. As explained above, we observed that some of the model kinds turned out to be active working instruments during the design of CIS, while other model kinds were picked up in a later stage, actually to support handling problems after initial deployment. In both cases, the set of viewpoints cover well the CI specific concerns of the stakeholders when realizing CIS in their organizations.

We report some lessons learned from the cases. First, a challenge of designing CIS is that architects need to address multiple concerns that are not or only partially covered by available viewpoints and frameworks. The architecture framework helps architects to focus on the basic elements and processes and systematically guides them to create CIS architecture descriptions even for complex application domains (code reuse, feature management), thus catalyzing innovation in CIS platforms. Second, documenting software architectures that have properties with which software architects are typically not familiar, such as the emergent stigmergy coordination process that is realized by CIS, is particular difficult for software architects because they need also to familiarize themselves with the underlying principles. A success factor for domain-specific architecture frameworks as the one presented in this paper is its capability to introduce and educate architects about the principles of the domain and how they are related to the approach supported by the architectural framework.

We conclude the evaluation with a discussion of some restrictions of the CIS-AF. Concerning applicability, the framework has been used for the design and development of new CIS on organizational level and for early-stage operations lifecycle. The degree to which the AF provides also supports other life-cycle stages that go beyond inception/early-stage have not been considered and it can be expected that the framework is not sufficient to completely address later stages. Concerning coverage, the framework focuses only on major CI-specific concerns, therefore additional VPs may be needed to address traditional concerns e.g. maintainability or availability in order to create a complete architecture description of a CIS. The evaluation highlighted a need for additional CI concerns that go beyond core elements and processes. Example concerns include keeping users motivated and engaged with the platform, privacy aspects of users and their personal data, qualitative and quantitative growth of the user-generated content, and CIS lifecycle and evolution.

VI. RELATED WORK

In this section we present related work on collective intelligence, architecture frameworks and stigmergy.

Collective Intelligence (CI) is a well-established phenomenon, that is currently under research in several fields like sociology, biology, political science and economics [2]. Lévy coined CI as *"A form of universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills"* [8]. In our work the focus is on

IT-mediated CI, which presumes a software-intensive system to enable CI capabilities. Lykourentzou et al. [9] categorize CIS into active/passive and collaborative, competitive and hybrid systems. In organization and corporate environments CIS are often found in context of Enterprise 2.0, which propagates the strategic integration of social web platforms within or between companies, their partners or customers [10].

Architecture Frameworks and Viewpoints are established approaches to document architectural knowledge and to support the creation of more effective architecture descriptions [1], [11]. The Networked Organizations Viewpoint for Architectures (NOVA) [12] addresses the scenario of connected online communities, but has limited coverage of technical realization and operation aspects compared to the CIS-AF. To our knowledge, there is a lack of architecture frameworks for CI and related domains, but there are some approaches in the field of agent-oriented software engineering like GAIA, INGENIAS and MaSE [13]. A limitation of these approaches is that they are methodologies to facilitate the design of multi-agent systems (MAS) in general, whereby the CIS-AF is an architecture framework focused to design a particular type of MAS.

Stigmergic Approaches have been studied extensively in the MAS community. One pioneering work is [14] in which the authors apply principles of stigmergy in manufacturing control systems. We took inspiration from the work on coordination artifacts of Omicini and colleagues [15] to define the notion of CI artifact in our framework. An interesting survey on human-human stigmergy was performed by Parunak [16]. This work shows how stigmergy works as a principle in human collaboration. Reference works on environment-mediated interaction for agents in general are [17], [18].

VII. CONCLUSION AND FUTURE WORK

In this paper we introduced an architecture framework to address CIS-specific processes and concerns like perpetuality, data aggregation and knowledge dissemination. The CIS-AF proved its capability to support the creation of architecture descriptions for CIS in two industry cases. Also, this work illustrates the benefit that can be gained by taking the step to create a new architecture framework for a particular system domain so that it allows a more nuanced perspective of the domain's needs and complexities. We are convinced that by this we not only provide software architects with a wider range of advanced frameworks to choose from, but also to give other computer science fields the mutually beneficial opportunity to share their knowledge with the software architecture community; like we build upon the work of multi-agent systems.

For future work, we plan to continue the support for architecting CIS by: (1) providing a process to effectively use the CIS-AF, (2) exploring possibilities for tool-support for the framework, and (3) enriching the CIS-AF with respect to CIS platform (co-)evolution and dynamics, e.g. actor engagement, trust, content curation. Further, we seek to investigate variations of existing CIS and how these variations are affected by underlying architectural elements and design decisions.

ACKNOWLEDGMENT

REFERENCES

[1] ISO/IEC/IEEE 42010. (2011) Systems and Software Engineering - Architecture Description.

[2] T. W. Malone and M. S. Bernstein, Eds., *Collective Intelligence Handbook*. MIT Press, 2015, in press.

[3] A. Omicini and P. Contucci, "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)*, ser. LNCS, vol. 8083. Springer Berlin Heidelberg, 2013, pp. 1–10.

[4] A. Omicini, A. Ricci, and M. Viroli, "Artifacts in the A&A Meta-model for Multi-agent Systems," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 3, pp. 432–456, Dec. 2008.

[5] F. Heylighen, "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, 2015, in press.

[6] J. Musil, A. Musil, and S. Biffl, "Towards a Coordination-Centric Architecture Metamodel for Social Web Applications," in *Proc. of the 8th European Conf. on Software Architecture (ECSA '14)*, ser. LNCS, vol. 8627. Springer International Publishing, 2014, pp. 106–113.

[7] J. Musil, A. Musil, D. Weyns, and S. Biffl. (2015) Supplementary Material: An Architecture Framework for Collective Intelligence Systems. [Online]. Available: http://qse.ifs.tuwien.ac.at/ci/material/pub/wicsa15/

[8] P. Lévy, *Collective Intelligence: Mankind's Emerging World in Cyberspace*. Perseus Books, 1997.

[9] I. Lykourentzou, D. J. Vergados, E. Kapetanios, and V. Loumos, "Collective Intelligence Systems: Classification and Modeling," *Journal of Emerging Technologies in Web Intelligence*, vol. 3, no. 3, pp. 217–226, Aug. 2011.

[10] A. McAfee, *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Harvard Business Press, 2009.

[11] D. Emery and R. Hilliard, "Every Architecture Description Needs a Framework: Expressing Architecture Frameworks Using ISO/IEC 42010," in *Joint Working IEEE/IFIP Conf. on Software Architecture & European Conf. on Software Architecture (WICSA/ECSA '09)*. IEEE, 2009, pp. 31–40.

[12] D. A. Tamburri, P. Lago, C. Dorn, and R. Hilliard, "Architecting in Networked Organizations," in *IEEE/IFIP Conf. on Software Architecture (WICSA '14)*. IEEE, 2014, pp. 247–250.

[13] O. Shehory and A. Sturm, Eds., *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks*. Springer Berlin Heidelberg, 2014.

[14] Hadeli, P. Valckenaers, M. Kollingbaum, and H. V. Brussel, "Multi-agent coordination and control using stigmergy," *Computers in Industry*, vol. 53, no. 1, pp. 75 – 96, 2004.

[15] A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini, "Coordination Artifacts: Environment-based Coordination for Intelligent Agents," in *3rd Int'l Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'04)*. IEEE Computer Society, 2004, pp. 286–293.

[16] H. Van Dyke Parunak, "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)*, ser. LNCS, vol. 3830, no. 2005. Springer Berlin Heidelberg, 2006, pp. 163–186.

[17] D. Weyns, A. Omicini, and J. Odell, "Environment as a first class abstraction in multiagent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 5–30, 2007.

[18] V. Parunak and D. Weyns, "Special issue on environment for multiagent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, 2007.