Introduction and Challenges of Environment Architectures for Collective Intelligence Systems

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

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

Abstract. Collective Intelligence Systems (CIS), such as wikis, social networks, and content-sharing platforms, are an integral part of today's collective knowledge creation and sharing processes. CIS are complex adaptive systems, which realize environment-mediated coordination, in particular with stigmergic mechanisms. The behavior of CIS is emergent, as high-level, system-wide behavior is influenced by low-level rules. These rules are encapsulated by the CIS infrastructure that comprises in its center an actor-created artifact network that stores the shared content. In this chapter, we provide an introduction to the CIS domain, CIS architectural principles and processes. Further, we reflect on the role of CIS as multi-agent system (MAS) environments and conclude with an outlook on research challenges for CIS architectures.

Keywords: Collective Intelligence, Coordination, Self-Organization, Software Architecture, Stigmergic Information System, Stigmergy

1 Introduction

Since the early 2000s, a new generation of web-based, social platforms has reshaped the way of knowledge creation and sharing. Well-known instances of such systems include social networking services (Facebook[1]), microblogging services (Twitter[1]), wikis and the online encyclopedia Wikipedia[1], content-sharing platforms (YouTube[1]), and review and rating platforms (Yelp[1]). These systems can be regarded as Collective Intelligence Systems (CIS), since these socio-technical platforms all have the capability to harness the collective intelligence of connected groups of people by providing a web-based environment for a community of participating users to share, distribute and retrieve topic-specific information in an efficient way. By contributing new content individually to these systems, their users build collectively a continuously growing repository of valuable information, knowledge and data and thus generate collective intelligence of a user community.

CIS are multi-agent systems (MAS), which operate on micro and macro levels and provide benefits both for their users and operators. The individual

[1] *http://www.{name}.com/* (all URLs last visited 06/18/2015)

user benefits from (1) the division of labor, since knowledge emerges from additive contributions of multiple users, and (2) efficient dissemination of knowledge among a large user group and leveraged awareness about activities and contributions of other users. For operators, CIS represent an approach to address complex knowledge-intensive problems on organizational, community and society level, which are improved in two ways. Firstly, hard to access knowledge is continuously aggregated from situated individuals on a global level, whereby *situatedness* of an individual means the *"physical, cultural, and social context, that guides, constrains and partially determines intelligent activities"* [28]. Secondly, the consolidated information is disseminated back to the individuals on a local level. The resulting feedback loop and quality of enabling the continuous adding, updating and restructuring of information gives CIS self-organizational capabilities that make them adaptable and resilient.

Therefore, CIS represent an interesting proving ground for the investigation of MAS-related concepts and theories. One concept that is central in this chapter is the environment [38]. In this chapter, we provide an integrated view of previous work by giving an introduction to the CIS domain as well as the architecting of CIS-specific environments, and conclude with an agenda for CIS architecture research. We argue, that self-organizational CIS are a particular family of MAS environments, which posses a characteristic system model [24]. The model consists of three layers which are a proactive actor base, a passive artifact network and a reactive/adaptive AMD (analysis, management and dissemination) system. Between these layers aggregation and dissemination dynamics exist that create a stigmergic feedback loop connecting the computational environment and the actor base [25]. This system model is the basis to derive an ISO/IEC/IEEE:42010 compliant software architecture framework [16], which should assist software architects to model CIS. So far software architects lack guidance in designing CIS that are tailored for specific application contexts, domains and for individual organizations. Thus, the framework provides consolidated systematic knowledge of the architectural principles and mechanisms that underlie each CIS. The CIS architecture framework (CIS-AF) consists of the three viewpoints CI Context, CI Technical Realization and CI Operation [25]. Each architectural viewpoint comes with its own stakeholders, concerns, model kinds and analytics. While working on this research, we discovered certain needs and limitations, which are described in the research agenda at the end of the chapter. The agenda deals with structure and dynamics of CIS, as well as future application domains. The work of this chapter builds upon advanced concepts of MAS, software architecture, and complex systems. For a deeper understanding of these concepts, we encourage the interested reader to explore the references [15], [23], [25], [34], [39].

The remainder of this chapter is structured as follows. Section 2 discusses related work on CIS and section 3 describes CIS characteristics and architectural principles which are illustrated with a real-world CIS platform. In section 4 we present an overview of the architecture framework for CIS comprising three viewpoints with their model kinds. An agenda for future research is discussed in section 5 outlining research challenges in the field of CIS architectures. Finally, section 6 concludes.

2 Related Work

This section presents an overview of related work on CIS foundations: coordination models, environment-mediated interaction and stigmergic coordination as well as IT-enabled collective intelligence.

2.1 Environment-Mediated Interaction & Stigmergic Coordination

Coordination is a key aspect of CIS. Central to the realization of coordination are coordination models, which were described by Gelernter et al. [11] as *"the glue that binds separate activities in an ensemble"* and by Omicini [27] as essential to define *"the abstractions and the computational models for ruling the interaction space in computational systems"*. Ciancarini [5] identified 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* enable communication among the entities, and serve as means for manipulations among the whole entity base. Examples of coordination media can be simple constructs like semaphores, monitors or complex constructs like tuple spaces [10] and blackboards [7]. Finally, *coordination laws* describe rules, constraints and mechanisms how entities are coordinated by means of the coordination media. The duality between coordination medium and laws was also described by Schmidt and Simone [32] in the context of Computer-Supported Cooperative Work (CSCW) using similar concepts of coordinative artifact and coordinative protocol.

In the last decade, particular focus has been drawn to the environment, which is created by the system, and its impact on the design of modern MAS [38]. Environment-mediated coordination approaches allow the decoupling of processes in space and time, and enable producers and consumers to stay anonymous [29]. A special form of environment-mediated coordination mechanisms is *stigmergy*, which was originally introduced by Grassé [12] to describe the spatial coordination among termite societies. Stigmergy enables not only environment-mediated coordination and indirect communication between agents, it possesses also a positive feedback mechanism [2, 3], so that an agent activity causes more activities. The mechanism promotes *awareness* among agents about the activities of other agents, which in turn reinforces their own activities [30]. Additionally, the process behavior of stigmergy is *emergent*, so certain system properties exist on a high-level, but not on a low-level and vice versa [1]. For stigmergy, this means that high-level, system-wide behavior is influenced by low-level rules, encapsulated by artifacts, the environment, and *local* activities. There is no explicit coordination control [8] and the agents are independent and choose autonomously which activities they perform [8, 9], [28]. In computer science, stigmergy has been explored extensively in various domains [40].

To better understand stigmergy in MAS, the concepts of the environment and artifact are of particular relevance [28], [30], [35]. Weyns et al. [38] defined the environment as *"a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources"*. According to the environment reference model [38], an environment's responsibility with respect to stigmergy is to act as a communication structure maintaining aforementioned dynamics. The *artifact* is used as a coordination medium and as an environment abstraction through which the agents communicate. Extensive discussions of coordination artifacts from a MAS perspective can be found in [8], [28] and from a CSCW perspective in [32]. Omicini et al. [28] provided 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 activities"* [28]. In addition, there are workspaces which represent local environments in which agents can interact with artifacts [28].

Susi et al. [35] provided a conclusive description of using stigmergy to support human cognitive processes and the usage of artifacts as mechanism to mediate emergent human collective behavior. Ricci et al. [30] adapted their 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 [30]. Parunak [37] surveyed stigmergic computational systems, which are used to coordinate human interactions. A comprehensive discussion of the current state of stigmergy and internet-supported collaboration was provided by Heylighen [15].

2.2 IT-Enabled Collective Intelligence

The phenomenon of *collective intelligence* (CI) has been investigated by researchers in a variety of disciplines like computer science, cognitive science, organization theory, biology and network science [20] and thus in literature a variety of CI definitions exists. According to Malone et al. [21], collective intelligence can be defined as *"groups of individuals doing things collectively that seem intelligent"*. The focus of this section is the discussion of computer science-related research of collective intelligence.

One of the first CI-related system concepts was introduced by Vannevar Bush in 1945 in his essay *As We May Think* [4]. In his work, he envisioned with the *Memex* a hypothetical system that had some of the features of modern CIS, in particular the concept of associative trails. Bush's work influenced other researchers like J.C.R. Licklider and Douglas C. Engelbart. Licklider argued in his work [18] for the needs of a tighter coupling of man and computing system, resulting in a hybrid, mutually-complementing overall system. It was then Engelbart's work on intelligence augmentation and in particular his seminal paper *Augmenting Human Intellect* [6], where he described the concept of a H-LAM/T system, which had two interesting aspects from a MAS environment perspective. Firstly, it highlighted the importance of artifacts, human-artifact interfaces and explicit-artifact processes. Secondly, he introduced the concept of an *executive superstructure*, which operates on a global system level so that *"more human time, energy and productive thought could be allocated to direct-contributive processes, which would be coordinated in a more sophisticated, flexible and efficient manner"* [6]. This executive superstructure can be regarded as an environment architecture and its occurrence, even in this early stage of research, supports the hypothesis that the environment concept is essential for the design of CIS. Interestingly, Engelbart already explicitly recognized the importance of computational automation in the system on a micro level (direct-contributive processes) and macro-level (executive processes).

Research efforts in IT-enabled collective intelligence have continued to gain momentum since the beginning of the Web 2.0 area and the rapid adoption of the first generation of CIS (wikis, social networking services, social media sharing) in a variety of domains and cultures. Besides Surowiecki's book *The Wisdom of Crowds* [34], there have been the works of Lévy [17] and Tapscott and Williams [36], which contributed to the wider adoption of the term collective intelligence. A repeatedly reported characteristic of CIS is the complementary interdependence between human and computing systems on a system level. In literature various terms refer to this attribute, which orbit around the same concept like socio-technical systems [27] or social machines [33]. Studies on the systematization of CI-related systems were conducted by Malone et al. [22], Lykourentzou et al. [19] and Smart et al. [33]. Grasso and Convertino [13] investigated tools and studies on CI in organizations, and Salminen [31] conducted a literature review on CI in humans. Gruber [14] examined how CI of the Social Web can be leveraged using knowledge representation and reasoning techniques from Semantic Web. A discussion of urban-level CIS and their challenges is provided by Zambonelli [39]. Two current collections on scientific CI literature are the book edited by Miorandi et al. [23] and the forthcoming book edited by Malone and Bernstein [20].

3 CIS Environments

This section describes major CIS characteristics and provides an overview of architectural principles as well as the underlying stigmergic process model. Finally, we illustrate the described architectural principles by applying them on the well-known example case of a Wiki.

3.1 CIS Characteristics

In the context of our research work a *collective intelligence system* is a *socio-technical multi-agent system* which mediates human interaction and provides support for distributed cognitive processes. As a socio-technical system, a CIS is driven by its users who contribute content (knowledge or information) to a globally-shared virtual information space located in a computational system, which in return feeds the consolidated information back to its users. This enables each user to benefit from novel and available information of high quality in her local space. Additionally, each user is stimulated to continue the contribution of further content into the globally-shared space. The feedback loop between the user base and the computational system is an essential feature of CIS, since it bridges the local and global space. Figure 1 shows a CIS process model consisting of 4 steps: (1) Actors (users) contribute/modify content of the shared computational platform. (2) The system analyses and processes content data and extracts consolidated information. (3) The system disseminates the information extracts among its actors. (4) Information stimulates either the actors' local activity or triggers a subsequent content contribution (revisit step 1).

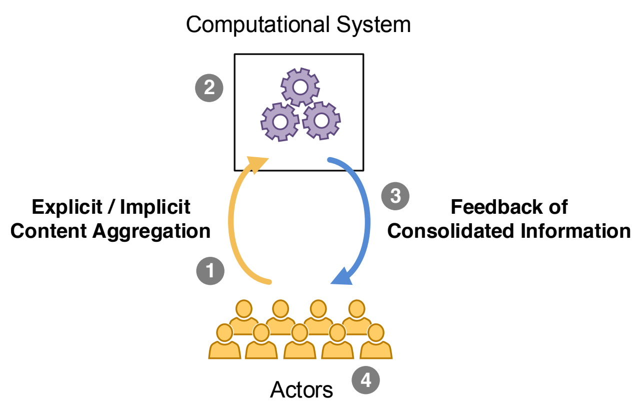

Fig. 1. CIS process with content aggregation and feedback of information

The created bottom-up feedback loop provides CIS with emergent, self-organizational capabilities and differentiates these systems from directed, top-down platforms used for crowdsourcing and human computation, where users are typically provided with task requests that await processing [20].

Another aspect of CIS is the conceptual restriction of the content in the information space to a certain *topic-of-interest*. It can be differentiated between two types of information stored in the space. *Topic-specific information* is data that is closely related to the information space's topic-of-interest, whereby *meta information* provides additional data about the topic-specific information as well as its creation and usage. There are three forms of how topic-specific information can be aggregated by a CIS:

– *Explicit content aggregation (ECA)* depends on the users to actively contribute content to the system. Example instances of such systems are Wikipedia, Facebook and YouTube.

– *Implicit content aggregation (ICA)* captures topic-relevant information as a side-product, while actors are performing a certain activity. A typical example of such a system is a web search engine.

– *Hybrid content aggregation (HCA)* accumulates some of the topic-specific information implicitly, but depends on users to actively contribute a remaining proportion of the data. An example of such a system is the navigation app Waze.

Further, we differentiate CIS by categorizing them according to their organizational structure within which they are typically used. We distinguish between the four levels of group, organization, community and society.

1. *Group level* CIS facilitate the collaboration within groups and teams. System examples comprise wiki systems (MediaWiki) and issue trackers (Redmine). Often systems, which are used on group level, are also applicable on organization level.

2. *Organization level* systems encompass an entire organization and can have an organization-internal or external focus. CIS, which are located on this level, are often associated with the terms Enterprise 2.0 and social collaboration. Representative system types include enterprise-level social networks (Yammer) and wiki farms (Confluence), employee suggestion systems, customer-feedback platforms (UserVoice) as well as a variety of custom-build CIS, which are tailored for a particular application context within the organization.

3. *Community level* CIS are dedicated to a particular aspect of a certain community which may be regional or a community of interest. CIS for regional communities include local review services (Yelp), but also platforms for emerging application domains like smart cities and collective governance. Illustrative examples of systems for communities of interest are TripAdvisor (travel), ResearchGate (social network for scientists), GitHub (code repositories), and MyExperiment (scientific workflows).

4. *Society level* CIS are systems that encompass one or more cultural regions and have developed a sphere of influence in or between these regions. Well-known examples are Wikipedia (encyclopedia), Facebook and VK (social network), Twitter and Sina Weibo (microblogging), as well as YouTube (video sharing).

The concepts that we introduced in the current and previous section provide the theoretical foundation for a systematic architecting approach for self-organizational CIS, which will be presented in the following sections.

3.2 CIS Architectural Principles

Our CIS approach builds on a multi-layer model (Fig. 2) consisting of three main components: (1) *human actors* as proactive components, (2) a *single, homogeneous CI artifact network* as a passive component, and (3) a *computational analysis, management and dissemination (AMD) system* as a reactive/adaptive component, which fosters information propagation among its actor base. In this configuration, the CIS architecture realizes a composite coordination mechanism facilitating *stigmergic, environment-mediated coordination*, whereby the coordination environment is formed by the artifact network and the AMD system [24]. The architecture enables the *bottom-up building of an artifact network* by allowing its actors to create/modify user-generated content stored in artifacts and thereby effectively accumulate and share information among each other [26]. This continuous flow of actor contributions within the system environment enables the *emergence of collective intelligence* that allows the individual to benefit for own purposes, and concurrently provides groups and organizations with self-organizational knowledge transfer and coordination capabilities. Further, a *perpetual feedback loop* is created between actor base (layer 1) and coordination infrastructure (layer 2 and 3), by instrumenting the actors' contributions to stimulate a subsequent reaction by other actors, causing a *stigmergic process* with aggregation (yellow arrow) and dissemination phase (blue arrow). The following paragraphs provide an overview of the CIS model layers.

1. Actor Base The actor base layer consists of *human actors*, who independently and actively perform *activities* on the *CI artifacts*.

2. Artifact Network The CI artifact network layer consists of passive *CI artifacts*, which store the topic-specific content that is generated by the actors. The conceptual content structure of the CI artifact is constrained by the system's topic-of-interest. CI artifacts are manipulated by *actor activities*, which resemble different types of create, read, update and delete operations. An important activity is the linking of artifacts using artifact links. *Artifact links* are links that actors can define between artifacts, leading to the emergent creation of an *artifact network* which is shared among the total actor base. Each performed activity is tracked in an *actor record*, whereby each actor has her own actor record. The actor record has two main purposes: Firstly, it logs the complete actor activities of each individual actor which allows the system to build knowledge about its actors and to provide advanced services like recommendations and shared interests. Secondly, the actor record acts as a proxy for the *ownership relationship* between the actor and the CI artifacts. The ownership relationship defines who is the owner of an artifact and thus who has extensive control to

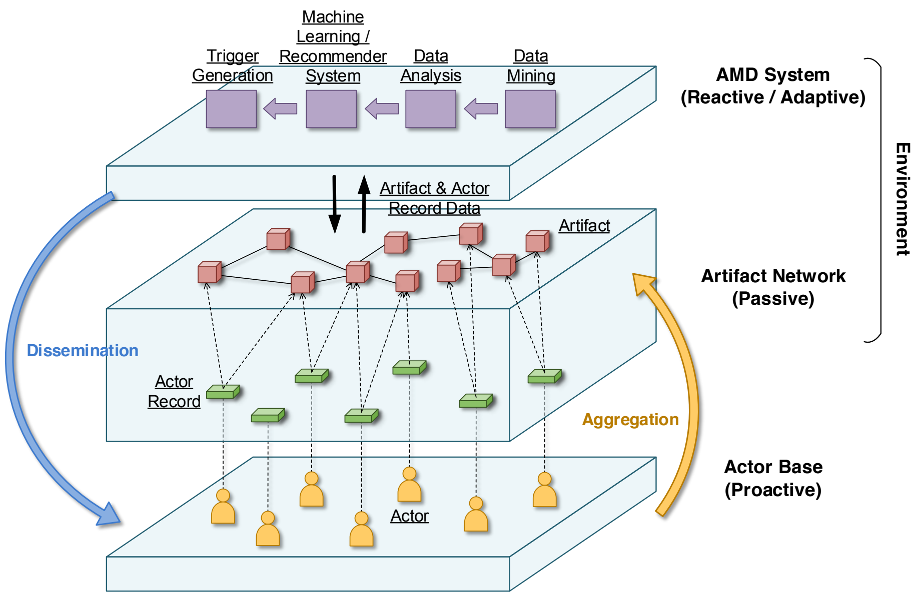

Fig. 2. Multi-layer CIS model with three main components and the stigmergic process

decide (1) to which extent other actors are able to contribute to the CI artifact, and (2) if contributions comply to predefined quality requirements.

3. AMD System The analysis, management and dissemination (AMD) system is a reactive/adaptive computational system that encompasses subsystems for *data mining*, *data analysis* and *machine learning* that are responsible for executing defined *rule sets*. In this process the subsystems use the aggregated artifact and actor record data and determined *dissemination mechanisms* to create various triggers. In detail, *triggers* are created to propagate changes of CI artifacts and to promote awareness about recent actor activities within the CIS among the total actor base. In addition, these triggers should also act as a stimulus to motivate each individual to react to these activities with a new contribution on an artifact, which in turn should attract other actors to contribute as well. For creating such triggers two different dissemination mechanisms can be applied. Pull-based, or passive, dissemination mechanisms rely on the actor 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 AMD system and its subsystems to forward updates and changes to the actors in order to make them revisit the platform. A common example is the sending of emails with personalized notifications and reports about artifact updates to actors.

3.3 Example: Wiki System

To illustrate the described architectural design principles of a CIS we map them to the well-known example case of a Wiki. In a Wiki-type CIS groups of people, known as editors, are interested in contributing and sharing knowledge about a certain topic. The actor base is formed by all users who have an active user account. Each actor primarily contributes new content to a Wiki either by creating a new article page or modifying an existing one, which represents the CI artifact. To improve the quality of a particular article, additional contribution activities are supported including adding of comments, starting discussions about an article's content using talk pages, and reviewing changed articles. Activities of each actor are tracked and stored in the actor record (AR)-like log, as is any modification of any article. Each contributed article modification creates a new revision of the article which improves the traceability of modifications by other actors and enables them to undo changes. Typically, all editors have equal ownership rights to all article pages in a Wiki which allows an editor the extensive manipulation of articles created by other editors. Articles can be linked together by actors using Wiki-links (internal links) and categories, creating a network of related articles which improves content discoverability. To improve awareness of artifact changes during an actor's absence the system uses internal and external (e.g. email) notification messages to deliver personalized information.

4 CIS Architecture Framework

To support software architects in the design of new CIS architectures, we developed an architecture framework for realizing CIS solutions (CIS-AF) following the ISO/IEC/IEEE 42010 standard [16]. In this section we present an overview of this CIS-AF which is discussed in detail in [25]. According to the standard, an architecture framework describes *"conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders"* [16]. An architecture framework typically addresses a set of concerns that stakeholders have with respect to the system-of-interest. These stakeholder concerns are framed by at least one architecture viewpoint. A viewpoint introduces conventions for constructing, interpreting and analyzing an architecture view which expresses the architecture of a system-of-interest from a specific perspective and addresses particular stakeholder concerns. Therefore, a viewpoint describes model kinds which specify modeling conventions used by architecture models that compose an architecture view.

In the context of the CIS-AF, the architecture framework aims to provide guidance for software architects to systematically describe key CIS elements and model a CIS that is well-suited for the context and goals of an organization. Therefore, the CIS-AF defines foundational principles of CIS, introduces key stakeholders and their concerns that need to be addressed in models and analysis, as well as provides architectural practices how to systematically design such CIS. Thereby, the focus of the framework is on CIS-specific concerns of the system realization from inception to operation and it consolidates architectural knowledge independent of a domain or technology. Hence, software architects may use additional architectural approaches to deal with other traditional stakeholder concerns, such as performance, availability or scalability.

The CIS-AF is based on our proposed meta-model for CIS [24] that defines key CIS elements which we described previously. The CIS-AF comprises three complementary architecture viewpoints together with their model kinds which define conventions for the construction and use of architecture views and models to deal with the identified essential CIS stakeholder concerns. An overview of the CIS-AF is illustrated in Fig. 3. The framework defines the following architecture viewpoints for realizing new CIS solutions: (1) *CI context viewpoint*, (2) *CI technical realization viewpoint*, and (3) *CI operation viewpoint*. Main stakeholder groups whose concerns are considered in the CIS-AF are architect(s) who design and describe the system architecture, owner(s) who define the CIS's purpose and business goals, manager(s) who are responsible for the management and operation of the provided services, builder(s) who develop the CIS, analyst(s) who are responsible for monitoring and assessment of the CIS performance and behavior, and actors who access and contribute to the CIS.

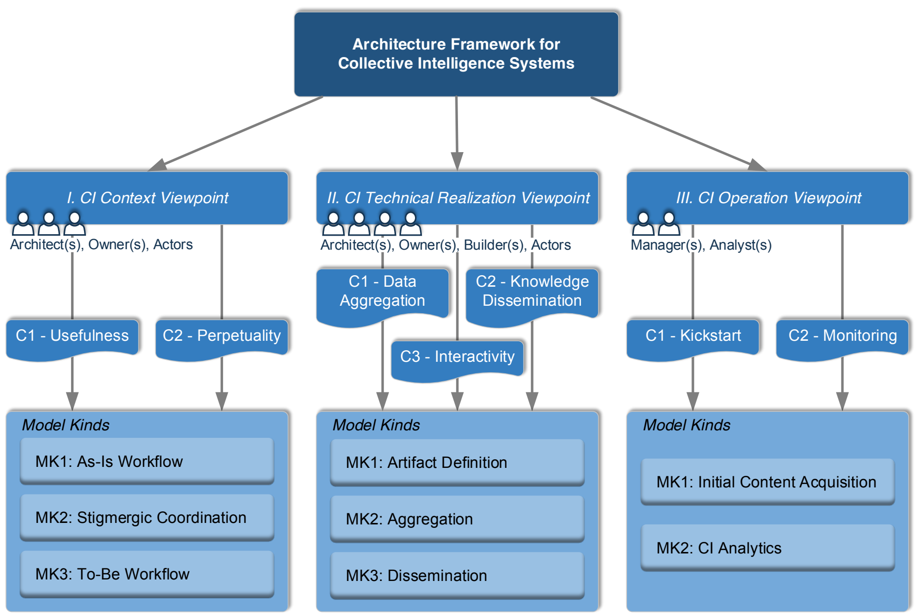

Fig. 3. Overview of architecture framework for Collective Intelligence Systems

CI Context Viewpoint

The context viewpoint deals with the design of CI-specific system capabilities especially with regards to the usefulness and perpetuality concerns of architects, owners and actors and describes the conventions to derive an architecture view which addresses these main stakeholder concerns. The viewpoint supports capturing relevant architectural design decisions to achieve the essential bottom-up information aggregation, management and distribution capabilities for hard-to-access dispersed knowledge and information. It defines three model kinds. The As-Is Workflow model kind governs models that show the current workflow of interest in the organization or context with the activities performed by users and an existing system environment that may be improved / extended by a CIS. A model created based on the Stigmergic Coordination model kind describes the domain items based on a particular topic-of-interest in the organization or context, the rules to interact with the domain items and for the dissemination mechanisms that use the network of domain items to generate stimuli in order to motivate the actor base. Finally, models based on the To-Be Workflow model kind express the future workflow of interest in the organization or context with the activities performed by users and the CIS environment, along with feedback from the CIS to the users.

CI Technical Realization Viewpoint

The technical realization viewpoint provides a more detailed perspective on the realization of the CIS and its specific capabilities and supports the concrete implementation of a new system with respective models. The viewpoint describes the conventions to derive an architecture view that frames the data aggregation, knowledge dissemination, and interactivity concerns of architects, owners, builders, and actors. It defines three model kinds. The Artifact Definition model kind governs models that describe the structure of the CI artifacts, how they can be linked, and which operations can be applied upon an artifact's content. A model created based on the Aggregation model kind shows details about how new data is aggregated from the actors, what activities can be performed by the actors to interact with the CI artifacts, what kind of data is aggregated, and to what extent these actor activities are captured. Models governed by the Dissemination model kind provides relevant information about the rules which realize the essential stigmergy-based dissemination of knowledge, the kind of content and ways how to effectively distribute this content in order to stimulate subsequent actor activities.

CI Operation Viewpoint

The operation viewpoint deals with the kickstart and monitoring concerns of system managers and analysts of CIS related to the successful startup of the perpetual feedback loop of a new CIS and its operation. Thus the viewpoint defines two model kinds to derive an architecture view that provides relevant information about initial data acquisition strategy and actor group as well as relevant indicators to measure CIS aggregation and dissemination performance. The Initial Content Acquisition model kind governs models that show potential sources from which initial content for the CI artifacts can be migrated and potential groups of initial actors to build up an actor community. A model created based on the CI Analytics model kind describes relevant metrics to measure the CIS performance and analysis results according to measurement profiles with probes to capture the data necessary for calculating the metrics.

First results of case studies, that we conducted to evaluate the framework's applicability and understandability among software architects, demonstrated that the framework effectively supports stakeholders with providing consolidated architectural knowledge in a documented and established form, a shared vocabulary of CIS concepts, and practical guidance to systematically apply the stigmergic principles of CIS. For a detailed description of the CIS-AF and the case studies results we refer the interested reader to [25].

5 Agenda for Future Research

Since the research of CIS architectures is at the beginning, we present in this section potential directions for future research. We discuss an agenda consisting of 11 research challenges across the areas of software architecture, technologies and system dynamics.

Nevertheless, CIS are complex systems and are dependent on areas that go beyond this research agenda. Figure 4 presents an extended overview of CI relevant areas with the four main areas of System, Influences, Agents and Stakeholders. A CIS is a hybrid system of agents and a computational system which consists of structure and dynamics. Its structure decomposes into the architecture and its conceptual design as well as its actual implementation using technologies. Its dynamics arise from the feedback mechanisms and the interplay of the agents and the system structure. Dynamics and the structure are dependent on each other. Additionally, a set of influences has an impact on the system's behavior and performance by enforcing various sets of constraints and rules. Influences are defined and negotiated between the stakeholders and, to varying extent, by the agent base. Main influences are ethics, culture, governance, and business, whereby each can be refined into more granular subject areas.

The following paragraphs focus on selected challenges from the areas architecture, technology and dynamics, that we expect to have high impact on future CIS design, development and research.

5.1 Architecture & Design

Architecture and Design challenges are concerned with conceptual and software architectural aspects of CIS and how they change over time and across application domains.

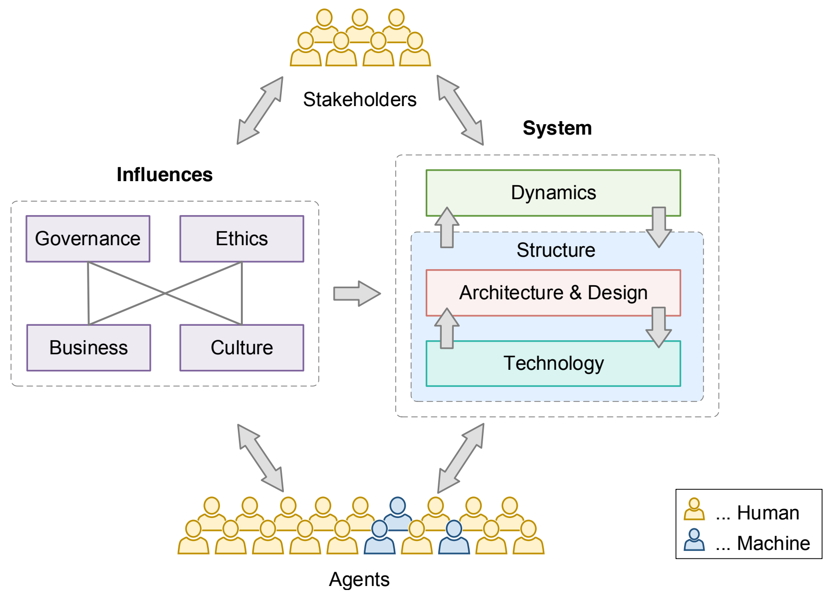

Fig. 4. CIS overview with System, Agent base, Influences and Stakeholders

1. *CIS MAS Architecture Models:* This chapter has illustrated that a coordination model like stigmergy can have a significant impact on the architectural structure of a CIS. Therefore, it would be interesting to explore if there are other nature-inspired coordination models [40] (i.e. chemical, biochemical, physical) that facilitate collective intelligence. How can these models be integrated into generalized architectural models, and what are strengths and limitations of these architectures? Also, how are these models translated into advanced software architectural models like architecture frameworks and reference architectures so that they are more applicable by practitioners?

2. *Platform Evolution Support:* CIS as service platforms tend to constantly advance over time to better address the needs of their actor base. This makes architectural evolution, erosion and architectural technical debt relevant issues that gain importance the longer the system is in service. Therefore, it is necessary to deepen the understanding of a CIS life-cycle, its different phases and their impact on the system architecture, as well as evolutionary transitions between life-cycle phases that support future growth paths. This is of particular relevance the larger a CIS's artifact network and actor base become, because then platform operators are more inclined to evolve the CIS into a more comprehensive form like a multi-sided platform or a software ecosystem. Besides the life-cycle, is there also a differentiation in maturity levels, which depend on the grade of a CIS's set of capabilities?

3. *Exploration of Architecture Variations:* A challenge is the current lack of architecture-relevant knowledge about commonalities and significant variabilities among key elements of CIS. Therefore, it is important to systematically investigate variations of existing CIS and how these variations are affected by underlying architectural elements and design decisions.

4. *Correlating Architectural Models and Dynamics Models:* A particular challenge represents the correlation of software architectural models with CIS dynamic models. A success in this area would allow new inter-disciplinary perspectives on the modeling of complex dynamic software systems. A promising future application scenario represents the *simulation* of CIS and their architectures which would enable to predict the effectiveness of system features before they are actually implemented.

5. *Beyond Human-to-Human Interactions:* CIS are typically understood as socio-technical systems, which mediate interaction between humans. A promising direction for future research would be to investigate the benefits of CIS environments with different types of actors, in particular human-machine and machine-machine configurations. This opens up new future application domains like *cloud robotics*, where robots can rely on humans as knowledge sources, or where robots can share task execution experiences among each other.

5.2 Technology

Technology challenges focus on how to support the implementation of CIS.

1. *CIS Middleware Frameworks:* The availability of advanced architectural models and frameworks enables the development of a new generation of CIS middleware frameworks that support the implementation of CI-intensive systems for particular application domains. The development of such frameworks will support the diversity of functionality and a wider range of technology stacks.

2. *Measurement and Analysis Components:* In order to support *data-driven development* practices in software development teams, it is important to explore the development of easy to use analysis and measurement components that provide architects and developers with CI-specific measures on system component and feature level. By this, developers are provided with a more accurate basis for making design and implementation decisions.

3. *Model-Driven CIS Engineering:* An efficient way to create new CIS implementations would be to apply methods from model-driven software engineering. In particular, how to adapt CIS architecture meta-models so that they can be the basis for model and code generation approaches? In combination with architecting tool support this would improve the utility and applicability of systematic CIS architecting and engineering among practitioners in industry.

5.3 System Dynamics

System Dynamics challenges are concerned with micro and macro level dynamics, the networks of CIS and effects on the system architecture.

1. *Network Models:* Since the artifact network is the central structure of a CIS, it is critical to also understand its characteristics also from a network science perspective. Findings from such investigations may inform CI-specific models on the dynamics *of* the network itself (changes of nodes and ties) as well as dynamics *on* the network (spreading processes like information cascades). Of particular interest is here the impact of the network on aggregation, dissemination as well as bottom-up and top-down feedback dynamics between the computational system and actor base.

2. *Growth and Perpetuality as First Class Concerns:* CIS are dependent on user-generated content and sustained high user activity levels. Therefore, it is important to understand factors that influence content growth and actor engagement like trust, content curation, and incentive mechanisms, and to document them in a way so they can support architectural decision making with regards to growth and perpetuality.

3. *Controllability:* The emergent, bottom-up nature of CIS is inherently non-deterministic and therefore only allows probabilistic estimates of the system's actual behavior [40]. Subsequently in order to improve CIS controllability, the underlying control principles, control points and their measures as well as observability and robustness aspects of CIS need to be better understood and validated. Advances in this area would not only provide the basis for novel mechanisms for the *dynamic adaptation* of CIS workflows and rules to achieve a certain system behavior, it would also extend the future applicability of CIS towards more critical domains.

We expect that research efforts in collective intelligence systems will continue to grow in the foreseeable future, making it a promising field of investigation. Therefore, finding solutions for the challenges described in this section will not only contribute to a better understanding of CIS and complex systems in general, it will also provide a benefit for the involved disciplines of software architecture, software engineering, multi-agent systems and network science alike.

6 Conclusion

This chapter provided an introduction to collective intelligence systems and how environment-oriented coordination mechanisms and abstractions can be used to describe them. The subsequent adaptation and integration of these concepts in an architecture framework enables software architects to adequately apply them for architecture descriptions of CIS. Additionally, the chapter presented research challenges that need to be addressed in future work for moving the field of CIS environment architectures forward. Advancing the presented models and approaches will not only increase our understanding on how CI-intensive systems work, it will also facilitate the exploration and invention of novel applications and usage scenarios.

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

References

1. Bedau, M.A., Humphreys, P.: Emergence: Contemporary Readings in Philosophy and Science. MIT Press, London (2008)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press (1999)
3. 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: Cheng, B., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525, pp. 48–70. Springer Berlin Heidelberg (2009)
4. Bush, V.: As We May Think. The Atlantic 176(1), 101–108 (1945)
5. Ciancarini, P.: Coordination Models and Languages as Software Integrators. ACM Computing Surveys 28(2), 300–302 (1996)
6. Engelbart, D.C.: Augmenting Human Intellect: A Conceptual Framework. Tech. rep., Stanford Research Institute, Menlo Park, CA (1962), http://www.dougengelbart.org/pubs/papers/scanned/Doug_Engelbart-AugmentingHumanIntellect.pdf
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 (1980)
8. Esparcia, S., Argente, E.: A Functional Taxonomy for Artifacts. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) Hybrid Artificial Intelligence Systems. LNCS, vol. 6077, pp. 159–167. Springer Berlin Heidelberg (2010)
9. Franklin, S.: Coordination without Communication. Tech. rep., Institute for Intelligent Systems, University of Memphis (1996), http://ccrg.cs.memphis.edu/~franklin/coord.html
10. Gelernter, D.: Generative communication in Linda. ACM Transactions on Programming Languages and Systems 7(1), 80–112 (1985)
11. Gelernter, D., Carriero, N.: Coordination Languages and Their Significance. Communications of the ACM 35(2), 96–107 (1992)
12. 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)
13. Grasso, A., Convertino, G.: Collective Intelligence in Organizations: Tools and Studies. Computer Supported Cooperative Work (CSCW) 21(4-5), 357–369 (2012)
14. Gruber, T.: Collective Knowledge Systems: Where the Social Web meets the Semantic Web. Semantic Web and Web 2.0 6(1), 4–13 (2008)

15. Heylighen, F.: Stigmergy as a Universal Coordination Mechanism: components, varieties and application. In: Lewis, T., Marsh, L. (eds.) Human Stigmergy: Theoretical Developments and New Applications. Springer (2015), http://pespmc1.vub.ac.be/papers/stigmergy-varieties.pdf
16. ISO/IEC/IEEE 42010: Systems and Software Engineering - Architecture Description (2011), http://www.iso-architecture.org/ieee-1471/index.html
17. Lévy, P.: Collective Intelligence: Mankind's Emerging World in Cyberspace. Perseus Books (1997)
18. Licklider, J.C.R.: Man-Computer Symbiosis. IRE Transactions on Human Factors in Electronics HFE-1(1), 4–11 (1960)
19. Lykourentzou, I., Vergados, D.J., Kapetanios, E., Loumos, V.: Collective Intelligence Systems: Classification and Modeling. Journal of Emerging Technologies in Web Intelligence 3(3), 217–226 (2011)
20. Malone, T.W., Bernstein, M.S. (eds.): Handbook of Collective Intelligence. MIT Press (2015), in press
21. Malone, T.W., Laubacher, R., Dellarocas, C.: Harnessing Crowds : Mapping the Genome of Collective Intelligence (2009), http://cci.mit.edu/publications/CCIwp2009-01.pdf, Working Paper No. 2009-001
22. Malone, T.W., Laubacher, R., Dellarocas, C.: The Collective Intelligence Genome. MIT Sloan Management Review 51(3), 21–31 (2010)
23. Miorandi, D., Maltese, V., Rovatsos, M., Nijholt, A., Stewart, J. (eds.): Social Collective Intelligence: Combining the Powers of Humans and Machines to Build a Smarter Society. Springer International Publishing (2014)
24. Musil, J., Musil, A., Biffl, S.: Towards a Coordination-Centric Architecture Metamodel for Social Web Applications. In: Avgeriou, P., Zdun, U. (eds.) Software Architecture. LNCS, vol. 8627, pp. 106–113. Springer International Publishing (2014)
25. Musil, J., Musil, A., Weyns, D., Biffl, S.: An Architecture Framework for Collective Intelligence Systems. In: Proc. of the 12th Working IEEE/IFIP Conference on Software Architecture (WICSA '15). pp. 21–30. IEEE Computer Society (2015)
26. 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)
27. Omicini, A., Contucci, P.: Complexity and Interaction: Blurring Borders between Physical, Computational, and Social Systems. In: Bdic, C., Nguyen, N., Brezovan, M. (eds.) Computational Collective Intelligence. Technologies and Applications. LNCS, vol. 8083, pp. 1–10. Springer Berlin Heidelberg (2013)
28. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A Meta-model for Multi-agent Systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
29. Papadopoulos, G.A., Arbab, F.: Coordination Models and Languages. Advances in Computers 46, 329–400 (1998)
30. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive Stigmergy: Towards a Framework Based on Agents and Artifacts. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) Environments for Multi-Agent Systems III. LNCS, vol. 4389, pp. 124–140. Springer Berlin Heidelberg (2007)
31. Salminen, J.: Collective Intelligence in Humans: A Literature Review. In: Proceedings of Collective Intelligence Conference 2012. eprint arXiv:1204.3401 (2012), http://arxiv.org/abs/1204.3401
32. Schmidt, K., Simone, C.: Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. Computer Supported Cooperative Work (CSCW) 5(2-3), 155–200 (1996)

33. Smart, P., Simperl, E., Shadbolt, N.: A Taxonomic Framework for Social Machines. In: Miorandi, D., Maltese, V., Rovatsos, M., Nijholt, A., Stewart, J. (eds.) Social Collective Intelligece, pp. 51–85. Springer International Publishing (2014)
34. Surowiecki, J.: The Wisdom of Crowds. Abacus (2005)
35. 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 (2001)
36. Tapscott, D., Williams, A.D.: Wikinomics: How Mass Collaboration Changes Everything. Portfolio (2006)
37. Van Dyke Parunak, H.: A Survey of Environments and Mechanisms for Human-Human Stigmergy. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) Environments for Multi-Agent Systems II. LNCS, vol. 3830, pp. 163–186. Springer Berlin Heidelberg (2006)
38. Weyns, D., Omicini, A., Odell, J.: Environment as a first class abstraction in multiagent systems. Autonomous Agents and Multi-Agent Systems 14(1), 5–30 (2007)
39. Zambonelli, F.: Engineering self-organizing urban superorganisms. Engineering Applications of Artificial Intelligence 41, 325–332 (2015)
40. Zambonelli, F., Omicini, A., Anzengruber, B., Castelli, G., De Angelis, F.L., Serugendo, G.D.M., Dobson, S., Fernandez-Marquez, J.L., Ferscha, A., Mamei, M., Mariani, S., Molesini, A., Montagna, S., Nieminen, J., Pianini, D., Risoldi, M., Rosi, A., Stevenson, G., Viroli, M., Ye, J.: Developing pervasive multi-agent systems with nature-inspired coordination. Pervasive and Mobile Computing 17, 236–252 (2015)