"Know thyselves" – Computational Self-Reflection in Collective Technical Systems

Jörg Hähner, Sebastian von Mammen,
Sabine Timpf, Sven Tomforde
University of Augsburg
{joerg.haehner|sebastian.von.mammen|sven.tomforde}
@informatik.uni-augsburg.de
sabine.timpf@geo.uni-augsburg.de

Bernhard Sick, Kurt Geihs, Thilo Goeble,
Gerrit Hornung, Gerd Stumme
University of Kassel
{bsick|geihs|thilo.goeble|gerrit.hornung}
@uni-kassel.de
stumme@cs.uni-kassel.de

Abstract—**The domain of self-adaptive and self-organising systems has faced noticeable attention within the last decade, since it investigates solutions to tackle complexity challenges arising from the increasingly coupled and dynamic character of emerging technical systems. The resulting solutions react to changing conditions and self-optimise their decisions over time. In this article, we outline that future intelligent technical systems will have to act as a collective, and this collective has to be equipped with novel techniques for decision strategies. Thereby, we have to go far beyond the existing reactive approaches in terms of proactive modelling of knowledge and goals, a continuous evaluation of goal achievement, and a dynamic goal adaptation process – to which we will refer to as "collective self-reflection". We provide a definition of this term, an architectural blueprint, and a draft of a research agenda towards collective self-reflection. We introduce an application scenario called "swarm fleet infrastructure" to motivate the need of such techniques.**

I. Motivation

The phrase "know thyself" goes back to the Greek ΓΝΩΘΙ ΣΑΥΤΟΝ which is said to have been inscribed at the wall of the Apollo's temple at Delphi in ancient times [1]. It hints at human self-conception that presumably distinguishes human cognitive capabilities from those of other species (and technical systems). More concretely, exclusively human abilities include concepts such as self-knowledge, self-awareness, and introspection. Clearly, these are abilities of individuals. By extending the phrase to consider collectives, i.e. "know *thyselves*", we highlight that some problems can hardly be solved with a limited individual scope, even if self-conceptional. Rather, they require a collective to establish and harness self-reflection in a distributed manner.

The term "computational self-reflection" has been coined to transfer the concepts underlying human self-reflection, i.e. self-knowledge, self-awareness and introspection, to technical systems [2]. This especially refers to the ability of technical systems to continuously monitor and improve their own behaviour in an uncertain, dynamic, and time-variant environment for situations that may not have been anticipated at design-time of these systems. This includes in particular (1) modelling the self, others, and the environment based on observations as well as (2) goal management, i.e. defining own goals, finding new ways to solve these goals, and to react accordingly. Especially in the context of systems that consist of a potentially large set of cooperating entities, e.g. smart grids consisting of a variety of different power producers [3] or a team of robots that have to cooperatively solve certain tasks [4], the notion of computational self-reflection has to be extended towards collective mechanisms. One example that we will refer throughout this article is expected to manifest within the next decades: A collective of autonomously driving and operating mostly electric vehicles. In this example, each vehicle is owned by different entities (i.e. individuals or companies) and may provide taxi services to various clients autonomously, only indirectly serving its owner. In addition, such a vehicle may also be self-responsible for recharging, for scheduling its repairs and maintenance, to minimise its costs of operation, and to maximise profits. Since the set of all vehicles within a certain environment, e.g. a city, have to compete for tasks such as taxi or delivery services, and since they might be organised in groups for efficiency reasons, decisions have to be made in response to the collective's status and demands.

In this article, we support the notion that collectives of intelligent technical systems need to harness means of self-reflection in order to maximise their utility. To this end, we propose the development of novel techniques that enhance the individual entities' awareness and their freedom of decision making. As we combine approaches of collective observation, modelling, and reasoning, we refer to the proposed concept as "collective self-reflection". We substantiate our concept with a definition of collective self-refection, an illustrative and easily transferable application scenario, a set of concrete, intertwined research questions, and the presentation of an architectural framework to integrate the targeted techniques.

The remainder of this article is organised as follows: Section II briefly summarises the state-of-the-art. Section III defines the term "collective self-reflection" and proposes an architectural framework. This is accompanied by an exemplary application scenario and two use cases in Section IV. Finally, Section V presents a research agenda how collective self-reflection can be achieved in intelligent, technical systems.

II. Related Work

The term *reflection* has its roots in social sciences [5], [6]. From a technical perspective, it is mainly associated with programming languages and their means of self-modifying programs. For instance, Maes already came up with a definition for *computational reflection* in 1987 by stating: "the activity performed by a computational system when doing computation about (and by that possibly affecting) its own computation." [7]. A specific example of this idea can be found in LISP where reflection is implemented using meta-programming. Such a meta-programme process is comparable to a standard programme by means of processing data. The difference lies in the ability to analyse and modify not just external data but also itself. Consequently, the self-modification of such a meta-programme is called (self-)reflective. Programming systems that support this type of reflection are called *procedurally reflective*.

As an alternative, programming languages may contain appropriate APIs to implement self-representation (e.g. in JAVA), which is typically known as *declarative reflection*. In this context, computational reflection (also called *behavioural reflection*) is concept used to alter the programming code of methods during runtime [8]. However, the main purpose of reflection in the programming context is not to write self-modifying algorithms but to circumvent restrictions of the programming language or to investigate the structures of objects at runtime, e.g. for checking the availability of certain methods or for debugging purposes.

Besides programming languages, research on reflective systems is also done in the domain of multi-agent systems. One example has been presented by Rehák et al. They suggest an abstract architecture to enhance multi-agent systems by means of reflective properties [9]. More precisely, they apply computational reflection as a technique to manage reflective processes, whereby changes previously applied to the system are used as knowledge-base to reflect on.

More recently, Bellman and Landauer transferred the concept of computational reflection to optimisation in complex systems [10]. In such a context, contradictory objectives and "mindlessness" (i.e. self-optimising processes that only on context models rather than on models of the self) define the need to find concepts beyond applying a static optimisation function. To model *reflective processes* so-called *wrappings* are used which comply to a more abstract representation of resources that allows to keep problems and their solutions separated [11]. This leads to a system that does not "call functions", "issue commands", or "send messages" but instead "poses problems". To solve a posed problem, the system "applies" resources (i.e., optimisation algorithms) in an automated manner.

From a legal perspective, issues of self-reflection have been a matter of research considered in the context of software-based agents in general [12], [13]. They comprise, for instance, the capability to conclude contracts [14], the development towards autonomy [15], or even towards a legal personality [16]. Two edited volumes address specific legal issues of, inter alia, liability, moral responsibility, technical design, and cyber-physical systems [17]. Hofmann/Hornung provide an overview of legal challenges of the connection of objects [18]. "Data-ownership" is being discussed in connection with automobile data by Hornung/Goeble [19].

III. Collective Self-Reflection in Intelligent Technical Systems

A. Computational Self-Reflection

In [2], the term "computational self-reflection" has been defined as follows:

Self-reflection in intelligent technical systems (or computational self-reflection) is the ability of the system to continuously monitor and improve its own behaviour in an uncertain, dynamic, and time-variant environment for situations that may not have been anticipated at design-time of the system.

This definition follows the assumption that self-reflection results from the interplay of three components:
1) Monitoring: Observing the environment, other systems (especially those the system is interacting with), and one's own behaviour,
2) Modelling: Building models for the oneself, the environment and other systems based on the observations as well as maintaining these models (including meta-knowledge such as experience gained by applying knowledge) during runtime, and
3) Goal management: Defining one's own goals, finding new ways to solve these goals, and to act and react accordingly.

The concept has some similarities with related approaches. For instance, the term "context awareness" has been coined to express the need for taking external information into account when deciding about necessary actions or adaptations [20]. Here, "context" is understood as "[...] any information that can be used to characterise the situation of an entity" [20], where an entity can be any relevant person, place or object that has impact. This context information is also important for self-reflective systems. Furthermore, the term "self-awareness" [21] is used to address the ability of a technical system to perform some kind of introspection [22]. Self-reflection goes beyond self-awareness since it explicitly takes goal management into account and consequently grants greater freedom of decisions to the system.

B. Collective Self-Reflection: Term Definition

We argue that tackling challenges in future intelligent systems will have to go beyond "self-reflection" as defined before. Instead of considering one individual, we need *collective* solutions that allow scaling the complexity of problems handled by technical systems, thereby potentially also increasing the complexity of the technical systems themselves. With the term "collective" we will refer to a group of entities that i) share a certain goal (or parts of it), ii) are motivated by the same needs, or iii) work together to achieve a common objective.

This definition originates from the domain of collective intelligence. Here, "collective intelligence" is defined as "a collective decision capability [that is] at least as good as or better than any single member of the group" by Hiltz and Turoff [23]. The members of the group can be considered technical systems or agents, although the original definition refers to human society. Smith narrowed the usage of the term down to "a group of human beings [performing] a task as if the group, itself, were a coherent, intelligent organism working one mind, rather than a collection of independent agents" [24]. A more general notion refers to "a form of universally distributed intelligence, constantly enhanced, coordinated in real-time, and resulting in the effective mobilisation of skills" [25].

Taking these origins and notions of "collective" into account and seizing the previously defined term of "computational self-reflection" (i.e. self-reflection in technical systems without a collective), we define collective self-reflection as the capability of a group of entities (or *software-based agents* to comply with the terminology of multi-agent systems [26] and Organic Computing [27]) as follows:

Collective Self-Reflection describes the capability of a group of entities or agents to jointly establish and harness a sense of self-reflection; this implies that the group of entities can build, communicate and make use of knowledge about the group's goals, its state, and its environment.

As a result, the aspects of computational self-reflection as considered in the case of individual entities find themselves in this new definition of collective self-reflection: The collective can introspect itself—each member may gather data about everyone else. Each member can build up knowledge about its peers, their situations, their states and build, adjust and share according models. Thus, the collective becomes aware of its state—as a group and in relationship to its context. Collective self-reflection does not claim to always yield complete and optimal tactics or strategies for the respective group of agents. Rather, only fragments of knowledge about a subset of the group might be available at a given point in time. Accordingly, the targeted techniques for realising collective self-reflection should work based on local communication and they should manifest themselves in self-organising, efficient, flexible and robust processes. Considering groups in a larger context, a collective might act as a single super-organism that assesses all external and internal conditions, that models and analyses perceived behaviours and decides about the most promising path to reaching its goals.

Our definition also implies the adjustment of the three components necessary for computational self-reflection of individuals: (1) monitoring, (2) modelling, and (3) goal management—without these components, building and harnessing knowledge about the group would not be possible. In addition, collective self-reflection requires the consideration of components concerning (4) communication and (5) organisation. In order to arrive at the desired collective capabilities, appropriate design concepts and techniques need to be fleshed out considering all five components/aspects. In the following section, we introduce a design concept that may serve as a blueprint to develop collective self-reflection.

The previous concept touches the domain of *Collective Adaptive Systems* (CAS) [28]. A CAS is a collection of heterogeneous autonomous entities that have autonomous goals and behaviours. Yet, these entities cooperate with each other and collectively adapt in order to accomplish their individual and common tasks and to reach their individual and common goals in efficient and effective ways [29]. CAS can be seen as the next step and logical continuation from context-awareness and self-adaptation of a single application or system. The main difference to collective self-reflection is that CAS focus on behaviour adaptation rather than goal management and distributed model maintenance. *Besides knowledge models, the novelty of the collective self-reflection approach lies in the system's ability to define new goals, alter existing goals, find new strategies to reach these goals, and finally, to make these processes subject to long-term self-improvement.*

C. An Architectural Blueprint for Collective Self-Reflection

Figure 1 illustrates the design concept for individual self-reflective systems. It picks up on the previous concept as introduced in [2] and builds on a generic Observer/Controller concept [30] known from the Organic Computing domain [27]. Therein, we distinguish between the *System under Observation and Control* (SuOC), sensors and actuators to perceive and modify the SuOC's conditions, and the self-reflective control mechanism (CM). This CM consists of four different layers operating on increasing levels of abstraction:

Layer 0: The **reaction layer** realises the standard functionality of the system. This includes observing the SuOC (performed by Observer 1 or O1), reacting to changes (performed by Controller 1 or C1), and modelling the knowledge (resulting in Knowledge base 1 or K1).

Layer 1: The **adaptation layer** enables the system to deal with new situations arising at runtime. This includes observing novel situations (O2), generating appropriate policies (C2), and modelling the knowledge base (K2). It is limited to relations and actions that can be explained with the existing models.

Layer 2: The **reflection layer** realises the concept of meta reasoning needed for self-reflection: O3, K3, and C3.

Layer 3: The **collaboration layer** can be triggered either by C2 or C3 and realises the communication with other, similar systems and, depending on the application, humans.

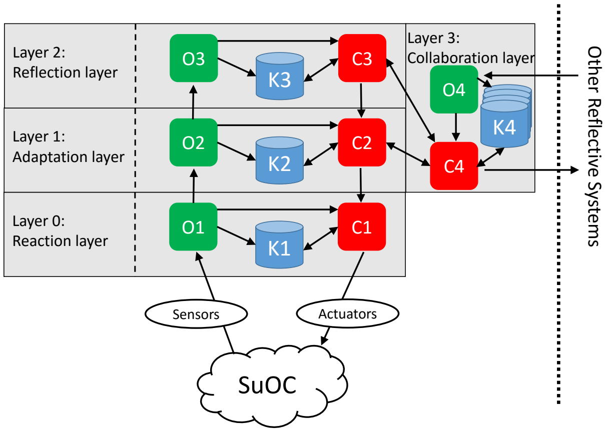

Fig. 1. Architectural blueprint for an individual self-reflective system [2].

In order to allow for collective self-reflection, we extend the concept in Figure 2. The extension focusses on layer 3, as it hosts the techniques for collective interactions. As in Figure 1, observer responsibilities are highlighted in green, while controller tasks are depicted in red. For simplicity reasons, we neglected details about the internal layers of each entity.

The decision making process driving the activities of the system consists of seven consecutive steps as marked in Figure 2: Initially (step 1), the participating entities gather raw data that describes the current conditions (including oneself, the peers, and the environment). The gathered data may be a result of all four layers of Figure 1. After its acquisition, it may be distributed among participating entities (step 2) and analysed. As a result, the underlying processes are modelled or existing models are updated according to the perceived information (step 3). Afterwards, the controller unit is triggered (step 4). Based on the available models, the controller decides about necessary actions to be taken by the collective (step 6). The actions are distributed to the participating entities (step 7), the internal knowledge bases (K1 to K4) are updated, or the currently active goal (or goal chain) is adapted.

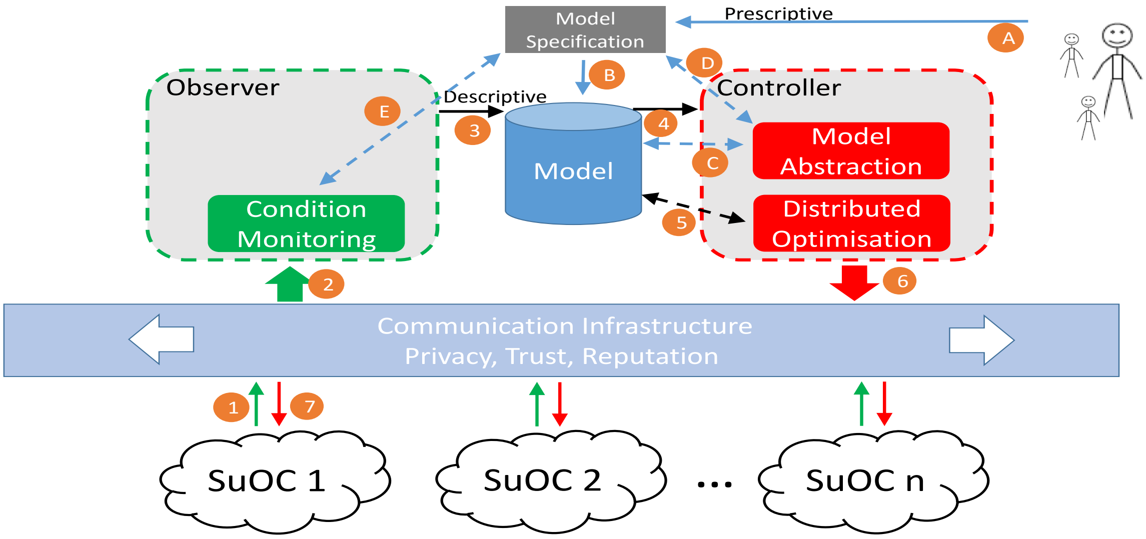

Fig. 2. Design Concept for Collective Self-Reflective Systems.

The modelling capabilities of the systems are also subject to modifications. Thereby, we distinguish between prescriptive (i.e. explicitly given from the outside, most importantly from the user) and descriptive (i.e. as a result of perceived observations) models. Initially (step A), user-provided models represent the basis for decision making. This necessitates a “common language”—for instance, a model of how observations relate to each other and which actions are available (step B). This refers, e.g., to the concept of ontologies. The model specification also defines which data is captured at which resolution. Based on the current conditions, the controller can alter the model setup (step C). In addition, it can also adapt the configuration of these observation models, i.e. varying the degree of abstraction (step D). This impacts the observation process itself, since the observer has to provide varying data and models (step E).

To illustrate the expected utilisation of the afore described concept, we introduce an application scenario for collective self-reflecting systems in the next section and explain how this architecture can be applied in this context.

IV. Application Scenario and Use Cases

As application scenario for investigating collective self-reflection capabilities, we propose a complex scenario that considers current trends in research and technology around individual mobility. Here, collective self-reflection aims at providing the required decision making capabilities to render traffic systems feasible that are composed of vastly heterogeneous units (from pedestrians to fully autonomous cars) and that need to respond with great dynamics (considering events as different as accidents, construction sites, evacuations or breakdowns in power supply). Due to the fact that collective self-refection could provide the foundation for seamlessly, adaptively and efficiently coordinating large numbers of traffic participants and shareholders, we refer to this application scenario as the “swarm fleet infrastructure” (SFI).

A. *Assumptions*

We assume that autonomously driving vehicles will have become standard technology in about ten to twenty years into the future. More precisely, we expect most of the vehicles in operation by 2035 to operate in autonomous mode (i.e. without direct user intervention), while a noticeable subset will be manually steered legacy machines. We further assume that ownership models and transportation demands will not change completely: Next to potentially publicly owned autonomous vehicles, cars will still be owned by private individuals and by companies. Lastly, we assume that the challenges inherent in establishing autonomous driving capabilities and making them available in the first place will have been solved by large companies, lobbyists, lawyers and politicians.

B. *Swarm Fleet Infrastructure*

Imagine a world where cars drive autonomously, decide on their own about fulfilment and prioritising of tasks (i.e. transporting people or goods from one location to another), and maintain their operational status themselves (i.e. control energy load and service status). From the current point of view, this might seem to be a utopian vision, but the way to making it reality is already paved[1]. In such a setting, novel technological concepts are needed that allow for an improved efficiency of resource utilisation, a decreased environmental impact of mobility, and an improved user-oriented behaviour [31]. We postulate that these aspects can be achieved more reliably, if we consider vehicles as complex, self-organising individuals. These individuals participate in a decentralised collective system that connects users, resources, and stakeholders – we will refer to this overall system as SFI.

On the one hand, the term describes the vision that the set of autonomous vehicles behaves as a collective (i.e. a swarm), while, on the other hand, they also serve themselves,

[1] See e.g. Elon Musk’s vision for his company Tesla at http://www.cbc.ca/news/business/self-driving-tesla-expected-within-3-years-elon-musk-says-1.3107475 (last access: July 15th, 2015).

the cities they dwell in, and their owners. The idea is to provide an organisational infrastructure that allows for an efficient and adaptive mastery of the resultant expectations. Different entities, pursuing individual goals in an according, decentralised infrastructure include:

Car: Heterogeneous vehicles types (i.e. manufacturer, size, capabilities) may participate. We model each entity as a self-motivated agent that has a set of desires (e.g. high healthiness status, reputation, low attrition) and beliefs (i.e. a model of the environment, others, and the self based on observations). A car belongs to an owner (i.e. private or commercial) and acts autonomously.

(Private) Owner: Private individuals or households can own autonomous vehicles that they share with the system during idle periods. The goal of participating in the SFI is many-faceted and includes earning money, provision of the car when needed, maximisation of the lifespan of car, automated maintenance, repairs, parking and charging, as well as keeping the car safe (avoidance of confrontations and malign customers).

(Commercial) Owner: The main difference to the private owner is that commercial owners will typically run fleets of cars that should not compete with each other – in contrast, they should collaborate to maximise the owner's profit. Ways to achieve this may include: Swarm formation for energy reduction, maximisation of the fleet's load, maximisation of customer satisfaction, autonomous adaptation of the fleet size, online information about fleet status, and meaningful statistics for planning purposes.

Client: Individuals who do not own a car may want to participate in SFI as clients. They need access to the infrastructure (e.g. through smart phones and credit card) in order to be served. Clients have varying needs including fast pick-up and delivery, cheap rates, customised rides (e.g. scenic route, smooth driving), transportation space, consent to smoke, etc. On top, they have multi-criterial priorities which kind of car should serve them and which route should be followed (i.e. reliable car, high reputation, safe routes) and they need precise information about waiting time, travel time, and costs.

City: Local authorities such as cities represent all their citizens and the roads. Their goal is the satisfaction and safety of their citizens. An array of subgoals can be inferred including minimal delays (i.e. no traffic jams), minimal pollution (i.e. exhaust, noise), minimised wear on infrastructure, guaranteed access for emergencies, avoidance of peaks in energy demand, and minimisation of re-structuring efforts.

Mechanics: Maintenance and repair of cars requires garage services. A garage has tools, employees, a specific location, and automotive supply. Its main goal is to earn money – which is supported by satisfied customers, minimised line-ups, minimised emergencies, minimised drop-in times, great numbers of long-term contracts, and a maximised load. In addition, the business schedule needs to be aligned with the number of available employees and their working hours.

Other Stakeholders: In addition to the aforementioned entities, other stakeholders such as advertisers, event locations, car manufacturers, car dealers, parking space providers or shop owners, also impact SFI's operation.

C. Use Case 1: Collective Group Behaviour

Imagine a group of commuters that work in the same office and mutually trust each other. These persons use their cars to go to work and on their way home. In order to improve the cost efficiency of owning a car, their cars autonomously serve as taxis while not being used by their owners. These cars cooperate in terms of forwarding tasks, negotiating intervals of maintenance and recharge, as well as deriving and maintaining the underlying world models. Important research questions arising in this use case are the following: a) How does a optimised strategy look like for a collective group of participating cars, especially if taking the openness of the system, the heterogeneity of the participants, and the uncertainty about the conditions into account? b) How should a strategy be generated and maintained collectively at runtime? c) For successfully solving collective tasks, e.g. serving a group of customers, timing constraints among the participating entities need to be considered. How can such dependencies be detected and taken into consideration at runtime?

D. Use Case 2: Cooperative Hazard Situation Prediction and Coordinated Maintenance and Repair

Accident-free driving is a key challenge for autonomous driving. Direct communication of cars might support this ambitious goal as it will allow for a collaborative situation-awareness. It will enable cars to predict critical driving conditions and allow them to either react autonomously or inform their drivers in a timely manner. However, the collected information will always be incomplete and uncertain. In addition, spatial and temporal effects have to be taken into account. As an example, imagine a cooperative prediction of aquaplaning hazards by means of information received from other cars [32].

Harzard prediction may immediately benefit drivers, clients and the cars themselves. Periodically scheduled maintenance sessions, on the other hand, and short-notice appointments for repairs are geared towards long-term profitability. To this end, different factors have to be taken into account, such as conflicting appointments at a garage, the time schedule of the owner, the commissioning of contracts to other cars, etc. As a result, numerous entities in the SFI need to coordinate themselves.

We assume that not all situations an autonomous car will ever face can be anticipated and addressed during its design. This assumption results in some important research questions: a) How can we enable a car to state when its own mechanisms for hazard situation prediction or maintenance and repair coordination perform poorly? b) How can the car improve these mechanisms (e.g., by adapting sensing and control parameters) in cooperation with other cars and, eventually, humans? c) How can a car assess these changes in order to guarantee that any modifications actually yield long-term improvements?

V. Research Agenda

Considering the SFI application scenario as outlined before, we developed a research agenda towards collective self-reflective systems. In particular, we propose the consideration of several interconnected aspects of reflection capabilities.

A. Descriptive Model Building, Self-Assessment, and Self-Improvement

The first part of the research agenda focuses on the observer and knowledge model parts of the reflection and collaboration layers with its capabilities for descriptive model building, self-assessment (cf. the concept of introspection), and long-term self-improvement. The most important research questions in this context are:

1) *Which knowledge and which knowledge models are necessary to accomplish certain goals and how can knowledge be modelled based on observations?* This question has to be answered with respect to the field of application. We have to model knowledge about the self, the SuOC, other entities of the overall system etc. As we have to consider the uncertainty of knowledge, probabilistic or possibilistic techniques could be taken into account. Another aspect is the way required information is gathered. It could, for example, be gathered in distributed way by actively collecting what is needed or by filtering the required knowledge from broadcasted messages.

2) *Which measures are required to assess knowledge models with regard to the current situation?* We need (a) the capability to compare knowledge about the self, the SuOC environment, other entities etc. to current observations in order to determine when expectations concerning current observations do not meet the actual observations anymore and (b) the ability to assess various aspects of (parts of) the current knowledge base with some objective and subjective measures (e.g., importance regarding a specific task or uncertainty regarding the observations or the parameterisation based on these observations).

3) *When and how do these models have to be adapted in order to achieve long-term improvements of the self and the overall system consisting of all entities?* Basically, this question has to be answered by the controller parts of the two layers, but the observer and knowledge model components have to provide an estimate of potential gain and risk of adaptation based on assessments of the current situation (we have to avoid either too fast or too slow reactions) as well as long-term self-inspection (monitoring).

B. Prescriptive Model Generation and Maintenance

In addition to the definition of the underlying descriptive model, the design and operation of collective self-reflective systems needs to deal with the prescriptive part of the model building. The prescriptive model represents the goals and intentions of the collective activities and thus is the foundation for collective decision making. In most cases, initially these goals are specified by the user. However, goals may be adapted during the runtime of the system when the overall situation changes. Important research questions in this context include:

1) *What logic formalisms, notations and techniques are appropriate to specify the goals of a collective system?* For example, is Answer Set Programming [33] a suitable approach here? Would it satisfy scalability and performance requirements?

2) *How and when will the prescriptive model be adapted?* We need the capability to monitor over time the degree of goal achievement and to collectively decide - in agreement with the user's intentions - that the goals should be changed. Clearly, there may be conflicts between the collective goals and the goals of individual agents. Such conflicts must be resolved by different kinds of negotiations or a priori defined priorities.

3) *What kind of reasoning and planning process is employed to generate efficient individual plans for the involved agents that they execute to achieve the defined goals?* Research in this realm also has to address the challenges of open environments where agents can leave and join the collective, e.g., due to communication link failures. Hence, we need the means to determine the membership in the collective, to (re-)distribute the current state of the knowledge including the current goals to joining agents, and all of this in a robust and consistent way.

C. Distributed Optimisation

The third major part of the research agenda is concerned with the distributed optimisation capability situated in the controller part:

1) *Given goals and sub-goals of a collective self-reflective system, what are methods for maintaining a required quality in fulfilling given system objectives?* This especially addresses the structure of the collective that aims at reaching a certain sub-goal or a goal with the best possible quality.

2) *Given an overall goal for a self-reflective entity, how can this goal be adapted such that it becomes suitable for the current state of the overall collective self-reflecting system?* Considering the dynamics a self-reflective system will face at runtime, the best way for reaching its objectives may change, goals have to be adapted in another way, or sub-goals have to be weighted differently.

3) *How can the success (e.g. gain, fairness, or cooperation) of the system be quantified in relation to success of individual elements?* Self-reflective entities within the collective self-reflecting system may have—to a certain degree—varying or even opposing goals. Therefore, approaches for balancing these different goals are needed.

D. Online Detection of Mutual Influences

Collective self-reflective systems consist of several autonomous entities. Collaborative fulfilment of tasks and cooperative information sharing can result in interdependencies among these entities (e.g. if entity A has to achieve its sub-goal prior to entity B starting its next task). In order to be able to incorporate such relations in the control strategies, they have to be identified in the first place:

1) *How can hidden mutual influences between entities be detected at runtime?* We need fast and efficient techniques since they have to be processed in parallel to the operative behaviour. In addition, influences have to be detected although they might not be explicitly recognisable (i.e. they are *hidden* or *indirect*).

2) *What kind of information sharing is necessary to detect transitive mutual influences?* In collectives of autonomous entities, mutual influences and dependencies affect more than just two entities. Consequently, we need concepts to identify groups of mutually influencing entities (or better: their behaviour and actions).

3) *How can cascading effects be prevented or at least be detected and signalled early?* Mutual influences can result in cascading effects that impact the overall system's performance. For instance, if entity A fails to achieve its sub-goal on which entity B relies to achieve its sub-goal (i.e. carrying a certain passenger), it fails as well. Consequently, we need mechanisms to avoid such cascades of failures at runtime.

E. Self-organised Model Abstraction

For collective self-reflection to work, the collaborating agents need to make their information accessible to each other. They need to fuse their collectively gathered data about the world and about themselves and update their distributed model base. In addition, they need to identify patterns in individually and collectively observed processes. Otherwise, they would only be able to adapt to changes in very limited (predefined) ways. Accordingly, in this fourth part of the research agenda, the following research questions arise:

1) *How can individual agents best recognise and harness patterns from their observations including communicated knowledge from other entities?* Pattern recognition is a costly procedure to begin with. To increase the challenge, the data exchanged among the agents in an open, heterogeneous system is generally multi-dimensional, possibly unstructured. Once identified, the agent can translate the pattern into facts and incorporate them into its model base. It might substitute more detailed, now obsolete, facts and thereby render the model base less costly to maintain and to communicate. More importantly, the added model facts will enrich the agent's decision making.

2) *How can the distributed model be increasingly simplified in order to serve a potentially ever-growing number of interwoven agents?* Self-organising collectives should ideally be open and new agents be allowed to join them. The growth of agents interacting and the concurrent growth in richness of accumulated data render it necessary to find ways to simplify the data that is communicated to large parts of the collective. The communicated messages need to condense fundamental insights in how to align the agents' activities, any details need to be avoided in order to minimise communication costs—and also the repercussions that maintaining detailed models would bring about. Building abstraction hierarchies on top of the accumulated data might hold the answer to this question: Disseminating the tip of the hierarchy would ensure communicating a minimal amount of data conveying a maximal amount of information.

F. Spatial Context Incorporation

In most cases, self-reflective behaviour takes place within a certain environment. Typically, such an environment is characterised by spatial attributes and relations. Consequently, we have to incorporate the spatial context within the collective self-reflective processes:

1) *When does the spatial situation affect a decision or a behaviour?* Each agent in the SFI is spatially and temporally situated. Broadcasting information ahead of one's direction of travel is different from communicating with agents in a specific spatial situation. Using spatial information to restrict communication requires spatial awareness of each agent. How is spatial context (other than location) represented and used for reasoning? How can it be used to restrict communication intelligently?

2) *How can spatio-temporal patterns be detected collectively?* How can we ensure that parts of patterns detected by distinct agents are collectively recognised as belonging to a distinct pattern? How can these patterns be characterised and modelled for communicating and recognition? Spatio-temporal patterns follow a specific rate of change, e.g. the starting point of a traffic jam moves ever backward when the jam is due to high density of vehicles but stays in place when due to an accident.

3) *How do changes in the environment affect collective self-assessment?* How do theses changes translate into changes in behaviour or decision making? If a single neighbouring agent changes its decision does this affect the collective decision making? What if it's the majority of neighbours? Where are the tipping points for decision changes in any given spatial situation? Which role does the perceived quality of a specific information and its location play in collective reasoning? How far away if far enough away for an agent to be unaffected? Can this qualitative distance be quantified in some fashion?

G. Human Self-reflection of Individuals and Communities

1) *Which influence do IT systems have on human self-reflection?* IT has fundamentally changed our way of communication and group interaction. An appropriate analysis and modelling of this influence on collective human opinion dynamics and self-reflection is required to design IT systems in a societally desirable way.

2) *How is collective self-reflection of IT systems related to self-reflection of human communities?* In order to support societal processes by collective self-reflective systems, we have to understand their relationship both on a structural and on a technical level. Which analogies do hold, which models can be adapted? Which human-computer interaction models are adequate? Which control mechanisms have to be established?

3) *How can IT improve collective awareness of a society for its environment and how can it trigger appropriate actions?* How can we take advantage of the synergies between both human and IT collective self-reflection to increase the collective of a society for its social and ecological environment, and how can we support human communities to take appropriate actions?

H. Legal Considerations

1) *Which legal rules does an autonomous system need to follow, particularly to prevent harm to users and third parties? Are there any essential rules that are a mandatory guideline for every action such a system takes, and for that reason have to be considered in the technical design?* It has to be carved out, whether the software has to be programmed to behave in certain ways in specific situations. Legal boundaries for an autonomous system have to be defined. Next, we need to analyse whether guidelines and boundaries deployed for individual entities scale well in case of interaction chains involving several entities, where attributability to single entities is not possible anymore.

2) *Who can be held responsible at the end of the day? Is there a need for an own liability insurance and legal estate if a system acts autonomously, or must there always be a natural or legal person held responsible? Could this person be absolved from liability?* A clearly defined relationship between an object and a person typically determines the degree to which a person is held responsible for any harm caused by the object. However, accountability is not only a matter of liability, but also important for the declaration of intent. *How must the relationship be designed legally, if the declaration of intent*

is autonomously produced by the system itself, instead of a legal person behind the system? Responsibility and liability are important concerns not only for the producer and owner of a self-reflecting system, but for the environment as well. Therefore, the current rules for contracts and torts have to be examined and, if they are not sufficient any more, new rules have to be defined.

3) *Which are sensible requirements for cooperation with other entities from a data protection view? How is it possible to determine the "controller" responsible for single processing operations, systems or data environments?* The legal standards have yet to be defined that allow to establish and work with the relationships between the various SFI entities and the environment. To this end, rules need to be defined to process the data in the system (store, modify, transfer, block and erase) and to specify the requirements by which it can be shared (transferred) with others. Transferring data is a pre-condition to make the systems really "connected". In this context the question of "data-ownership" arises, particularly with respect to non-personal data. Property rights are an absolute right. As a consequence, property rights might have to be redefined, if data should be a product like every other good.

A more visionary future perspective could consider the car as completely independent, not belonging to somebody as a legal object. *Would it be possible that an SFI entity is not considered a legal object anymore, but a (fully or limited) legal personality with its own rights and duties?*

References

[1] A. Scholtz, "Gnōthi sauton – "know thyself"," online: http://harvey.binghamton.edu/~grk101/gnothi_sauton.pdf, Binghamton University, Department of Classical and Near Eastern Studies, Binghamton, NY, 2006, (last access: 06/04/2014).

[2] S. Tomforde, J. Hähner, S. von Mammen, C. Gruhl, B. Sick, and K. Geihs, ""Know Thyself" – Computational Self-Reflection in Intelligent Technical Systems," in *8th IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014, London, UK, September 8-12, 2014*, 2014, pp. 150–159.

[3] G. Anders, A. Schiendorfer, J. Steghöfer, and W. Reif, "Robust Scheduling in a Self-Organizing Hierarchy of Autonomous Virtual Power Plants," in *ARCS 2014 - 27th International Conference on Architecture of Computing Systems, Workshop Proceedings, February 25-28, 2014, Luebeck, Germany*, 2014, pp. 1–8.

[4] J. Baxter, E. Burke, J. Garibaldi, and M. Norman, "Multi-Robot Search and Rescue: A Potential Field Based Approach," in *Autonomous Robots and Agents*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2007, vol. 76, pp. 9–16.

[5] D. A. Schön, *The Reflective Practitioner, How Professionals Think In Action*. New York, NY: Basic Books, 1983.

[6] G. Gibbs, "Learning by doing, a guide to teaching and learning methods," online: http://www2.glos.ac.uk/gdn/gibbs/index.htm, 1988, (reproduced by The Geography Discipline Network, last access: 06/20/2014).

[7] P. Maes, "Concepts and experiments in computational reflection," in *ACM Sigplan Notices*, vol. 22, no. 12, 1987, pp. 147 – 155.

[8] É. Tanter, "From Metaobject Protocols to Versatile Kernels for Aspect-Oriented Programming," Ph.D. dissertation, Univ. of Nantes, Nov. 2004.

[9] M. Rehák, M. Pěchouček, and M. Rollo, "An abstract architecture for computational reflection in multi-agent systems," in *Intelligent Agent Technology*, no. PR2416 IEEE, Los Alamitos, 2005.

[10] K. Bellman and C. Landauer, "Reflection Processes Help Integrate Simultaneous Self-Optimization Processes," in *ARCS 2014 – Feb. 25-28, 2014, Lübeck, Germany - Workshop Proc.*, 2014, pp. 1 – 5.

[11] C. Landauer, "Infrastructure for studying infrastructure," in *Presented as part of the 2013 Workshop on Embedded Self-Organizing Systems (last access: 07/14/2014)*. Berkeley, CA: USENIX, 2013. [Online]. Available: (https://www.usenix.org/conference/esos13/workshop-program/presentation/Landauer)

[12] C. Sorge, *Softwareagenten. Vertragsschluss, Vertragsstrafe, Reugeld.* Universitätsverlag Karlsruhe, 2006.

[13] R. Gitter, *Softwareagenten im elektronischen Geschäftsverkehr*, ser. Der elektronische Rechtsverkehr. Nomos Verlag, 2007, vol. 16.

[14] T. Allen and R. Widdison, "Can Computers Make Contracts?" *Harvard Journal of Law and Technology*, vol. 9, no. 1, pp. 25 – 52, 1996.

[15] S. Kirn and C.-D. Müller-Hengstenberg, "Intelligente (Software-)Agenten: Von der Automatisierung zur Autonomie? – Verselbstständigung technischer Systeme," *MMR - MultiMedia und Recht, C.H.Beck Verlag*, pp. 225–232, 2014.

[16] F. Andrade, P. Novais, J. Machado, and J. Neves, "Contracting agents: legal personality and representation," *Artificial Intelligence and Law*, vol. 15, no. 4, pp. 357–373, December 2007.

[17] E. Hilgendorf, *Robotik im Kontext von Recht und Moral*. Nomos Verlag, Baden-Baden, 2014.

[18] K. Hoffmann and G. Hornung, "Rechtliche Herausforderungen des Internets der Dinge," in *Internet der Dinge: Über smarte Objekte, intelligente Umgebungen und die technische Durchdringung der Welt*, F. Sprenger and C. Engemann, Eds. transcript Verlag, 2015.

[19] G. Hornung and T. Goeble, ""Data Ownership" im vernetzten Automobil: Die rechtliche Analyse des wirtschaftlichen Werts von Automobildaten und ihr Beitrag zum besseren Verständnis der Informationsordnung," *Computer und Recht*, p. 265ff., 2015.

[20] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Proc. of Handheld and Ubiquitous Computing, 1st Int. Symp.*, 1999, pp. 304–307.

[21] A. Gorbenko, V. Popov, and A. Sheka, "Robot Self-Awareness: Exploration of Internal States," *Applied Mathematical Sciences*, vol. 6, no. 14, pp. 675 – 688, 2012.

[22] M. Cox, "Field Review: Metacognition in Computation: A Selected Research Review," *Artif. Intell.*, vol. 169, no. 2, pp. 104–141, 2005.

[23] S. R. Hiltz and M. Turoff, *The Network Nation: Human Communication via Computer*. Addison-Wesley, 1978.

[24] J. Smith, *Collective Intelligence in Computer-Based Collaboration*. Erlbaum Verlag, 1994.

[25] P. Levy, *Collective Intelligence: Mankind's Emerging World in Cyberspace*. Basic Books, 1999, iSBN-13: 978-0738202617.

[26] M. J. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Hoboken, NJ, US: John Wiley & Sons Publishers, 2009.

[27] C. Müller-Schloer, "Organic Computing: On the Feasibility of Controlled Emergence," in *Proc. of CODES+ISSS'04*. ACM, 2004, pp. 2–5.

[28] F. Sestini, "Collective Awareness Platforms: Engines for Sustainability and Ethics," *IEEE Technology and Society Magazine*, pp. 54 – 62, 2012.

[29] S. Kernbach, T. Schmickl, and J. Timmis, "Collective adaptive systems: Challenges beyond evolvability," *ACM Computing Research Repository (CoRR)*, 2011, last access: 07/14/2014. [Online]. Available: ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/fet-proactive/shapefetip-cas09_en.pdf

[30] S. Tomforde, H. Prothmann, J. Branke, J. Hähner, M. Mnif, C. Müller-Schloer, U. Richter, and H. Schmeck, "Observation and Control of Organic Systems," in *Organic Computing – A Paradigm Shift for Complex Systems*. Birkhäuser Verlag, 2011, pp. 325 – 338.

[31] S. Tomforde, J. Hähner, H. Seebach, W. Reif, B. Sick, A. Wacker, and I. Scholtes, "Engineering and Mastering Interwoven Systems," in *Proc. of ARCS 2014 Workshops*, 2014, pp. 1–8.

[32] M. Reiss, B. Sick, and M. Strassberger, "Distributed Situation-Awareness in Vehicles by Means of Spatio-Temporal Information Fusion With Probabilistic Networks," in *Proc. of IEEE Works. on Adaptive and Learning Systems (SMCals'06)*, 2006, pp. 189–194.

[33] V. Lifschitz, "Answer set programming and plan generation," *Artificial Intelligence*, vol. 138, no. 12, pp. 39 – 54, 2002, knowledge Representation and Logic Programming.