



Coordination of Complex Socio-Technical Systems: Challenges and Opportunities

Stefano Mariani^(✉) 

Department of Sciences and Methods for Engineering,
Università degli Studi di Modena e Reggio Emilia, Reggio Emilia, Italy
stefano.mariani@unimore.it

Abstract. The issue of coordination in Socio-Technical Systems (STS) mostly stems from “humans-in-the-loop”: besides software-software we have software-human interactions to handle, too. Also, a number of peculiarities and related engineering challenges make a socio-technical gap easy to rise, in the form of a gap between what the computational platform provides, and what the users are expecting to have. In this paper we try to shed some light on the issue of engineering coordination mechanisms and policies in STS. Accordingly, we highlight the main challenges, the opportunities we have to deal with them, and a few selected approaches for specific STS application domains.

Keywords: Coordination · Socio-technical systems · MoK
Speaking objects · *ArgoRec* · Self-organisation · BIC · Argumentation

1 Introduction

Modern society is a growing interconnection of (sub)systems, such as healthcare, transportation networks, supply chain, etc. Its complexity thus stems from both the inherent complexity of each subsystem, and the added complexity of *interactions* amongst subsystems. The same can be said for modern IT systems, where managing interactions among components is at least as complex as designing the computational function of each component itself.

Furthermore, IT systems and the society rarely are isolated systems: rather, the latter relies on the former for many vital services and functionalities, giving birth to the so-called *Socio-technical Systems* (STS). There, “humans-in-the-loop” [20] are the norm rather than the exception, thus management of interactions further complicates: besides software-software, we now have software-human interactions to account for, too.

STS, in fact, arise when cognitive and social interaction is mediated by information technology rather than by the natural world (alone) [52]. As such, STS include people, processes, etc., which are inherent parts of the system. An example of STS is a Smart City, a social network, a Computer Supported Collaborative Work platform, any Internet of Things (IoT) deployment featuring human users—i.e. assisted living, smart homes, retail applications, etc.

STS are technically difficult to design, mostly because social activity is flexible and multi-faceted [1]. Also, a number of peculiarities and related engineering challenges have been highlighted by various research works [1, 12, 25]. Failing to recognise any one of the above facets, therefore missing to deal with the related issue, leads to a *socio-technical gap* in the STS, that is, a gap between what the computational platform provides, and what its users are expecting to have.

In this paper we try to shed some light on the issue of engineering coordination mechanisms and policies in STS. Accordingly, Sect. 2 highlights the main challenges to be faced in the process, Sect. 3 discusses promising approaches dealing with each of them, and Sect. 4 describes a few research contributions putting these approaches in practice in specific STS application domains.

2 Socio-Technical Systems: Challenges

Unsurprisingly, the challenges to be faced when engineering STS are twofold: technical, and socio-cognitive ones. Among the former, we can put any issue concerned with how to design the STS so that it serves at best its users' needs. Among the latter, we can include issues related to how users interact with the STS as well as to how they perceive it—in psychological terms. The remainder of this section somewhat follows this distinction, first by focussing on STS technical requirements such as self-organisation and adaptation, then by analysing the mindset humans have when dealing with STS, and finally by discussing how they perceive their computational (“algorithmic”) part. Please, notice that the latter two issues pose their own technical requirements, too.

2.1 Emergence, Adaptation, Awareness

In [1] the distinguishing properties of STS are discussed. Among the many, the following are particularly interesting in the context of this paper:

- STS have emergent properties, which therefore cannot be attributed to individual parts of the system, but rather stem from the dependencies between system components. Given this complexity, these properties can be evaluated only once the system has been tested and deployed, not at design time
- *awareness*, that is, knowing who is present, and peripheral awareness, that is, monitoring of others' activity, are fundamental in STS [21], because visibility of information flow – thus observability of dependencies – enables learning and improves efficiency [22]
- people *adapt* to the systems they use, but also strive to adapt those systems to best meet their needs [35, 37]

Emergent properties of a STS may be modelled and analysed, for instance by exploiting agent-based modelling and simulation frameworks [34]. Nevertheless, actual deployment of the STS inevitably has differences w.r.t. the “synthetic” version – i.e. unpredictability of human behaviour vs. predictability of software

agents – thus cannot exactly predict the actual STS dynamics. Supporting awareness may seem more feasible, but enabling observability of actions, interactions, and dependencies among activities – namely, anything that can happen in a STS – poses serious scalability and privacy issues which have no silver bullet available, yet—just look at the Cambridge Analytica scandal [40]. Adaptation received a lot of attention from computer scientists, there included the FOCLASA community—i.e. see [48]. What is most challenging here, unsurprisingly, is taking into account unpredictability of human behaviour: adaptations may solicit unexpected reactions in users, which may begin “fighting” against the system, possibly because they ignore the reason for adaptation and/or the expected benefits.

Dealing with each of the facets above is challenging on its own, let alone conceiving and designing a system successfully tackling all of them.

2.2 Goals and Assertions vs. Actions and Perceptions

In [24] further considerations about peculiarities of STS are made, in the specific context of a novel approach to engineering Social Internet of Things applications, that is, applications in which IoT devices and software must have social interactions to achieve the goals of the system, between both themselves and human users [3]. The proposed Speaking Objects approach puts emphasis on two traits of human interaction which sharply contrast with device-to-device interaction:

- humans better reason in terms of *goals* to be achieved, rather than by directly thinking at the actions needed to achieve them. Accordingly, they usually interact by expressing their own goals, not by explicitly commanding others what to do
- humans also better reason in terms of complex situations, state of affairs holding in the past, at present, or desired for the future, rather than by thinking at all the specific perceptions of their surroundings which make up a situation. Accordingly, they usually interact by exchanging *assertions* about those situations and their properties, not by debating over specific measurements of them

This is quite the opposite of what sensors and actuators do: they simply provide perceptions, which are then composed by someone else (usually, ad-hoc machine learning algorithms), and react to explicit commands for undertaking specific actions. Thus, their interaction is a mere exchange of measurements, sampling a specific facet of a complex situation, and commands about what to do (not what to achieve).

It is therefore quite difficult for humans and devices to fruitfully communicate and interact unless either humans learn “the language of devices”, or devices learn to think more like humans, in terms of goals and assertions—instead of actions and perceptions.

2.3 Algocracy and Trust

As clearly witnessed by statements such as the ACM “Statement on Algorithmic Transparency and Accountability” [2] we are living in an *algocracy*, that is, in a society in which increasingly pervasive, complex, and delicate aspects of our everyday lives are decided, or at least influenced, by computer algorithms [12]. A typical example is the filter bubble effect [38], caused, i.e., by the ranking algorithms running behind the news feed of social networks such as Facebook, or by more subtle ones, which are ultimately everywhere—for instance, algorithms regulate stock-trading, access to healthcare and insurance, employment chances, and much more [31,33].

While living in an algocracy does not necessarily represent an issue on its own, the lack of *transparency* and *accountability* does: if users of a STS have no clue of what is going on “behind the scenes”, and who is to blame when something goes wrong, they are likely to lose trust of the system—to eventually stop using it. The path toward making algorithms accountable and transparent is full of challenges and open issues, and often heavily depends on the specific scenario where the STS is deployed [14,18].

3 Coordination: Opportunities

As for Sect. 2, the approaches to coordination in STS may focus more on the technical side or on the socio-cognitive one. In the remainder of this section we discuss both: we start with the latter, by summarising the evolution from stigmergic to observation-based coordination, then proceed with the technical side by analysing how self-organisation can be engineered, and finally go back to the socio-cognitive facet of STS by briefly discussing how argumentation may alleviate the issue of trust and the fear of algocracy.

3.1 Observation-Based Coordination

Observation-based coordination captures the idea of coordinating an ensemble of agents by enabling them to observe each other’s actions, or the traces that those actions leave in the environment where they happen [41].

The most well known example of observation-based coordination is *stigmergy*. The term has been originally introduced to define the coordination approach of a specific species of termites in collectively building their nest, communicating and synchronising their activities through the environment rather than by directly communicating [19]. Then, throughout the years, it undergone many generalisations/specialisations/extensions [36,39,43]. Here, we refer to a generic set of coordination mechanisms mediated by the environment. Accordingly, stigmergic coordination requires that:

- agents act on the environment leaving some traces, or markers, which can then be *locally* perceived by others—and, possibly, affect their behaviour

- all interactions among agents are mediated by the environment, through traces—like ants’ pheromones
- emission of traces is generative, namely, once they are produced their lifecycle is independent of their producer’s one
- evolution of traces over time (and in space) may depend on the environment—as in the case of pheromone diffusion, aggregation, and evaporation in ant colonies

The interplay between these requirements produces *self-organisation*: whereas actions occur on a local basis – i.e. termites assemble soil locally, and ants sort broods locally – their effect is global in terms of the structures and behaviours they originate system-wide, by emergence [45]—as in the case of termites’ nests, or ants’ brood sorting.

Stigmergy is not confined to the world of insects: research in cognitive sciences emphasises the fundamental role that stigmergy plays also within human societies [44, 47]—hence in STS, too. There, stigmergy provides novel opportunities, because traces become amenable of human interpretation within a conventional system of signs, thus may be exploited by the cognitive abilities of the interacting agents to better coordinate. Along this line, [43] introduces the notion of *cognitive stigmergy* as the evolution of stigmergic coordination in those contexts where agents are capable of symbolic reasoning, as in the case of humans.

As such, cognitive stigmergy is a key enabler of *awareness*: through traces, in fact, agents may perceive what others are doing, and if traces are amenable of symbolic interpretation, they may even try to infer their intentions and goals. In turn, awareness is a pre-condition for *adaptation*: in order to plan actions aimed at improving the current situation, one must know what the current situation is, there including what others are doing. Then, emergent behaviours are likely to arise from the interplay between adaptation and awareness [6].

Both stigmergy and cognitive stigmergy are mostly concerned with traces of actions left in a shared environment, rather than on actions themselves. Also, they do not consider the effect that awareness of observability of actions and their traces have on the acting agent: if an agent knows that its actions could be observed, it may decide to act in a given way just to communicate something. This is where *Behavioural Implicit Communication* (BIC) [11] enters the picture, as a further generalisation of cognitive stigmergy embracing actions in their own right, too.

BIC is a cognitive theory of communication fostering the idea that practical behaviour can be used as a means for communicating, even without any additional specialised signal. On the contrary, communication actions are normally carried on by specialised behaviours (e.g., speech acts). BIC has been already taken as a reference for observation-based coordination, mostly based on a list of “tacit messages” that practical behaviours may convey. For instance, the “presence” tacit message communicates that “agent A is here”, by the fact that whichever is the action that A made, it is evident now for who observed it that A exists. Or tacit message “intention”, which communicates that “ A plans to do action β ”, by the fact that actions may (partially) reveal the plan behind

them, such as in the case where agents follow a pre-determined workflow therefore observing action α may reveal at which point of the workflow A is—thus the next action it has to commit to. The complete list of tacit messages and many illustrative examples can be found in [10].

BIC clearly represents a step forward on the path laid by stigmergy and cognitive stigmergy, enabling further forms of observation-based coordination based on practical behaviour rather than on dedicated communication acts, and on a process of signification of the intentions, conditions, and opportunities behind actions [9].

It is worth emphasising that existence of a suitable environment, where actions take place and their traces are recorded, is a necessary pre-condition for BIC, because it is the environment that enables and constrains observation of actions and their traces—in a computational world, in the physical one, and in the mixed world of STS. Also, it should be noted that BIC is a key enabler in raising the abstraction level within a STS, by allowing designers to think in terms of *intentions* and *goals* behind actions, and agents as well as interaction mechanisms to be designed accordingly.

3.2 Self-organisation

Given the importance of awareness and observability as witnessed by stigmergy and BIC, one could be tempted to adopt a centralised approach to coordination in STS, where a single component (the coordinator) has complete knowledge of the state of the system, and accordingly schedules others' actions globally so as to guarantee absence of unwanted interference and efficient collaboration. Nevertheless, *decentralisation* is one of the keys enabling self-organisation, which is a sort of holy grail for coordination models and languages: the ability of a system as a whole to *autonomously* (re-)configure itself in face of change, without any global supervision but rather relying on locally available information solely [17]. Besides self-* properties, decentralisation enables greater scalability, efficiency, and fault-tolerance w.r.t. centralised approaches, usually at the price of the complexity of implementation.

Awareness is apparently in contrast with decentralisation, since the latter explicitly avoids gathering of complete information by any component of the STS. Nevertheless, decentralisation often implies that computations depend on the context local to the executing component, that is, on what the component locally perceives about the state of the system, and on how it can act on its local portion of the computational environment to carry out its duties. Awareness is thus conveniently re-defined on a local basis in place of a global one, not lost. And there, *situatedness* plays a crucial role, as the property of actions (computation) and interactions (coordination) of being deeply intertwined with the environment they are immersed in [46]: on the one hand, they are affected by it, as it enables and constrains what agents can and cannot do, can and cannot perceive (thus, be aware of); on the other hand, they can affect it in turn, by changing its properties and structure (thus, possibly, also the admissible actions and perceptions).

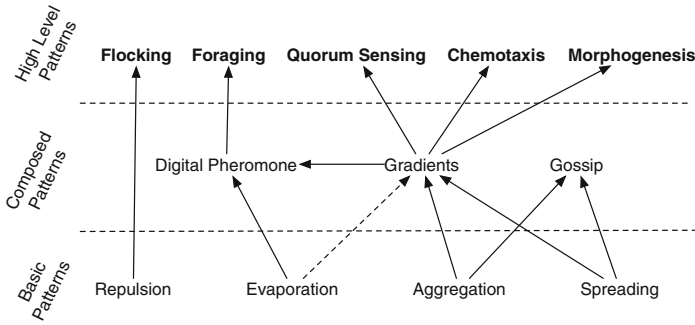


Fig. 1. Design patterns according to [16]. (Dashed) Arrows indicate (optional) composition.

Decentralised approaches to self-organisation have been widely studied, as regards both computation [4,32] and coordination [13,16], and already proved to be effective in dealing with the many issues of distributed computing in general [53]. In [16], for instance, the authors surveyed the literature regarding nature-inspired self-organising mechanisms, with the goal of compiling a catalogue of design patterns to promote reusability—pretty much like object-oriented design patterns do. As a result, the patterns depicted in Fig. 1 are detected, and related to each other in a compositional hierarchy consisting of three layers:

- Basic Patterns** – can be used to form more complex patterns, but cannot be further decomposed into smaller ones
- Composed Patterns** – obtainable as a composition of some basic mechanisms, and which in turn can serve as building blocks for higher level ones
- High Level Patterns** – patterns directly supporting complex self-organising emergent behaviours, showcasing how to exploit basic and composed patterns

For instance, gradients compose spreading and aggregation (optionally evaporation, too) to dynamically build routing paths inspired by force fields in physics [26], whereas flocking exploits repulsion as birds and schools of fishes do to maintain a given structure despite disruption.

All of the patterns may be implemented in a decentralised way, and leverage situatedness of interacting components to achieve self-organisation.

3.3 Argumentation

As just discussed, self-organisation is undoubtedly a desirable property to have for any given system, also because it alleviates the developers’ burden of explicitly intervene on the system to manually adapt its functionalities and behaviour upon need. Nevertheless, self-organisation comes at a price: for instance, it makes it more difficult to deterministically guarantee what the configuration or behaviour of the system would be in any given situation, due to the lack of

a global supervisor or strict design-time rules [20]. Although many approaches have been proposed to help engineers correctly design self-organisation, either through model-checking of desired safety and liveness properties [23], run-time verification of expected behaviour [7], or by construction [51], an important issue is still open: understanding what exactly went wrong in case of abnormal behaviour. The reason being that self-organisation is by definition a (partially) *opaque* process, where it is not always clear how to attribute a given global behaviour to the locally programmed rules—the so-called “local-to-global” issue [27].

Here, an intriguing opportunity is represented by *argumentation*, as the interdisciplinary research field studying dialogues and debates to understand their dynamics and the corresponding reasoning process [49]. In particular, computational argumentation exploits computational techniques to automatically analyse and build arguments and their relationships, there including generating explanations and justifications of decision making [8]. For instance, argumentation-based negotiation applies argumentation principles to negotiation-based coordination in multi-agent systems [42]. There, in fact, negotiation mechanisms are usually blind with respect to the strategy adopted by the agents participating in the protocol, that is, to their motivations in performing bids. By adding argumentation, instead, agents can disclose the reason why they are taking a given stance, thus improving the odds of reaching an agreement by collectively reasoning on conflicting goals and motivations. In the case of self-organisation, having components being able to explain why they performed a given action is a potentially effective way of promoting *accountability*, that is, exactly the practice of identifying someone or something as the cause of an effect.

Argumentation has therefore the great advantage of promoting trustability in a STS: if users can get justifications about the decision making undergoing “behind the scenes”, they are likely to increase their confidence in the capabilities of the system. In this respect, it is worth emphasising that striving to provide trustability and accountability is an increasingly hot topic well beyond coordination in STS, but in many fields of AI – from big data [33] to algorithms in general [31] – as witnessed by the recent “transparency initiative” endorsed by many organisations worldwide¹.

4 Selected Applications and Proposed Approaches

With the aim of showing how the research works described in Sect. 3 can be actually exploited in the real-world to tackle the challenges discussed in Sect. 2, the remainder of this section reports on three promising yet novel proposals each integrating some of the approaches in its own unique way: the *Molecules of Knowledge* model blends self-organisation with BIC, the Speaking Objects vision focusses on giving to users the right level of abstraction when interacting with technical systems while leveraging situatedness and argumentation, *ArgoRec* is

¹ <http://www.transparency-initiative.org/>.

mostly concerned with promoting trust and fighting algocracy through argumentation and awareness.

4.1 Self-organising Knowledge Management with *MoK*

The *M*olecules of *K*nowledge model [29] (*MoK*) fosters a novel way to engineer computational platforms supporting knowledge management in STS, according to which the software exploits users' interactions to continuously and spontaneously (self-)organise information. *MoK* is built around the integration of a biochemical metaphor [50] and BIC [11]: the former defines how to carry out computations, while the latter how to manage interactions.

Accordingly, a *MoK* system is a network of *compartments* (representing information repositories), where *seeds* (sources of information) continuously and spontaneously inject *atoms* (atomic information), which may then aggregate into *molecules* (composite information), diffuse to other compartments, gain/lose relevance, and so on. These processes are enacted by *MoK reactions* (the coordination laws dictating how the system evolves) executing within compartments, and influenced by *enzymes* (the reification of agents' actions) and *traces* (their side effects). Both enzymes and traces are left within compartments by *catalysts* (the agents) while performing their activities.

Reactions leverage decentralisation and situatedness to promote self-organisation: first, they rely only on information local to their compartment and can only affect neighbours, at most; second, they are scheduled according to dynamic rate expressions inspired by natural chemical reactions, which are sensitive to the contextual information (possibly) affecting their own outcomes. Enzymes and traces instead fully exploit the BIC theory for enabling awareness and observation-based coordination: by reifying actions themselves as well as their traces, in fact, they make agents aware of what others are doing, and thus enable their coordination. For a description of each reaction, enzyme, and trace, as well as their relationships, the interested reader is referred to [28].

In [30], a citizen journalism scenario is taken as a case study: there, users share a *MoK*-coordinated IT platform for retrieving, assembling, and publishing news stories. They use the *MoK* middleware for a number of actions such as searching for relevant information and working on these information to shape their own news stories. While they carry out their activities, users release enzymes and traces within their working space (a compartment), which ends up attracting similar information from other compartments through *MoK* reactions. Namely, the *MoK* middleware exploits users' (local) interactions to improve the (global) spatial organisation of information (Fig. 2): whenever users implicitly manifest interest in information, *MoK* interprets their intention of exploiting information, and the opportunity for others to exploit it as well, by attracting similar information toward the compartment where the action took place.

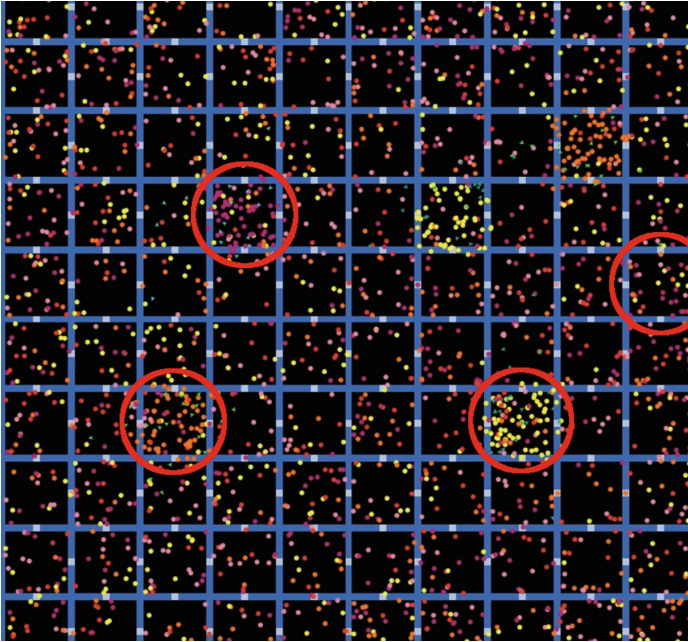


Fig. 2. Clusters of similar information appearing in *MoK* by emergence, as a result of users' interactions (image from [30]).

4.2 Traffic Control with Speaking Objects

In [25] the novel concept of Speaking Objects is presented as a brand new way to conceive and design distributed systems in general, with a particular emphasis on the Internet of Things vision. There, the core idea is that in a few years sensor and actuator devices will no longer simply provide measurements of pre-defined metrics and react to simple commands for affecting the state of the local environment. Rather, they will become able to assert complex situations about the state of the world and to *autonomously* pursue *goals* ascribed to users or explicitly designed for the system itself. Essentially, this amounts at transitioning from actions and perceptions to goals and assertions. Key enabler of such a paradigm shift is the increasing computational power that can be embedded in everyday objects, along with advancements in machine learning techniques, which, for instance, are making it possible to analyse data locally [5].

In such a setting, coordination becomes the capability of *argumenting* about the current “state of affairs”, and of triggering *conversations* to collectively decide how to act in order to achieve the desired future ones. Besides supporting decentralised coordination by leveraging opportunities for negotiation [53], argumentation also embraces humans-in-the-loop by enabling users to interact in natural language [8], and deals with the issues of trust and allocracy by mak-

ing explanations and justifications of decision making available and amenable of inspection and interpretation by human supervisors.

In [24], a traffic control scenario is taken as a case study. There, vehicles approaching an intersection are supposed to be equipped with an array of speaking and hearing objects, as the intersection itself—i.e., cameras, traffic lights, etc. As they get closer to the intersection, vehicles start arguing with the traffic light about who has the right of way and who should instead stop and wait (Fig. 3). After a negotiation phase where vehicles try to persuade the traffic light to decide in their favour, the dispute is settled when the argumentation process finds a solution for which no vehicle has to stop.

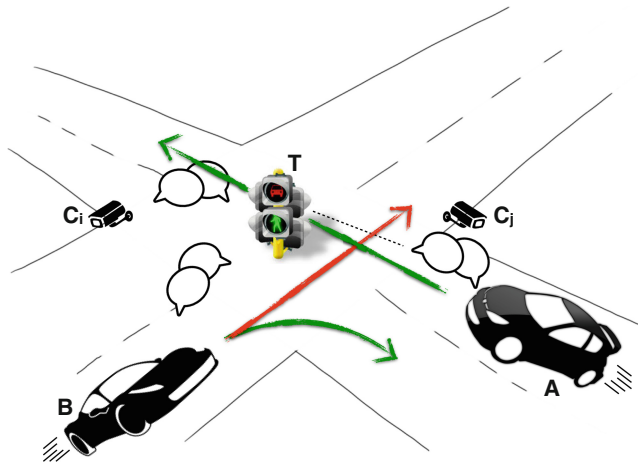


Fig. 3. Argumentation-based intersection management (image from [24]).

4.3 Personalised Medicine with *ArgoRec*

In [15] another example of fighting the fear of algocracy by promoting trust and interpretability with argumentation can be found. There, a recommender system named *ArgoRec* is introduced, whose distinctive feature is to rely on *argumentation* to provide *justifiable* and *personalised* recommendations. In fact, on the one hand argumentation empowers *ArgoRec* with explanatory power regarding why and how recommendations are provided, while, on the other hand, argumentation improves the user experience of patients thanks to natural language generation.

In particular, *ArgoRec* adopts a simple framework for structured argumentation—depicted in Fig. 4 as an argumentation graph: arguments (shaded boxes) are made up of claims (darker nodes) and their premises (lighter ones), and put together by attack (solid arrows) and support relations (dashed arrows). Attacks may be rebuttals (darker arrows) or undercuts (lighter ones),

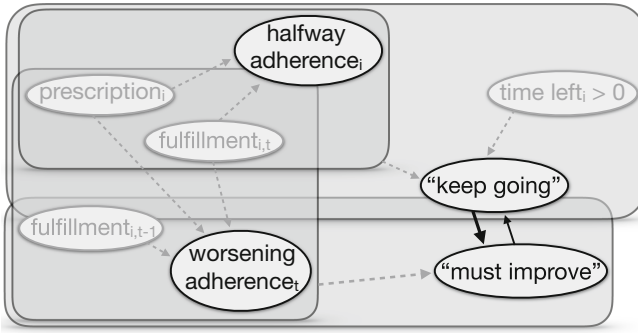


Fig. 4. Example of argumentation graph exploited by $\mathcal{A}rgo\mathcal{R}ec$ [15].

that is, contradictory claims or premises, respectively. All relations have a strength represented by a numeric weight (thickness of the line).

In $\mathcal{A}rgo\mathcal{R}ec$, recommendations are interpreted as arguments composed by a claim and its supporting premises: the former is, for instance, the suggestion to keep going with the current activities, while the latter could be the fact that the patient has a good compliance with the prescriptions assigned by the medical staff. Attack relations are the consequence of the fact that patients' compliance is monitored through automated tools – such as wearable devices – which provide raw measurements, which are unaware of the goal behind the prescription, nor of the history of the patient, nor of any other contextual information. The argumentation process enacted by $\mathcal{A}rgo\mathcal{R}ec$, instead, is aware of all the different measurements available, as well as of the status and history of the patient, thus exploits this awareness to both produce justifiable and personalised recommendations: the former by tracking the argumentation graph, the latter by generating sentences tailored to the patient (i.e. calling him/her by name, recalling the time left to complete a prescription, informing him/her on the current compliance, etc.).

For instance, in Fig. 4 recommendation “keep going” is the strongest argument: although comparison of latest event ($fulfillment_{i,t}$) with previous one ($fulfillment_{i,t-1}$) suggests to warn the patient (recommendation “must improve”) since her/his adherence to prescriptions is worsening, the fact that there is still time left to complete prescription ($prescription_i$) makes recommendation “keep going” win the dispute.

5 Conclusion and Outlook

Engineering socio-technical systems is a complex task, and this comes at no surprise: the technical perspective is usually quite different from the socio-cognitive one, thus adopting either standpoint easily leads to a socio-technical gap. Nevertheless, there exist approaches attempting to integrate these two facets, by carefully linking socio-cognitive theories with technical solutions, as in the case

of the three research proposals described in Sect. 4. There, for instance, *MoK* attempts to integrate decentralised computations with stigmergic coordination in the same framework, so as to achieve a sort of user-driven self-organisation—as happens with the clustering emergent phenomenon depicted in Fig. 2. Speaking Objects, instead, focuses on integrating goal-orientation with argumentation-based negotiation, so as to provide more flexible coordination schemes in distributed scenarios, while also increasing the abstraction level. All of this with the ultimate goal of shrinking the socio-technical gap arising when engineering STS as much as possible.

It is thus apparent that *integration* is the key here: as scientists and engineers, we need to find a way to include socio-cognitive aspects in our technical solutions since the very beginning of the design phase by using proper models and theories, not as an orthogonal dimension to be added later on, or dealt with in an ad-hoc way.

Accordingly, the three approaches discussed in Sect. 4 are not to be seen as mutually-exclusive solutions to the same problem, but rather as *complementary* one to each other as focussed on a different layer or perspective of the STS at hand. For instance, *MoK* is perfectly suited at working as the information handling layer in a Smart City deployment adopting the Speaking Objects vision. Just think of a Smart City as a large-scale STS: speaking and hearing objects are scattered throughout the city to compose the IT infrastructure with which human users constantly and seamlessly interact in their everyday activities. All the information recorded in this urban STS continuously and spontaneously evolves according to the *MoK* vision, and is made available to speaking and hearing objects as they need premises to either support or attack each other arguments. There, even *ArgoRec* could be added to the picture, for instance as an approach to build personal digital assistants offering guidance and assistance regarding. i.e., public transportation services, touristic attractions, interaction with the local administration, etc.

In this paper, we tried to shed some light on the possible paths to follow in order to make such an integration happen, with the aim of providing fertile ground for further discussion and research on the matter.

Acknowledgement. This work has been partially supported by the CONNECARE (Personalised Connected Care for Complex Chronic Patients) project (EU H2020-RIA, Contract No. 689802).

References

1. Ackerman, M.S.: The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Hum.-Comput. Interact.* **15**(2–3), 179–203 (2000)
2. ACM, US Public Policy Council: statement on algorithmic transparency and accountability, January 2017. https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf

3. Atzori, L., Iera, A., Morabito, G., Nitti, M.: The social internet of things (SIoT)-when social networks meet the internet of things: concept, architecture and network characterization. *Comput. Netw.* **56**(16), 3594–3608 (2012)
4. Babaoglu, O., et al.: Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst. (TAAS)* **1**(1), 26–66 (2006)
5. Bourzac, K.: Millimeter-scale computers: now with deep-learning neural networks on board, February 2017. <https://goo.gl/sciVTC>
6. Brun, Y., et al.: Engineering self-adaptive systems through feedback loops. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) *Software Engineering for Self-Adaptive Systems*. LNCS, vol. 5525, pp. 48–70. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02161-9_3
7. Calinescu, R., Ghezzi, C., Kwiatkowska, M., Mirandola, R.: Self-adaptive software needs quantitative verification at runtime. *Commun. ACM* **55**(9), 69–77 (2012)
8. Caminada, M.W., Kutlak, R., Oren, N., Vasconcelos, W.W.: Scrutable plan enactment via argumentation and natural language generation. In: *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS 2014*, pp. 1625–1626. International Foundation for Autonomous Agents and Multiagent Systems, Richland (2014)
9. Castelfranchi, C.: Modelling social action for AI agents. *Artif. Intell.* **103**(1–2), 157–182 (1998)
10. Castelfranchi, C., Pezzullo, G., Tummolini, L.: Behavioral implicit communication (BIC): communicating with smart environments via our practical behavior and its traces. *Int. J. Ambient. Comput. Intell.* **2**(1), 1–12 (2010)
11. Castelfranchi, C.: From conversation to interaction via behavioral communication: for a semiotic design of objects, environments, and behaviors. In: *Theories and Practice in Interaction Design*, pp. 157–79 (2006)
12. Danaher, J.: The threat of algocracy: reality, resistance and accommodation. *Philos. Technol.* **29**(3), 245–268 (2016)
13. De Wolf, T., Holvoet, T.: Design patterns for decentralised coordination in self-organising emergent systems. In: Brueckner, S.A., Hassas, S., Jelasity, M., Yamins, D. (eds.) *ESOA 2006*. LNCS (LNAI), vol. 4335, pp. 28–49. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-69868-5_3
14. Diakopoulos, N.: Accountability in algorithmic decision making. *Commun. ACM* **59**(2), 56–62 (2016)
15. Fernández, J.M., et al.: Towards argumentation-based recommendations for personalised patient empowerment. In: Elswiler, D., et al. (eds.) *Proceedings of the 2nd International Workshop on Health Recommender Systems co-located with the 11th International Conference on Recommender Systems (RecSys 2017)*. CEUR Workshop Proceedings, Como, Italy, 31 August 2017, vol. 1953, pp. 2–5. CEUR-WS.org (2017)
16. Fernandez-Marquez, J.L., Di Marzo Serugendo, G., Montagna, S., Viroli, M., Arcos, J.L.: Description and composition of bio-inspired design patterns: a complete overview. *Nat. Comput.* **12**(1), 43–67 (2013)
17. Di Marzo Serugendo, G., Karageorgos, A.: Self-organisation and emergence in MAS: an overview. *Informatica* **30**(1), 45–54 (2006)
18. Goodman, B., Flaxman, S.: European Union regulations on algorithmic decision-making and a “right to explanation”. ArXiv e-prints, June 2016
19. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles 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)

20. Hillston, J., Pitt, J., Wirsing, M., Zambonelli, F.: Collective adaptive systems: qualitative and quantitative modelling and analysis (Dagstuhl Seminar 14512). Dagstuhl Rep. **4**(12), 68–113 (2015)
21. Hudson, S.E., Smith, I.: Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. In: Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, CSCW 1996, pp. 248–257. ACM, New York (1996)
22. Hutchins, E.: Cognition in the Wild. MIT Press, Cambridge (1995)
23. Latella, D., Loretì, M., Massink, M.: On-the-fly PCTL fast mean-field approximated model-checking for self-organising coordination. Sci. Comput. Program. **110**, 23–50 (2015)
24. Lippi, M., Mamei, M., Mariani, S., Zambonelli, F.: An argumentation-based perspective over the social IoT. IEEE Internet Things J. **5**, 1 (2017)
25. Lippi, M., Mamei, M., Mariani, S., Zambonelli, F.: Coordinating distributed speaking objects. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 1949–1960, June 2017
26. Mamei, M., Zambonelli, F., Leonardi, L.: Co-fields: towards a unifying approach to the engineering of swarm intelligent systems. In: Petta, P., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2002. LNCS (LNAI), vol. 2577, pp. 68–81. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39173-8_6
27. Mariani, S.: On the “local-to-global” issue in self-organisation: chemical reactions with custom kinetic rates. In: 8th IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014, pp. 61–67. IEEE CS, London, September 2014
28. Mariani, S.: Coordination of Complex Sociotechnical Systems - Self-organisation of Knowledge in MoK. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-47109-9>
29. Mariani, S., Omicini, A.: Molecules of knowledge: self-organisation in knowledge-intensive environments. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) IDC 2012. SCI, vol. 446, pp. 17–22. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-32524-3_4
30. Mariani, S., Omicini, A.: Anticipatory coordination in socio-technical knowledge-intensive environments: behavioural implicit communication in *MoK*. In: Gavanelli, M., Lamma, E., Riguzzi, F. (eds.) AI*IA 2015. LNCS (LNAI), vol. 9336, pp. 102–115. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24309-2_8
31. Medsker, L.: Algorithmic Transparency and Accountability - AI Matters (2017) <https://sigai.acm.org/aimatters/blog/2017/06/01/algorithmic-transparency-and-accountability/>
32. Nagpal, R.: A catalog of biologically-inspired primitives for engineering self-organization. In: Di Marzo Serugendo, G., Karageorgos, A., Rana, O.F., Zambonelli, F. (eds.) ESOA 2003. LNCS (LNAI), vol. 2977, pp. 53–62. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24701-2_4
33. Nature: More accountability for big-data algorithms. Nature **537**(7621), 449–449 (2016)
34. Nikolai, C., Madey, G.: Tools of the trade: a survey of various agent based modeling platforms. J. Artif. Soc. Soc. Simul. **12**(2), 2 (2009)
35. O’Day, V.L., Bobrow, D.G., Shirley, M.: The social-technical design circle. In: Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, pp. 160–169. ACM (1996)

36. Omicini, A.: Agents writing on walls: cognitive stigmergy and beyond. In: Paglieri, F., Tummolini, L., Falcone, R., Miceli, M. (eds.) *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi, Tributes*, vol. 20, Chap. 29, pp. 543–556. College Publications, London, December 2012
37. Orlikowski, W.J.: The duality of technology: rethinking the concept of technology in organizations. *Organ. Sci.* **3**(3), 398–427 (1992)
38. Pariser, E.: *The Filter Bubble: What the Internet is Hiding From You*. Penguin, Harmondsworth (2011)
39. Dyke Parunak, H.: A survey of environments and mechanisms for human-human stigmergy. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) *E4MAS 2005. LNCS (LNAI)*, vol. 3830, pp. 163–186. Springer, Heidelberg (2006). https://doi.org/10.1007/11678809_10
40. Persily, N.: Can democracy survive the internet? *J. Democr.* **28**(2), 63–76 (2017)
41. Piunti, M., Castelfranchi, C., Falcone, R.: Anticipatory coordination through action observation and behavior adaptation. In: *Proceedings of AISB (2007)*
42. Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. *Knowl. Eng. Rev.* **18**(4), 343–375 (2003)
43. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive stigmergy: towards a framework based on agents and artifacts. In: Weyns, D., Parunak, H.V.D., Michel, F. (eds.) *E4MAS 2006. LNCS (LNAI)*, vol. 4389, pp. 124–140. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71103-2_7
44. Schmidt, K., Wagner, I.: Ordering systems: coordinative practices and artifacts in architectural design and planning. *Comput. Support. Coop. Work. (CSCW)* **13**(5–6), 349–408 (2004)
45. Di Marzo Serugendo, G., et al.: Self-organisation: paradigms and applications. In: Di Marzo Serugendo, G., Karageorgos, A., Rana, O.F., Zambonelli, F. (eds.) *ESOA 2003. LNCS (LNAI)*, vol. 2977, pp. 1–19. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24701-2_1
46. Suchman, L.A.: *Situated actions*. In: *Plans and Situated Actions: The Problem of Human-Machine Communication*, Chap. 4, pp. 49–67. Cambridge University Press, New York (1987)
47. Susi, T., Ziemke, T.: Social cognition, artefacts, and stigmergy: a comparative analysis of theoretical frameworks for the understanding of artefact-mediated collaborative activity. *Cogn. Syst. Res.* **2**(4), 273–290 (2001)
48. Usman Iftikhar, M., Weyns, D.: A case study on formal verification of self-adaptive behaviors in a decentralized system. *ArXiv e-prints*, August 2012
49. Van Eemeren, F.H., Grootendorst, R., Johnson, R.H., Plantin, C., Willard, C.A.: *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Routledge, London (2013)
50. Viroli, M., Casadei, M.: Biochemical tuple spaces for self-organising coordination. In: Field, J., Vasconcelos, V.T. (eds.) *COORDINATION 2009. LNCS*, vol. 5521, pp. 143–162. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02053-7_8
51. Viroli, M., Damiani, F.: Type-based self-stabilisation for computational fields. *Log. Methods Comput. Sci.* **11** (2015)
52. Whitworth, B.: Socio-technical systems. In: *Encyclopedia of Human Computer Interaction*, pp. 533–541 (2006)
53. Zambonelli, F., et al.: Developing pervasive multi-agent systems with nature-inspired coordination. *Pervasive Mob. Comput.* **17**, 236–252 (2015). Special Issue “10 years of Pervasive Computing” In Honor of Chatschik Bisdikian