Science of Computer Programming 184 (2019) 102317

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Coordination in Socio-technical Systems: Where are we now? Where do we go next?

Stefano Mariani

Department of Sciences and Methods for Engineering, Università degli Studi di Modena e Reggio Emilia, Reggio Emilia, Italy

ARTICLE INFO

Article history:
Received 12 February 2019
Received in revised form 1 August 2019
Accepted 23 September 2019
Available online 30 September 2019

Keywords:
Socio-technical gap
Coordination
Socio-technical systems
Social machines
Observation-based coordination

ABSTRACT

Despite Socio-Technical Systems (STS) have been defined and described long ago, and dedicated software engineering techniques and guidelines have been designed and assessed in different application domains, the issue of *coordinating* people and software artefacts in these STS has been widely recognised only recently, and gained momentum to generate some promising engineering approaches. In this paper, we aim to shed some light into the current status of the research landscape concerned with *engineering coordination* within STS. Accordingly, we highlight the main challenges yet to be tackled as stemming from real world problems, present the opportunities to deal with them as arising from different research threads, and delve into a few selected coordination approaches for specific STS application domains.

1. Introduction

Socio-technical Systems (STS) have been originally defined by Trist [76] as systems involving complex dependencies between humans, machines, and the *environment* within which they operate. According to Whitworth [80] STS arise when cognitive and social interaction is *mediated* by information technology rather than by the natural world (alone). Another definition is given by Chopra and Singh [25]: a STS is a system of principals (stakeholders such as people and organisations) whose *interaction* is supported by technical components for both computation and communication.

Regardless of what particular definition is adopted, it is already apparent the fact that modern society is a growing interconnection of STS spanning heterogeneous application domains such as e-health, intelligent transportation, information systems, Computer Supported Collaborative Work (CSCW), electronic markets, and more. More specifically, examples of STS either already existing or close to come are Electronic Medical Records infrastructures [59], which impact the work practice of medical staff considerably, open source software development platforms [16], where all sorts of communication tools are exploited to coordinate the efforts of geographically dispersed programmers, and (semi-)autonomous driving [37], for which overconfidence in technology was probably not an expected issue, but turned out to cause more (sometimes, fatal) harm than good [35,6].

Engineering such a sort of systems is difficult for a number of reasons analysed in many different contexts and from many different perspectives throughout the years [76,1,73,26,42]. Mostly, complexity stems from the radical difference between the two souls of STS: the *social* one, where cognitive, psychological, and organisational aspects of people work practices have to be taken into account to actually design supporting IT solutions, and the *technical* one, where performance, reliability, and correctness of software artefacts in enabling and supporting enhanced work practices is the main concern. For instance,

E-mail address: stefano.mariani@unimore.it.

https://doi.org/10.1016/j.scico.2019.102317

the technical side of STS should adapt to pursue goals ascribed by the social side, and such goals may usually be achieved by more than one means, hence the admissible technical adaptations are many and a design choice is needed.

Lately, another source of complexity has been clearly identified in STS: *interaction*. As people interact with other people through technology – as the definition of STS itself implies –, supporting and governing interactions plays a role of foremost importance in STS. Accordingly, *coordination*, as the discipline of ruling the interaction space to enable collective action, has been acknowledged as a necessary ingredient to tame STS complexity, thus successfully engineer them [54]. Nevertheless, engineering coordination in STS presents its own peculiar challenges, which can be attacked from many different research perspectives. For instance, social machines [36], human-agent collectives [39], e-Institutions and normative multi-agent systems (nMAS) [4,5] each offer its own set of abstractions, methods, and mechanisms to fill in the gap between the social and the technical dimension of STS.

Despite heterogeneity, we argue that these research efforts follow a precise direction while attempting to fill this gap, which influences their resulting benefits as well as their unintended shortcomings: *social-to-technical* efforts strive to reflect all the nuances of social activity and relationships in technical terms, so as to take advantage of additional information available to technical components while supporting people work practice and planning adaptations; *technical-to-social* approaches instead focus on exploiting the full potential of technical platforms to enhance and augment the effects of social interactions, with the goal of improving the level of support provided to users. For instance, the former approach includes coordination models based on commitments and social protocols [12,25], nMAS [71], and e-Institutions [53,3], whereas the latter is – to the best of our knowledge – less explored, and mostly features solutions falling under the umbrella of observation-based coordination [22], such as stigmergic approaches [61], behavioural implicit communication [77], or the most recent $\mathcal{M}$olecules $\mathcal{o}$f $\mathcal{K}$nowledge [47].

In this paper, we aim to shed some light into the current status of the research landscape concerned with engineering coordination within STS. In particular, our goal is to put in relation a few approaches selected from the literature so as to emphasise their *complementarity*. Accordingly, our contribution articulates along three main points:

- we select and describe a few significant challenges that research in STS coordination has yet to address, motivated by real-world problems, with the aim of devising out general requirements that the technical components of the STS should satisfy in order to be successfully designed
- we categorise coordination approaches in "social-to-technical" and "technical-to-social", as a means to (a) clearly identify which of the two souls of STS the approach mostly puts focus on and, accordingly, (b) its expected benefits and potential shortcomings
- we detect opportunities for *integration* by putting in relation different approaches seeking for some degree of complementarity

The remainder of the paper is hence structured as follows: Section 2 describes a few peculiar challenges of engineering coordination in STS, selected amongst the literature (a) as they stem from documented case studies and (b) for their specific relevance w.r.t. the coordination approaches presented later on; Section 3 discusses existing approaches to coordination in STS and draws attention to their similarities and differences so as to emphasise complementarity; Section 4 deepens analysis of a few specific proposals stemming from the aforementioned approaches and discusses potential integrations; finally, Section 5 provides final remarks and an outlook to future research efforts.

Disclaimer. This article is an extended version of paper *"Coordination of Complex Socio-technical Systems: Challenges and Opportunities"* firstly presented at the 16th International Workshop on Foundations of Coordination Languages and Self-adaptive Systems (FoCLaSa'18) and later appeared in Lecture Notes in Computer Science volume 11176 [45].

2. Coordination challenges in STS

In [59] deployment of Electronic Medical Records (EMR) is the subject of a study meant to assess the actual consequences on the doctors' work practice of introducing such an IT tool. This is a clear example of a STS where doctors, nurses, and students (the people) rely on the EMR platform (the technical component) for charting, taking notes, analysing lab results (computation) and exchange information, delegating tasks, etc. (communication). The study presents several (possibly) unexpected findings, which emphasise the unintended effects of the technical components on the social work practice. For instance: (a) decreased coordination with nurses, since doctors no longer completed paper charts at the nursing station, *while chatting with them*, but in a separate, dedicated EMR charting room; (b) increased delays in handing in completed charts, there including prescriptions to be delivered by nurses, since the EMR platform enables to save work and resume it later, *actually promoting a fragmentation in time of doctors' workflow*; (c) unchanged paperwork, since doctors developed the habit to carry around hand-written personal notes created at the patient room and then digitised in the EMR charting room to not forget information.

All these are enlightening examples of the *socio-technical gap* described in the introduction: in the specific case of [59], an undocumented but crucial interaction, doctors chatting with nurses while taking and dropping charting papers, was eliminated by the introduction of the EMR platform, resulting in decreased collaboration in the STS—which is explicitly in contrast with the intended purpose of IT solutions such as EMR.

Table 1
High level requirements for STS engineering.

Social desiderata	Requirement	Technical mechanism
coordination, efficiency	*(peripheral) awareness*	observability of actions, their side effects, their outcomes
flexibility	*adaptation*, *self-organisation*	observability, situatedness, pro-activeness
low cognitive overhead	low *abstraction gap*	goal-orientation, situation recognition, argumentation, institutional setting
trust	*accountability*	observability, transparency, argumentation, commitments, norms

Another insightful study is presented in [16]: there, free open source software development practice is analysed to uncover the extent to which the concept of *socio-technical congruence* is satisfied. Socio-technical congruence measures the degree of matching between the communication needs of an organisation and the coordination capacity of the supporting IT platform. As a byproduct of their study, finding evidence that coordination in absence of explicit communication is possible and beneficial, they (unintentionally?) emphasise that the many communication means provided by the technical components of the STS (such as email archives, chat, issue boards with comments, etc.) are rarely used, in favour of direct action on the source code in the form of commits, companion commit messages, and corresponding comments.

Yet again, this exemplifies a gap in the STS, as a situation in which *the technical layer of the STS has unforeseen effects on the social layer*: the functionality of highlighting differences w.r.t. previous versions for each commit, and the possibility of complementing the commit with a message, was apparently all the coordination capacity that developers were asking for, regardless of the communication needs envisioned beforehand—mail and chat, for instance, were rarely used despite active and prolonged collaborations.

Finally, the recent rollout of self-driving/autonomous cars capable of semi-autonomous driving[1] revealed an issue apparently overlooked by researchers and practitioners working in the field: *overconfidence (or, excessive trust) in technology*. Apparently, in fact, a few (fatal) car crashes involving semi-autonomous driving have been caused by over-reliance on the self-driving ability of the vehicle [30], as in the case of a recent Tesla accident [35,6]. There, as well as in other cases among those reported to California Department of Motor Vehicles, the driver seemingly ignored multiple warning notifications requesting human attention and/or intervention, as revealed by post-hoc analysis of the algorithmic decision making process carried out by the onboard software.

Unsurprisingly, the real-world scenarios described above emphasise that the challenges to be faced when engineering STS are twofold: *technical*, and *socio-cognitive* ones. The remainder of this section roughly follows this distinction, by focussing (i) first on a few specific technical requirements for enabling and supporting fruitful coordination in STS, such as the importance of promoting *awareness* and the need for loose specifications to allow for *adaptations*, then shifting to the socio-cognitive side by (ii) discussing the *abstraction gap* between the mindset humans have when reasoning over situations for the accomplishment of a goal as opposed to how machines “think”, and finally (iii) how humans perceive and relate to the technical side of STS and what the latter can do to improve trust, hence adoption. Notice, the latter two issues bring along their own technical challenges, too, and overcoming each of them means to actually satisfy the *requirements* posed by the STS to be engineered—summarised in Table 1.

2.1. Awareness, adaptation, emergence

In [1] the distinguishing properties of STS are discussed. Among the many, the following are particularly interesting as they have been recognised independently and in other research threads, too—see for instance Badham et al. [9], Noriega et al. [54], Jennings et al. [39]:

- *awareness* (knowing who is present) and *peripheral awareness* (monitoring of others’ activity) are fundamental in STS because visibility of information flow, thus observability of dependencies amongst activities, enables learning and improves efficiency [38] as well as supports and promotes coordination [54,39]
- the technical components of STS should *adapt* to and pursue goals which are ascribed to them by an external *environment* [9]. In addition, people adapt to the systems they use while also striving to adapt those systems to best meet their needs [57]—in a sort of joint optimisation loop
- 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, often times these properties can be fully evaluated only once the system has been tested and deployed, not at design time

Emergent properties of STS may indeed be technically modelled and analysed, for instance by exploiting agent-based modelling and simulation frameworks [52]. Nevertheless, actual deployment of the STS inevitably has differences w.r.t. the “synthetic” version, for instance due to *unpredictability* of human behaviour w.r.t. predictability of software agents, thus cannot exactly emulate and assess the actual STS dynamics. In the aforementioned EMR case study, for instance, the doc-

[1] These terms are often used interchangeably, but the most accurate is “semi-autonomous” as no fully self-driving car has the right to hit the streets, as of today.

tors' workaround of carrying personal handwritten notes as a memory aid was an emergent phenomena observed after deployment, during actual system operation, that was not anticipated by designers of the STS, thus impossible to simulate.

Supporting awareness may seem less of a problem, but enabling *observability* of participants, their actions, interactions, and dependencies among activities – namely, anything that can happen in a STS – poses serious scalability and privacy issues (see for instance the Cambridge Analytica scandal [62]) which have no silver bullet available, yet. In addition, promoting peripheral awareness may also have unintended consequences resulting in either (a) degraded performance instead of better collaboration and improved work practices, or (b) pushing away users from the intended means of communication promoted by the technical components of the STS: in the aforementioned open source software development example, for instance, mail and chat were barely used in favour of implicit coordination based on commits, enabled by awareness of meta-information provided to developers—such as commit diffs and messages, who took action, when, due to which item on the issue board, etc.

Finally, what is most challenging regarding adaptation, 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 [19], or deviating from usual behaviour as they are continuously adapting to the system, in turn, too. For instance, in reference to the aforementioned Tesla accident, the forensics analysis performed as part of the inquiry procedures attributed most of the responsibility for the fatal crash to the driver himself, which was overconfident (namely, had excessive, misplaced trust) in Tesla auto-pilot functionalities, thus stopped paying attention to notifications and loosened his supervision of the car. As a side note, Tesla was blamed too for marketing his driving assistant with the misleading name of *auto*-pilot—as well as the whole self-driving cars industry was put in the spotlight since commercial cars are not really "autonomous" nor "self-driving" as they barely touch level 3 in the classes of autonomy defined by SAE International [67].

2.2. Abstraction gap

In [41] 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 interactions between IoT devices, software, and people are interpreted as a kind of social relationship [7]. The proposed approach, called "Speaking Objects", puts emphasis on two traits of human-human interaction which sharply contrast with device-to-device interaction—hence, with the way devices would naturally interact with humans:

- humans better reason in terms of *goals* to be pursued and state of the affairs to be achieved, rather than by directly thinking at the specific individual actions they need to compose and carry out accordingly. Therefore, people usually interact at the same level of abstraction, that is, by expressing their own goals or the state of the affairs to be collectively achieved, rather than by explicitly detailing every single action each participant is expected to do, and when[2]
- humans also better reason in terms of complex situations, *state of affairs* holding in the past, at present, or desired for the future, and their differences/similarities, rather than by thinking at all the specific perceptions which collectively constitute a recognised situation. Accordingly, people usually interact by exchanging and debating about assertions of those situations and their properties, rather than by focussing on the specific perceptions and measurements that contribute to identify and characterise a familiar situation

This is quite the opposite of what most commercially available sensors, actuators, and general purpose devices enabler of the Internet of Things vision do at present days: they usually (a) provide specific measures, sampling a particular facet of a complex situation, which are then often composed by ad-hoc software algorithms (for instance, exploiting neural networks), and (b) react to explicit and detailed commands for undertaking specific actions—namely, not what to achieve, but what to do. 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 cooperate 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. The issue here corresponds essentially to an *abstraction gap* between the social side of STS and the technical one. Obviously, the preferred solution would be to increase the abstraction level understandable by devices, hence letting them recognise goals and situations, rather than decrease that of humans—forcing them to break goals and situations down to sequences of actions and composition of perceptions, respectively.

The aforementioned scenario of semi-autonomous driving brings an example of such an abstraction gap and its impact on, for instance, performance of the overall STS but also trust in technology by the human user (better discussed in next section). In [40], a study about the impact of providing to a human driver real-time verbal explanations of the actions taken by a semi-autonomous car delivers an interesting result, which confirms the reasoning explained above: authors found out that focussing on communicating *why* ("obstacle ahead") the action was taken, instead of *how* ("braking now"), improved the overall performance of the STS (the driver and the vehicle altogether) as well as trust in the vehicle by the driver. This

[2] An exception to this general habit is represented by learning/educational contexts, where a detailed "walkthrough" is instead beneficial.

result is in line with the considerations in [41]: communicating the why behind actions helps revealing the goal behind them ("the car decelerates *because* an obstacle is ahead *so as to* avoid collision"), whereas delivering the how does not provide any clue about the intentions behind action.

2.3. Trust and accountability

As clearly witnessed by initiatives such as the ACM "Statement on Algorithmic Transparency and Accountability" [2] we are already living in an *algocracy*, that is, a society in which increasingly pervasive, complex, and delicate aspects of our everyday lives are decided, or at least influenced, by computer algorithms [26]. A striking example is the filter bubble effect [58], caused, for instance, by the ranking algorithms running behind the news feed of social networks such as Facebook and Twitter. Nevertheless, less evident examples can be found almost everywhere: for instance, in algorithms regulating stock-trading, access to healthcare and insurance, and employment chances [51,48].

While living in an algocracy does not necessarily represent an issue on its own, the lack of *transparency* and *accountability* is: 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* in the system—to eventually stop using it. The path toward making algorithms accountable and transparent is out of the scope of this manuscript, and is full of challenges and open issues on its own, often heavily depending on the specific scenario where the STS is deployed to [29,33]. Nevertheless, what is relevant instead for our purpose of categorising the coordination approaches proposed in the literature is that accountability has been widely recognised as an essential property to be supported and safeguarded by the technical components of a STS [39]. For instance, Chopra and Singh [25] built a novel theory of *interaction-oriented software engineering* by advocating the prominent role played by accountability, social expectations, and commitments in the way social interactions between humans and organisations play out in the real world.

As regards trust, literature on normative multi-agent systems is a fertile source of technical approaches to promote and leverage trust between interacting parties, often times through a scheme of sanctions and rewards meant to discourage or incentivise, respectively, non-compliant and compliant behaviours [50]. There, the technical components of the STS have access to a representation of the *norms* to consider, as stemming from social protocols and conventions, as well as the means to enforce compliance of users. In reference to the aforementioned case study about open source software collaborative development, among the many reasons why also programmers working in private companies rely on such kind of STS to host their code is accountability: the platform in fact records any action taken by any user, hence keeping track of who did what, when, and often times why, too, is extremely easy. Then, not only the platform enables collaboration to deliver a quality software artefact, but also supports and eases governance.

3. Approaches to coordination in STS

As for the challenges discussed in Section 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 a handful of coordination approaches, selected due to their explicit attention to one or more of the aforementioned challenges, and categorise them as being technical-to-social or social-to-technical. While doing so, we emphasise expected benefits and potential shortcomings so as to highlight complementarity of approaches.

3.1. Technical-to-social

We define as "technical-to-social" those coordination approaches mostly concerned with how to *exploit the technical components of the STS to* (a) *enhance the interaction capabilities of human users as well as to* (b) *exploit the activities, side-effects, and outcomes of their collaborations to improve themselves*—as depicted in Fig. 1. Observation-based coordination, there including models based on stigmergy, and coordination inspired to self-organisation in natural systems are examples of this kind of approaches, whose strengths and weaknesses are summarised in Table 2.

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 [63].

The most well known example of observation-based coordination is *stigmergy*. The term has been originally introduced by Grassé [34] 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. Then, throughout the years, it undergone many generalisations/specialisations/extensions [60,65,56]. 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

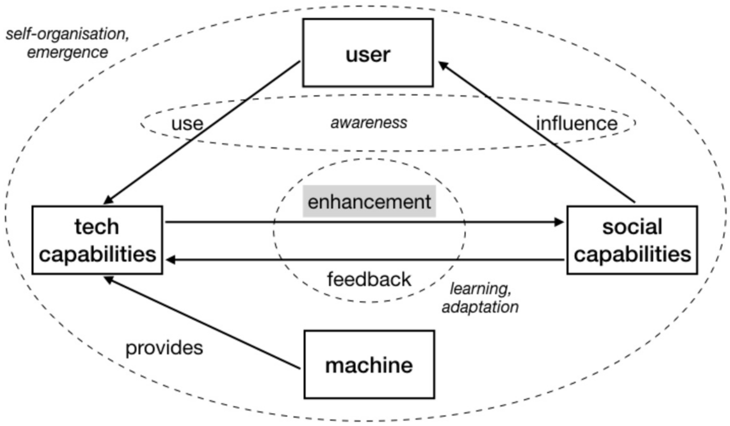

Fig. 1. "Technical-to-social" approaches to coordination in STS: focus is on *enhancement* of social capabilities through technology.

- 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 [69]—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 [75,68]—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 Ricci et al. [65] introduced 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 [19].

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) [24] 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 [23].

BIC clearly represents a step further 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 [21].

Table 2
"Technical-to-social" approaches to coordination in STS: strengths and weaknesses.

Research thread	Strengths	Weaknesses
observation-based coordination	well-studied, operational formalisations, awareness enabler, high abstraction level	privacy trade-offs, scalability issues, rationality assumption
self-organising coordination	computationally simple, scales well, adaptation enabler, promotes system autonomy	unpredictability trade-offs, accountability trade-offs, difficult to design, difficult to control

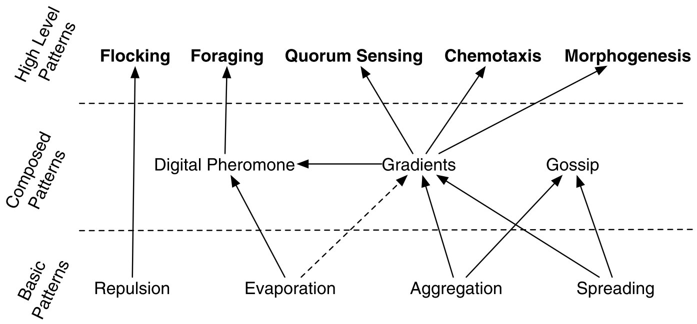

Fig. 2. Design patterns according to Fernandez-Marquez et al. [31]. (Dashed) Arrows indicate (optional) composition.

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 constrain awareness and peripheral awareness through 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. For these reasons, we categorise observation-based coordination approaches in general as "technical-to-social" approaches: they exploit computational mechanisms to enhance and augment the opportunities for interaction, the reach of communications, and effectiveness of collaborations. For instance, the aforementioned open source software development case study explicitly points at stigmergic coordination to explain the high level of socio-technical congruence achieved by the STS despite the lack of explicit communications [16].

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 [28]. Besides self-* properties (self-healing, self-adaptation, self-configuration, etc.), 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 [74]: 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).

Decentralised approaches to self-organisation have been widely studied, as regards both computation [49,8] and coordination [27,31], and already proved to be effective in dealing with the many issues of distributed computing in general [81]. For instance, Fernandez-Marquez et al. [31] 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. 2 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

As an example, gradients compose spreading and aggregation (optionally evaporation, too) to dynamically build routing paths inspired by force fields in physics [43], 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. As such, we categorise coordination approaches based on self-organisation inspired by the

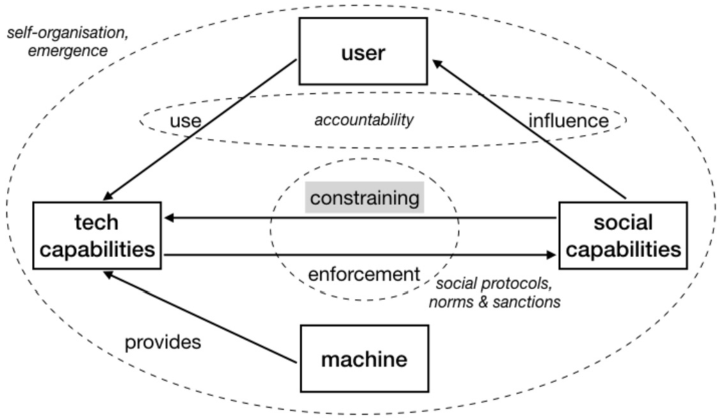

Fig. 3. "Social-to-technical" approaches to coordination in STS: focus is on *constraining* the social capabilities through technology.

aforementioned kind of natural systems as "technical-to-social": both decentralisation and situatedness, in fact, are exploited to let emergent phenomena arising from self-organisation amplify and harmonise the effects of local interactions in a coherent way. Gossiping, for instance, is a clear example of amplification of local interactions whose global effect enhances the communication reach of isolated individuals.

3.2. Social-to-technical

We define as "social-to-technical" the coordination approaches mostly concerned with *computationally represent social relationships, protocols, expectations, conventions, norms, etc. so as to let technical components of the STS facilitate, promote, manipulate, and enforce them*—as depicted in Fig. 3. The concepts of social protocol and commitment and the mechanisms for computational argumentation, as well as the models of e-Institutions and the framework of nMAS are all examples of social-to-technical approaches, whose strengths and weaknesses are summarised in Table 3.

Social protocols. Chopra and Singh [25] proposed a brand new software engineering paradigm they call *Interaction-Oriented Software Engineering* (IOSE). There, the technical components of an STS, named "SE-machines", are implementations of social protocols specifying how social relationships, expressing the *accountability* of the interacting parties, evolve during interaction. In particular, they focus on a handful of relationships: commitment to take action, witness the truth of an assertion, prohibit another party to disclose some information or act in a certain way, establish a generic "friendship" relation. Then, the respective protocols only specify which expectations a participant may legitimately have on another one, and who is accountable for what.

SE-machines are then expected to provide support to representation, manipulation, adaptation, enforcement, monitoring, progress, etc. of such a set of protocols. While doing so, every operational detail which has no impact on the social protocol should not be considered. For instance, only if the social protocol itself considers robustness to malicious or faulty behaviour a required property, then enforcement of sanctions should be factored into the implementation of the SE-machine. In general, thus, the SE-machine is only concerned with guaranteeing compliance to the social protocol, whatever it specifies. Also, if and when the principals (people or organisations) served by the SE-machine have their own set of requirements, different but not in contrast with the ones expressed by the social protocol, the SE machine can implement those requirements.

The technical mechanisms used by Chopra and Singh [25] to achieve their goal relies on the notion of *commitment*, also exploited by Baldoni et al. [12]. In brief, a commitment states that in case some pre-conditions hold, a debtor will bring about some post-conditions of interest for its creditor. Through this simple notion, a relationship of expectations and accountability is established between two interacting parties: the creditor expects the debtor to obey its commitment, while the debtor accepts to be accountable for that. Handling commitments, in the sense of monitoring, coordinating, and

Table 3
"Social-to-technical" approaches to coordination in STS: strengths and weaknesses.

Research thread	Strengths	Weaknesses
social protocols	separation of concerns, accountability enabler	no operational formalisations, more theory than practice
argumentation	high abstraction level, well-studied, trust enabler	scalability, computationally expensive, more theory than practice
e-Institutions & *nMAS*	well-studied, operational formalisations, accountability enabler, fine control	scalability, decentralisation difficult, adaptation difficult

enforcing them (through a sanctioning scheme, for instance), is an effective way to realise in computational terms the social protocols playing a crucial role in STS.

Given the above description of the concepts of social protocol, social machine, and commitment, it is apparent that coordination approaches based on these are "social-to-technical" according to our categorisation: they are concerned with how to model and engineer social relationships, their properties, dynamics, and the means to change them in computational terms.

Argumentation. *Computational argumentation* exploits computational techniques to automatically analyse and build arguments and their relationships [78], there including generating explanations and justifications of decision making [20]. For instance, argumentation-based negotiation applies argumentation principles to negotiation-based coordination in multi-agent systems [64]. 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.

As already discussed in reference to the semi-autonomous driving scenario described in Section 2, 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 [51] to algorithms in general [48] – as witnessed by the recent "transparency initiative" endorsed by many organisations worldwide.[3]

We categorise argumentation-based coordination approaches as "social-to-technical" as they strive to capture in computational terms the engagement and progress rules of different kinds of human dialogues, all aimed at letting people cooperate for decision making. Hence, here, the social protocol that agents must abide to is set by the rules of the argumentation framework taken as reference, and the set of norms and regulations stem from the rules deciding which arguments are admissible, better than others, and, ultimately, win the debacle.

e-Institutions & nMAS. The multi-agent systems community has widely used the concept of *institution* [55] in computational systems as a means to model and implement a variety of STS, by regulating interaction among autonomous participants towards achievement of a collective goal by reifying conventions, rules, and *norms* [14], thus giving birth to the so-called *electronic Institutions* concept (e-Institutions) [70]. e-Institutions are then the computational framework within which agents interactions take place, there including the norms as well as the rewards and sanctions necessary to promote and enforce compliant behaviour and discourage non-compliance.

The above description of e-Institutions makes apparent a strong connection with *normative MAS* (nMAS), as those MAS where achievement of and compliance to a desirable global behaviour is pursued through normative constraints [13] regulating the interactions amongst artificial agents to promote coordination, cooperation, and support decision-making. Most of the research efforts in this area are indeed concerned with how to represent norms efficiently, how to let agents observe, understand, and manipulate them, how to let the technical platform monitor and/or enforce compliance, and possibly act in reward or punishment of agents behaviour.

In summary, e-Institutions and nMAS are closely related as the former usually embeds the latter while the latter often times implicitly defines the former, and are clearly belonging to the "social-to-technical" category given their focus on how to translate elements of the social world (norms, rules, conventions, sanctions, etc.) into the computational one.

4. Selected proposed approaches: friends or foes?

With the aim of showing how the research works described in Section 3 can be actually exploited in the real-world to tackle the challenges discussed in Section 2, the remainder of this section reports on a few promising proposals each integrating some of the approaches in its own unique way: the $\mathcal{M}$olecules $\mathcal{o}$f $\mathcal{K}$nowledge 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, JaCaMo+ puts at the center the notion of social commitment as an accountability enabler.

4.1. Self-organising knowledge management with $\mathcal{MoK}$

The $\mathcal{M}$olecules $\mathcal{o}$f $\mathcal{K}$nowledge model [46] ($\mathcal{MoK}$) 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. $\mathcal{MoK}$ is built around the integration of a biochemical metaphor [79] and BIC [24]: the former defines how to carry out computations, while the latter how to manage interactions.

[3] http://www.transparency-initiative.org/.

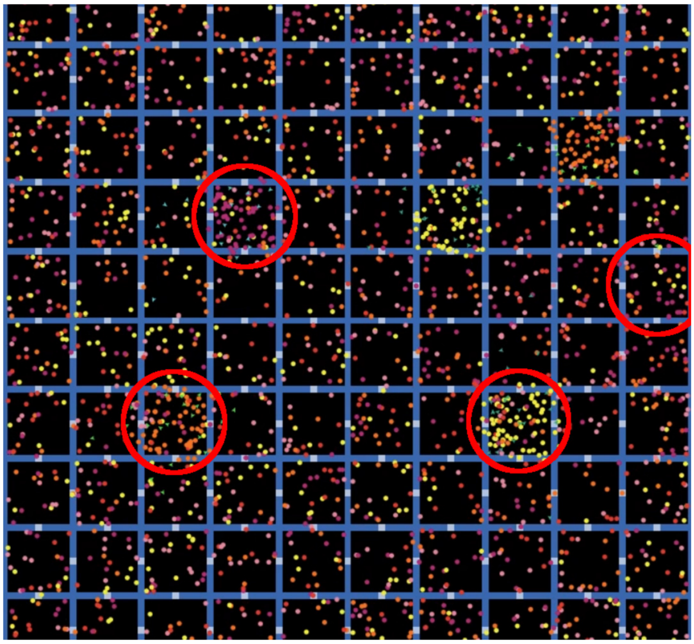

Fig. 4. Clustering of similar information appearing in $\mathcal{MoK}$ *by emergence*, as a result of users' interactions (image from [47]).

Accordingly, a $\mathcal{MoK}$ 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 $\mathcal{MoK}$ *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 [44].

In [47], a citizen journalism scenario is taken as a case study: there, users share a $\mathcal{MoK}$-coordinated IT platform for retrieving, assembling, and publishing news stories. They use the $\mathcal{MoK}$ 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 $\mathcal{MoK}$ reactions. Namely, the $\mathcal{MoK}$ middleware exploits users' (local) interactions to improve the (global) spatial organisation of information (Fig. 4): whenever users implicitly manifest interest in information, $\mathcal{MoK}$ 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.

4.2. Traffic control with speaking objects

In [42] 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 [18].

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 [81], argumentation also embraces humans-in-the-loop by enabling

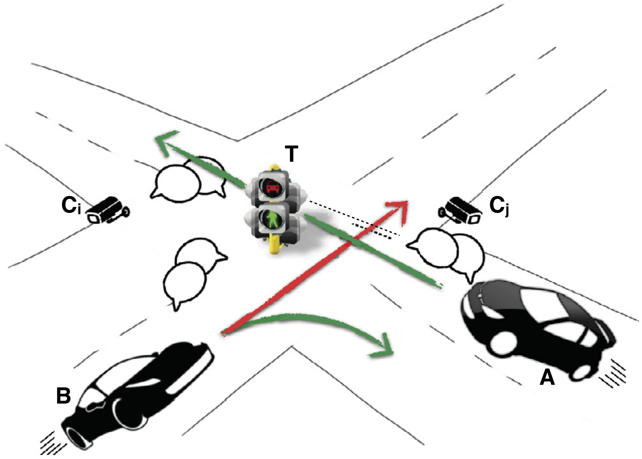

Fig. 5. Argumentation-based intersection management (image from [41]).

users to interact in natural language [20], and deals with the issues of trust and algocracy by making explanations and justifications of decision making available and amenable of inspection and interpretation by human supervisors.

In [41], 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 argumenting with the traffic light about who has the right of way and who should instead stop and wait (Fig. 5). 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.

4.3. Business processes coordination with social commitments

In [11] the authors enhance business artefacts with a *normative layer* based on the notion of *social commitment*, with the goal of enabling forms of coordination naturally promoting and enforcing *accountability* of participants actions and interactions. The work starts from the premise that if in an institutional setting norms are explicitly represented and known by the participants of an organisation, then it is possible to leverage *expectations* about others behaviours for the sake of coordination. Based on this, the authors exploit the theoretical framework of social protocols [25] to ground these expectations in the notion of social commitment: a relation known to a debtor and a creditor stating that the former commits to the latter to bring about the consequent (post-conditions) of a norm as soon as the antecedent (pre-conditions) holds.

The way in which such commitments evolve over time as a consequence of agents interactions, especially, of their adoption of roles and agreement about shared goals, is described in the ADOPT protocol formalised by the same authors in [12]. There, the FIPA Contract-Net protocol [72] and a combination with the FIPA Request Interaction Protocol [32] regulate progress of the two stages in which ADOPT develops: the role adoption, and the goal agreement stages. In the former stage, agents willing to join the organisation must adopt a certain role, which implies that they accept the set of commitments that such a role poses; in the latter stage, commitments are checked for satisfaction, violation, new commitments can be created or old one canceled, depending on what the agent does to pursue the organisational goals ascribed to it by the adopted role.

To demonstrate effectiveness of the commitment-based approach, in [11] the hiring process scenario therein described is dealt with by using the JaCaMo+ commitment-based infrastructure [10]. JaCaMo+ is an extension of the JaCaMo platform for MAS development [15]: the latter focus on agents, environment, and organisational aspects programming whereas the former enhances it with commitments. Agent programming in JaCaMo is supported by the Jason programming language [17], whereas environment engineering is enabled by the notion of artefact provided by the CArtAgO framework [66]. In JaCaMo+, the former is exploited to let interacting agents react to a change in the state of commitments, while the latter is exploited to build a custom coordination artefact tracking the normative state of the MAS—that is, the collection of commitments.

4.4. Integration as key

Now that a brief account of the approaches to deal with coordination in STS as well as a few selected models and technologies have been presented, considerations about their complementarity and integration can be done. First of all, we want to stress that approaches belonging to the "technical-to-social" category and those in the "social-to-technical" are complementary by the very nature of the taxonomy: as depicted in Fig. 6, the former stresses the extent to which the technical part of the STS can enhance, augment, improve, facilitate the social interactions expected to happen in the social side, whereas the latter instead stresses the completeness and accuracy of the computational representations of the

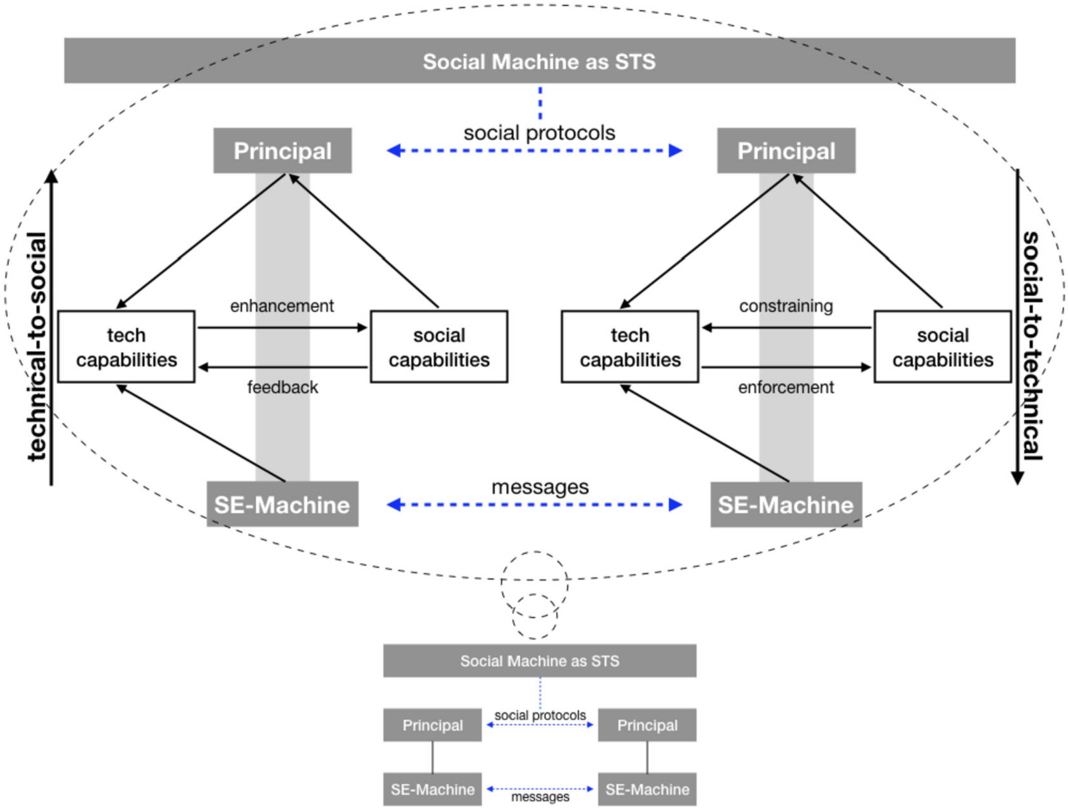

Fig. 6. Complementarity of the proposed approaches, integrated in the Social Machines framework conceived by Chopra and Singh [25].

social relationships, norms, rules, protocols that regulate social activity, but they are not conflicting in any way. Accordingly, in Fig. 6 we depicted one relationship between a principal and its "SE-Machine" (according to Chopra and Singh [25] terminology) as being modelled and implemented through a "technical-to-social" approach (left-side), while another one as being designed through a "social-to-technical" approach (right-side). As an aside, Fig. 6 also frames our categorisation into the generic and theoretical framework set by Chopra and Singh [25]: indeed, our proposed taxonomy fills a gap in their conceptual modelling, that is, how to translate the social protocols specified at the social layer into the lower-level interaction means provided by the technical side (or the other way around). In particular, our taxonomy naturally provides two *mindsets* with which the issue of "connecting" the two sides of an STS (the social and the technical) can be dealt with: one is by focussing on which mechanisms enable to exploit the social interactions mediated by technology at the advantage of the technical platform supporting the STS itself, enabling its continuous improvement and/or adaptation to the users' needs, the other is by focussing on the formalisms able to specify all the properties and constraints belonging to the social side os the STS in technical terms, so as to be monitored, enforced, preserved during operation.

Again, we want to emphasise that these two mindsets are not conflicting, rather, it is highly desirable they are both taken into consideration while developing an STS, to combine the advantages of the approaches belonging to either category and mitigating their shortcomings. For instance, $\mathcal{MoK}$ 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 $\mathcal{MoK}$ vision, and is made available to speaking and hearing objects as they need premises to either support or attack each other arguments. There, whenever action has to be taken, commitments amongst speaking and hearing objects establish accountability, hence liability, relationships, which can be used in the open system represented by a Smart City to deliver rewards or punishments in an institutional, normative framework meant to govern the global goals of the system despite non-compliant behaviour of participating agents.

The simple exercise of putting the benefits and shortcomings of "technical-to-social" (Table 2) and "social-to-technical" (Table 3) approaches against each other, and against the requirements laid out in Table 1 suffices to highlight how none of the approaches mentioned in this paper is capable of dealing with everything alone. Rather, by jointly exploiting mechanisms and borrowing concepts from both worlds it is much more realistic to conceive an STS able to deal with all the nuances of social interaction mediated by technology in an efficient and effective way.

5. Conclusion & 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 Section 4. There, for instance, $\mathcal{MoK}$ 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. 4. 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 approaches discussed in Section 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.

We hope that the analysis and considerations expressed in this paper help researchers in the many fields devoted to deal with coordination in socio-technical systems to find novel stimuli for pushing their research not only further beyond the state of art, but also to reach out to other disciplines and research areas seeking for the kind of cross-fertilisation needed to make the envisioned integration attempts successful.

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] M.S. Ackerman, The intellectual challenge of cscw: the gap between social requirements and technical feasibility, Hum.-Comput. Interact. 15 (2000) 179–203.
[2] ACM US Public Policy Council, Statement on algorithmic transparency and accountability, https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf, 2017.
[3] H. Aldewereld, S. Álvarez-Napagao, V. Dignum, J. Jiang, W. Vasconcelos, J. Vázquez-Salceda, OperA/ALIVE/OperettA, Springer International Publishing, Cham, 2016, pp. 173–196.
[4] H. Aldewereld, O. Boissier, V. Dignum, P. Noriega, J. Padget, Social Coordination Frameworks for Social Technical Systems, 1 ed., Law, Governance and Technology, vol. 30, Springer International Publishing, 2016.
[5] G. Andrighetto, G. Governatori, P. Noriega, L.W.N. van der Torre (Eds.), Normative Multi-Agent Systems, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, http://drops.dagstuhl.de/opus/portals/dfu/index.php?semnr=13003.
[6] Association for Psychological Science, Self-driving cars face psychological speed bumps, https://www.psychologicalscience.org/news/motr/autonomous-vehicles-need-to-make-psychological-inroads-before-widespread-adoption.html, 2017.
[7] L. Atzori, A. Iera, G. Morabito, M. Nitti, The social internet of things (siot)–when social networks meet the internet of things: concept, architecture and network characterization, Comput. Netw. 56 (2012) 3594–3608.
[8] O. Babaoglu, G. Canright, A. Deutsch, G.A.D. Caro, F. Ducatelle, L.M. Gambardella, N. Ganguly, M. Jelasity, R. Montemanni, A. Montresor, et al., Design patterns from biology for distributed computing, ACM Trans. Auton. Adapt. Syst. 1 (2006) 26–66.
[9] R. Badham, C. Clegg, T. Wall, Socio-Technical Theory, Wiley, 2000.
[10] M. Baldoni, C. Baroglio, F. Capuzzimati, R. Micalizio, Commitment-based agent interaction in JaCaMo+, Fundam. Inform. 159 (2018) 1–33.
[11] M. Baldoni, C. Baroglio, F. Capuzzimati, R. Micalizio, Process coordination with business artifacts and multiagent technologies, J. Data Semant. 8 (2019) 99–112, https://doi.org/10.1007/s13740-019-00100-8.
[12] M. Baldoni, C. Baroglio, K.M. May, R. Micalizio, S. Tedeschi, Computational accountability in mas organizations with adopt, Appl. Sci. 8 (2018), https://doi.org/10.3390/app8040489, https://www.mdpi.com/2076-3417/8/4/489.
[13] T. Balke, C. da Costa Pereira, F. Dignum, E. Lorini, A. Rotolo, W. Vasconcelos, S. Villata, Norms in MAS: definitions and related concepts, in: G. Andrighetto, G. Governatori, P. Noriega, L.W.N. van der Torre (Eds.), Normative Multi-Agent Systems, in: Dagstuhl Follow-Ups, vol. 4, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013, pp. 1–31, http://drops.dagstuhl.de/opus/volltexte/2013/3998.
[14] G.D. Bibu, Modelling organisations as electronic institutions, Int. J. Sci. Technol. Res. 8 (2019) 37–43.
[15] O. Boissier, R.H. Bordini, J.F. Hübner, A. Ricci, A. Santi, Multi-agent oriented programming with JaCaMo, Sci. Comput. Program. 78 (2013) 747–761, https://doi.org/10.1016/j.scico.2011.10.004.
[16] F. Bolici, J. Howison, K. Crowston, Coordination without discussion? Socio-technical congruence and stigmergy in free and open source software projects, in: Socio-Technical Congruence Workshop, 2009.
[17] R.H. Bordini, J.F. Hübner, M. Wooldridge, Programming Multi-Agent Systems in AgentSpeak Using Jason, vol. 8, John Wiley & Sons, 2007.
[18] K. Bourzac, Millimeter-scale computers: now with deep-learning neural networks on board, https://goo.gl/sciVTC, 2017.
[19] Y. Brun, G. Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Müller, M. Pezzè, M. Shaw, Engineering self-adaptive systems through feedback loops, in: Software Engineering for Self-Adaptive Systems, in: Lecture Notes in Computer Science, vol. 5525, Springer, 2009.
[20] M.W. Caminada, R. Kutlak, N. Oren, W.W. Vasconcelos, Scrutable plan enactment via argumentation and natural language generation, in: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems, 2014.
[21] C. Castelfranchi, Modelling social action for AI agents, Artif. Intell. 103 (1998) 157–182, https://doi.org/10.1016/S0004-3702(98)00056-3.
[22] C. Castelfranchi, Silent agents: from observation to tacit communication, in: Advances in Artificial Intelligence, Springer, 2006, pp. 98–107.
[23] C. Castelfranchi, G. Pezzullo, L. Tummolini, Behavioral implicit communication (BIC): communicating with smart environments via our practical behavior and its traces, Int. J. Ambient Comput. Intell. 2 (2010) 1–12, https://doi.org/10.4018/jaci.2010010101.

[24] C. Castlefranchi, From conversation to interaction via behavioral communication: for a semiotic design of objects, environments, and behaviors, in: Theories and Practice in Interaction Design, 2006, pp. 157–179.

[25] A.K. Chopra, M.P. Singh, From social machines to social protocols: software engineering foundations for sociotechnical systems, in: Proceedings of the 25th International Conference on World Wide Web, 2016.

[26] J. Danaher, The threat of algocracy: reality, resistance and accommodation, Philos. Technol. 29 (2016) 245–268, https://doi.org/10.1007/s13347-015-0211-1.

[27] T. De Wolf, T. Holvoet, Design patterns for decentralised coordination in self-organising emergent systems, in: Engineering Self-Organising Systems, Springer, 2007, pp. 28–49.

[28] G. Di Marzo Serugendo, M.P. Gleizes, A. Karageorgos, Self-organisation and emergence in MAS: an overview, Informatica 30 (2006) 45–54.

[29] N. Diakopoulos, Accountability in algorithmic decision making, Commun. ACM 59 (2016) 56–62, https://doi.org/10.1145/2844110.

[30] D.M.V. California, Report of traffic collision involving an autonomous vehicle (OL 316), https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/autonomousveh_ol316+, 2019.

[31] J.L. Fernandez-Marquez, G. Di Marzo Serugendo, S. Montagna, M. Viroli, J.L. Arcos, Description and composition of bio-inspired design patterns: a complete overview, Nat. Comput. 12 (2013) 43–67, https://doi.org/10.1007/s11047-012-9324-y.

[32] Foundation for Intelligent Physical Agents, Fipa request interaction protocol specification, online, http://www.fipa.org/specs/fipa00026/SC00026H.html, 2002.

[33] B. Goodman, S. Flaxman, European Union regulations on algorithmic decision-making and a "right to explanation", arXiv e-prints, arXiv:1606.08813, 2016.

[34] P.P. Grassé, 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 Soc. 6 (1959) 41–80, https://doi.org/10.1007/BF02223791.

[35] A. Halsey, Ntsb says driver in fatal tesla crash was overreliant on the car's 'autopilot' system, https://wapo.st/2I9KoJG, 2017.

[36] J. Hendler, T. Berners-Lee, From the semantic web to social machines: a research challenge for AI on the world wide web, in: Special Review Issue, Artif. Intell. 174 (2010) 156–161, https://doi.org/10.1016/j.artint.2009.11.010.

[37] T. Holstein, G. Dodig-Crnkovic, P. Pelliccione, Ethical and social aspects of self-driving cars, CoRR, arXiv:1802.04103, http://arxiv.org/abs/1802.04103, 2018.

[38] E. Hutchins, Cognition in the Wild, MIT Press, 1995.

[39] N.R. Jennings, L. Moreau, D. Nicholson, S. Ramchurn, S. Roberts, T. Rodden, A. Rogers, Human-agent collectives, Commun. ACM 57 (2014) 80–88, https://doi.org/10.1145/2629559.

[40] J. Koo, J. Kwac, W. Ju, M. Steinert, L. Leifer, C. Nass, Why did my car just do that? Explaining semi-autonomous driving actions to improve driver understanding, trust, and performance, Int. J. Interact. Des. Manuf. 9 (2015) 269–275.

[41] M. Lippi, M. Mamei, S. Mariani, F. Zambonelli, An argumentation-based perspective over the social IoT, IEEE Int. Things J. (2017), https://doi.org/10.1109/JIOT.2017.2775047.

[42] M. Lippi, M. Mamei, S. Mariani, F. Zambonelli, Coordinating distributed speaking objects, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 1949–1960.

[43] M. Mamei, F. Zambonelli, L. Leonardi, Co-fields: towards a unifying approach to the engineering of swarm intelligent systems, in: P. Petta, R. Tolksdorf, F. Zambonelli (Eds.), Engineering Societies in the Agents World III, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 68–81.

[44] S. Mariani, Coordination of Complex Sociotechnical Systems - Self-organisation of Knowledge in MoK, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer, 2016.

[45] S. Mariani, Coordination of complex socio-technical systems: challenges and opportunities, in: M. Mazzara, I. Ober, G. Salaün (Eds.), Software Technologies: Applications and Foundations, Springer International Publishing, Cham, 2018, pp. 295–310.

[46] S. Mariani, A. Omicini, Molecules of knowledge: self-organisation in knowledge-intensive environments, in: G. Fortino, C. Bădică, M. Malgeri, R. Unland (Eds.), Intelligent Distributed Computing VI, Springer, 2013, pp. 17–22.

[47] S. Mariani, A. Omicini, Anticipatory coordination in socio-technical knowledge-intensive environments: behavioural implicit communication in MoK, in: M. Gavanelli, E. Lamma, F. Riguzzi (Eds.), AI*IA 2015, Advances in Artificial Intelligence, in: Lecture Notes in Computer Science, vol. 9336, Springer International Publishing, 2015, pp. 102–115 (Chapter 8).

[48] L. Medsker, Algorithmic transparency and accountability – AI matters, https://sigai.acm.org/aimatters/blog/2017/06/01/algorithmic-transparency-and-accountability/, 2017.

[49] R. Nagpal, A catalog of biologically-inspired primitives for engineering self-organization, in: Engineering Self-Organising Systems, Springer, 2004, pp. 53–62.

[50] L.G. Nardin, T. Balke-Visser, N. Ajmeri, A.K. Kalia, J.S. Sichman, M.P. Singh, Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems, Knowl. Eng. Rev. 31 (2016) 142–166, http://dblp.uni-trier.de/db/journals/ker/ker31.html#NardinBAKSS16.

[51] Nature, More accountability for big-data algorithms, Nature 537 (2016) 449, https://doi.org/10.1038/537449a.

[52] C. Nikolai, G. Madey, Tools of the trade: a survey of various agent based modeling platforms, J. Artif. Soc. Soc. Simul. 12 (2009) 2, http://jasss.soc.surrey.ac.uk/12/2/2.html.

[53] P. Noriega, D. de Jonge, Electronic Institutions: The EI/EIDE Framework, Springer International Publishing, Cham, 2016, pp. 47–76.

[54] P. Noriega, J. Padget, H. Verhagen, M. D'Inverno, Towards a framework for socio-cognitive technical systems, in: Revised Selected Papers of the International Workshops on Coordination, Organizations, Institutions, and Norms in Agent Systems, Springer-Verlag New York, Inc., 2015.

[55] D.C. North, Institutions, Institutional Change and Economic Performance, Political Economy of Institutions and Decisions, 1991.

[56] A. Omicini, Agents writing on walls: cognitive stigmergy and beyond, in: F. Paglieri, L. Tummolini, R. Falcone, M. Miceli (Eds.), The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi, in: Tributes, vol. 20, College Publications, London, 2012, pp. 543–556 (Chapter 29).

[57] W.J. Orlikowski, The duality of technology: rethinking the concept of technology in organizations, Organ. Sci. 3 (1992) 398–427.

[58] E. Pariser, The Filter Bubble: What the Internet is Hiding From You, Penguin, UK, 2011.

[59] S.Y. Park, S.Y. Lee, Y. Chen, The effects of EMR deployment on doctors' work practices: a qualitative study in the emergency department of a teaching hospital, Int. J. Med. Inform. 81 (2012) 204–217, https://doi.org/10.1016/j.ijmedinf.2011.12.001.

[60] H.V.D. Parunak, A survey of environments and mechanisms for human-human stigmergy, in: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems II, in: LNCS, vol. 3830, Springer, 2006, pp. 163–186.

[61] H.V.D. Parunak, S. Brueckner, J. Sauter, Digital pheromone mechanisms for coordination of unmanned vehicles, in: C. Castelfranchi, W.L. Johnson (Eds.), 1st International Joint Conference on Autonomous Agents and Multiagent systems, ACM, New York, NY, USA, 2002, pp. 449–450.

[62] N. Persily, Can democracy survive the internet?, J. Democr. 28 (2017) 63–76.

[63] M. Piunti, C. Castelfranchi, R. Falcone, Anticipatory coordination through action observation and behavior adaptation, in: Proceedings of AISB, 2007.

[64] I. Rahwan, S.D. Ramchurn, N.R. Jennings, P. Mcburney, S. Parsons, L. Sonenberg, Argumentation-based negotiation, Knowl. Eng. Rev. 18 (2003) 343–375, https://doi.org/10.1017/S0269888904000098.

[65] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, E. Oliva, Cognitive stigmergy: towards a framework based on agents and artifacts, in: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for MultiAgent Systems III, in: LNCS, vol. 4389, Springer, 2007, pp. 124–140.

[66] A. Ricci, M. Piunti, M. Viroli, A. Omicini, Environment Programming in CArtAgO, Springer US, Boston, MA, 2009, pp. 259–288.
[67] SAE International, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2018.
[68] K. Schmidt, I. Wagner, Ordering systems: coordinative practices and artifacts in architectural design and planning, Comput. Support. Coop. Work 13 (2004) 349–408.
[69] G.D.M. Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S.K. Mostéfaoui, O.F. Rana, M. Ulieru, P. Valckenaers, C. Van Aart, Self-Organisation: Paradigms and Applications, Springer, 2003.
[70] C. Sierra, J.A. Rodríguez-Aguilar, P. Noriega, M. Esteva, J.L. Arcos, Engineering multi-agent systems as electronic institutions, UPGRADE Eur. J. Inform. Prof. V (2004) 33–39.
[71] M.P. Singh, M. Arrott, T. Balke, A.K. Chopra, R. Christiaanse, S. Cranefield, F. Dignum, D. Eynard, E. Farcas, N. Fornara, F. Gandon, G. Governatori, H.K. Dam, J. Hulstijn, I. Krueger, H.P. Lam, M. Meisinger, P. Noriega, B.T.R. Savarimuthu, K. Tadanki, H. Verhagen, S. Villata, The uses of norms, in: G. Andrighetto, G. Governatori, P. Noriega, L.W.N. van der Torre (Eds.), Normative Multi-Agent Systems, in: Dagstuhl Follow-Ups, vol. 4, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013, pp. 191–229, http://drops.dagstuhl.de/opus/volltexte/2013/4004.
[72] Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Trans. Comput. C-29 (1980) 1104–1113, https://doi.org/10.1109/TC.1980.1675516.
[73] I. Sommerville, G. Baxter, Socio-technical systems: from design methods to systems engineering, Interact. Comput. 23 (2010) 4–17, https://doi.org/10.1016/j.intcom.2010.07.003.
[74] L.A. Suchman, Situated actions, in: Plans and Situated Actions: The Problem of Human-Machine Communication, Cambridge University Press, New York, NY, USA, 1987, pp. 49–67 (Chapter 4).
[75] T. Susi, T. Ziemke, Social cognition, artefacts, and stigmergy: a comparative analysis of theoretical frameworks for the understanding of artefact-mediated collaborative activity, Cogn. Syst. Res. 2 (2001) 273–290.
[76] E.L. Trist, The Evolution of Socio-Technical Systems: A Conceptual Framework and an Action Research Program, Ontario Ministry of Government Services, 1981.
[77] L. Tummolini, C. Castelfranchi, A. Ricci, M. Viroli, A. Omicini, "Exhibitionists" and "voyeurs" do it better: a shared environment approach for flexible coordination with tacit messages, in: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems, in: LNAI, vol. 3374, 2005, pp. 215–231.
[78] F.H. Van Eemeren, R. Grootendorst, R.H. Johnson, C. Plantin, C.A. Willard, Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments, Routledge, 2013.
[79] M. Viroli, M. Casadei, Biochemical tuple spaces for self-organising coordination, in: J. Field, V.T. Vasconcelos (Eds.), Coordination Languages and Models, in: LNCS, vol. 5521, Springer, Lisbon, Portugal, 2009, pp. 143–162.
[80] B. Whitworth, Socio-technical systems, in: Encyclopedia of Human Computer Interaction, 2006, pp. 533–541.
[81] F. Zambonelli, A. Omicini, B. Anzengruber, G. Castelli, F.L. DeAngelis, G. Di Marzo Serugendo, S. Dobson, J.L. Fernandez-Marquez, A. Ferscha, M. Mamei, S. Mariani, A. Molesini, S. Montagna, J. Nieminen, D. Pianini, M. Risoldi, A. Rosi, G. Stevenson, M. Viroli, J. Ye, Developing pervasive multi-agent systems with nature-inspired coordination, in: Special Issue "10 years of Pervasive Computing" in Honor of Chatschik Bisdikian, Pervasive Mob. Comput. 17 (2015) 236–252, https://doi.org/10.1016/j.pmcj.2014.12.002.