FoCAS manifesto: a roadmap to the future of collective adaptive systems

Introduction

The socio-technical fabric of our society more and more depends on systems that are constructed as a collective of heterogeneous components and that are tightly entangled with humans and social structures. Their components increasingly need to be able to evolve, collaborate and function as a part of an artificial society.
The FoCAS coordination action (http://www.focas.eu/about-focas/) is an umbrella project which aims to integrate, coordinate and help increase visibility for research carried out in the FOCAS Proactive Initiative and in research fields related to collective adaptive systems.

One of the key activities of the project was roadmapping, i.e. defining the future research agenda by means of consultations of experts in the field.
This manifesto represents the result of the three-year work of the project, in which we have collected material, interviewed people, run consultation events, analysed and summarised all the collected inputs.

We hope that it will be useful to all researchers that aims at working in the field of collective adaptive systems or related fields.

History of the document

Date	Status	Contributors	Notes
July 2015	First draft (0.1)	Participants to the FoCAS meeting in Bologna	
September 2015	Second Draft (0.2)	Participants to the FoCAS Science Café in Boston	
November 2015	Third draft (0.3)	Participants to the meeting in London	
January 2016	Fourth draft (0.4)	Participants to the Video Sprint in Edinburgh	
February 2016	Final version	FoCAS executive board	

People involved

Participants at the FoCAS research agenda “lock-in” meeting in Bologna (1-3 July 2015)

Name	Affiliation	City	Country
Alois Ferscha	University of Linz	Linz	A
Andrea Omicini	Università di Bologna	Bologna	I
Antonio Bucchiarone	FBK	Trento	I
Giacomo Cabri	University of Modena and Reggio Emilia	Modena	I
Giovanna Di Marzo	University of Geneve	Geneve	CH
Jacqueline Heinerman	Vrije University Amsterdam	Amsterdam	NL
José Halloy	Université Paris Diderot	Paris	F
Mirko Viroli	Università di Bologna	Cesena	I
Peter Lewis	Aston University	Birmingham	UK
Rocco De Nicola	IMT	Lucca	I
Stefano Mariani	Università di Bologna	Bologna	I
Stuart Anderson	University of Edinburgh	Edinburgh	UK
Thomas Schmickl	University of Graz	Graz	A

Participants at the FoCAS Executive meeting held in London (18 November 2015)

Name	Affiliation	City	Country
Alois Ferscha	University of Linz	Linz	A
Giacomo Cabri	University of Modena and Reggio Emilia	Modena	I
Evert	Vrije University Amsterdam	Amsterdam	NL
Jeremy Pitt	Imperial College London	London	UK
Jennifer Willies	Edinburgh Napier University	Edinburgh	UK
Franco Zambonelli	University of Modena and Reggio Emilia	Modena	I

Participants at the consultation held during the Video Sprint in Edinburgh (8 January 2016)

Alessandra Russo, Imperial College London, UK
Alois Ferscha, JKU, Linz, Austria
Antonio Buccarionne, FBK, Italy
Axel Haddad, ULB, Belgium
Benoit Baudry, IRISIA/INRIA, France
Callum Egan, Edinburgh Napier Uni, UK
Emma Hart, Edinburgh Napier Uni, UK
Evert Haasdijk, VUA, Netherlands
Giacomo Cabri, Uni Modena & Reggio Emilia, Italy
Gusz Eiben, VUA, Netherlands
Jane Hillston, Uni Edinburgh, UK
Jennifer Willies, Edinburgh Napier Uni, UK
Jeremy Pitt, Imperial College London, UK
Laura Nenzi, IMT Lucca, Italy
Michele Loreti, Uni Florence & IMT Lucca, Italy
Nadia Radzman, John Innes Centre, UK
Nicolas Markey, CNRS/ENS Cachan, France
Stuart Anderson, Uni Edinburgh, UK
Thomas Schmickl, Uni Graz, Austria
Tyler McCleery, John Innes Centre, UK
Vasilios Andrikopoulos, Uni Stuttgart, Germany
Yehia Abd Alrahman, IMT Lucca, Italy

Structure of the document

This manifesto is structured in four chapters.
Chapter 1 shortly introduces collective adaptive systems, providing a definition and a vision.
Chapter 2 presents the challenges in the fundamentals of collective adaptive systems.
Chapter 3 proposes some challenges in engineering collective adaptive systems, which is, in our opinion, the next step in the research field.
Chapter 4 concludes the manifesto, summarizing the key concepts and research directions.

Chapter 1 definition and vision of CAS

Collective Adaptive Systems (CASs) consist of diverse heterogeneous entities (i.e., computers, services, devices, sensors, humans, networks, robots, etc.) that are autonomous but have to cooperate with each other to accomplish collective tasks [1]. As each entity is *autonomous*, it is characterized by its own behaviour. To enable collaboration, it can also expose functionalities to the outside world. As such, while entities preserve individuality, they still can form collectives for collaboration. As a rule, the environment in which such a system operates, continuously changes, starting from the changes of the context in which entities live, to the availability of new entities (or their exit from the system), to changes in the system requirements and preferences. As a rule, dynamic changes might affect the operation of a CAS and have to be properly handled at run time by adapting system configuration and behaviour of constituent entities. This *continuous adaptation* becomes the key feature of CAS when it comes to operating in continuously changing environment. Concepts that are close to those introduced above, and that characterize CAS, have been studied in various domains like, for example, *Swarm Intelligence*, where actors are essentially homogeneous and are able to adapt their behaviour considering only local knowledge [2, 3], or *Autonomic computing*, where the actors types are typically limited and the adaptation is guided by predefined policies with the objective to optimize the system rather than evolve it [4, 5, 6], or *Service-based systems* where services are designed independently by different service providers and are composed to achieve a predefined goal (i.e., user tasks [7] or business goals [8]), or *Multi-agent based systems* where activities of different actors are regulated by certain collectively defined rules (norms) [9, 10]. Most of the results obtained in these domains are tailored to solve a specific problem using a specific language or model and lack of generality.

At the same time, they tackle only some of the individual challenges, for example, while in these different approaches the focus of research on adaptation has not been put on adaptive behaviour of the composite system yet, but is limited to the definition of adaptation solutions for individual entities, we should move from individual-based applications to collective systems proposing also adaptation techniques that support adaptation of collectives. This will be achieved by defining a new model that is highly flexible and can be specialized to fulfil different tasks in different ways. At the same time, they will introduce features for the collaboration and coordination among entities as this is an essential prerequisite for building adaptive collective systems.

As we said above, each entity of a CAS exposes a set of functionalities to the outside world and through its life cycle can both work independently or in collaboration with other entities. We can characterize a CAS with two main core concepts: *cells* and *ensembles*. Cells are basic building blocks representing the different functionalities provided by the entities and ensembles are collections of cells collaborating together to accomplish certain goal in a given execution environment. Each cell, during its life, can change its behaviour (i.e., cell specialization) such that it is able to achieve a given goal in collaboration with other cells in an ensemble. At the same time, an ensemble must be configured (i.e., ensemble configuration) in such a way that its goal and the goals of the cells collaborating in it are fulfilled. A system like this can be considered as a '*system of systems*' and is characterized by the following aspects:

- *Autonomicity and Large-scale*. It is built as a set of multiple and autonomous entities (often very large numbers) where each of them has its own individual goals and can take actions.

- *Collectiveness*. While entities are autonomous they may need to form collectives in order to achieve their goals jointly. A primitive collective is a pair of entities using peer-to-peer communication. In general, collectives may be arbitrarily complex and may involve complex moderation and collective decision making (i.e., cooperation, competition, negotiation, etc..).
- *Dynamic environment*. The dynamicity of the environment is mainly determined by 1) the autonomicity of the entities (we, as an entity, cannot control the non-deterministic behaviour of the surrounding entities), 2) openness of the environment (entities can dynamically enter and exit the system) 3) factors that are external to the system (e.g., some failures that are not caused by any entity in the system). These three factors result in the challenges that we call dynamic environment, dynamic partners and entity customization;
- *Adaptability.* To be robust, each entity must be able to dynamically adapt its behaviour and its goals to changes in the environment but also to collaborative interactions with other entities. At the same time the adaptation must not be controlled centrally and imposed by the system but must be administered in a decentralised fashion among the entities.
- *Hierarchy.* Smaller collectives may become parts of larger collectives, thus making up a hierarchy of collectives that have different scales in time and space.

[1] S. Kernbach, T. Schmickl, and J. Timmis. Collective adaptive systems: Challenges beyond evolvability. ACM Computing Research Repository (CoRR), August 2011.

[2] C. Pinciroli et al. Argos: A modular, multi-engine simulator for heterogeneous swarm robotics. In IROS, pages 5027{5034, 2011.

[3] P. Levi and S. Kernbach. Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evo-lution, volume 7. Springer Verlag, 2010.

[4] D. B. Abeywickrama, N. Bicocchi, and F. Zambonelli. Sota: Towards a general model for self-adaptive systems. In WETICE, pages 48{53, 2012.

[5] R. Bruni, A. Corradini, F. Gadducci, A. Lluch-Lafuente, and A. Vandin. A conceptual framework for adaptation. In FASE, pages 240-254, 2012.

[6] P. Lewis, M. Platzner, and X. Yao. An outlook for self-awareness in computing systems. Awareness Magazine, 2012.

[7] R. Kazhamiakin, M. Paolucci, M. Pistore, and H. Raik. Modelling and automated composition of user-centric services. In OTM Conferences (1), pages 291-308, 2010.

[8] A. Marconi, M. Pistore, and P. Traverso. Automated composition of web services: the astro approach. IEEE Data Eng. Bull., 31(3):23-26, 2008.

[9] G. Cabri, M. Puviani, and F. Zambonelli. Towards a taxonomy of adaptive agent-based collaboration patterns for autonomic service ensembles. In CTS, pages 508-515, 2011.

[10] E. Lavinal, T. Desprats, and Y. Raynaud. A generic multi-agent conceptual framework towards self-management. In NOMS, pages 394-403, 2006.

Chapter 2 Challenges in fundamentals of CAS

We have collected a lot of material, in particular challenges related to the fundamentals of collective adaptive systems.

In this chapter of the manifesto we will present a structured report of the challenges. We have divided the challenges into two macro categories: the former relates to the features of CAS, the latter to transversal aspects of CAS.

The main features of CAS we have identified are:

- Self-organization
- Autonomy
- Adaptation
- Collectiveness

Section 2.1 is devoted to explain the challenges related to these four features of CAS.
The identified transversal aspects of CAS are:

- Modelling of CAS
- Governance of CAS

The challenges related to these two aspects are discussed in Section 2.2.

Section 2.1 Challenging in features of CAS

Section 2.1.1 Self-Organisation

Very Large Scale Collectives

Collective Adaptive System (CAS) as a broad term refers to today's observable indications of a deployment of Information and Communication Technologies (ICT) at a very large scale (e.g. $3.5*10^9$ personal computers, $6.2*10^9$ smart phones, $1.1*10^9$ cars on planet). Taking such a massive ICT deployment, it is not just considered possible, but already a reality that these are programmed to operate cooperatively as very large scale collectives. As for example, smartphone apps have demonstrated globe-spanning cooperative sensing applications, liquid democracy applications, logistics and traffic management applications, energy management applications, etc. One essential aspect of such globe-spanning ICT collectives is that they often exhibit properties typical observed in complex systems, like (i) spontaneous, dynamic network configuration, with (ii) individual nodes acting in parallel, (iii) constantly acting and reacting to what the other agents are doing, and (iv) where the control tends to be highly dispersed and decentralized. If there is to be any coherent behaviour observed on the level of the collective (macro-level), it (v) it usually arises from competition and cooperation among the individual elements (micro-level). The overall behaviour of the collective (macro-behaviour) is the result of (vi) a huge number of decisions made every moment by many individual entities (micro-interactions), possibly leading to (vii) the emergence of unexpected phenomena.

Self-Organizing and Self-Managing Systems

A critical challenge for CAS is to design and implement collectives able to operate in a more or less autonomous way, with little or no human interaction. In this endeavour, self-adaptation and self-organisation have emerged as two interrelated facets of CASs. Self-adaptive collectives in a top down strategy evaluate their own global behaviour and change it when the evaluation indicates that they are not accomplishing what they were intended to do, or when better function or performance is possible. Here a challenge is to identify how to change the behaviours of the individuals so as to achieve the desired collective improvement. In self-organizing collectives, individuals interact locally, typically according to a small set simple rules. Achieving a desirable collective behaviour is attempted with a bottom up strategy, in that local interaction policies are adopted so as to steer global behaviour.

REF PAPER (Dimarzo et al. 2005)

Self-organisation is the process enabling a system to change its organisation in case of environmental changes without explicit external command. We can distinguish between: weak self-organisation where from an internal point of view, there is re-organisation under an internal central control or planning; and strong emergence where there is re-organisation with no explicit central control, either internal or external.

Superorganisms

Building upon the mechanisms of self-adaptation and self-management, we may ask the questions on the potentials and opportunities of turning massively deployed systems to a globe spanning superorganism of socially interactive elements. While the individual elements are of heterogeneous provenance and typically act autonomously, we can assume that they can (and will) self-configure into large scale cooperative collectives, either with or without human governance. We may not assume a common objective or central control, but rather volatile network topologies, co-dependency and internal competition, non-linear and non-continuous dynamics, and sub-ideal, failure prone operation. The recent literature has referred to these emerging massive collectives as "superorganisms", since they exhibit properties of a living organism (like e.g. 'collective intelligence') on its own. In order to properly exploit such super-organisms, we need to develop a deeper scientific understanding of the foundational principles by which they operate.

Designing Emergent Behaviour / Properties

An emergent behaviour or property appears (is observable) at the level of the system - when it is seen as a whole (super-organism or a full collective) - while none of the entities/components constituting the system exhibit that property. From a programming point of view, a property emerges at the global level, but it is not explicitly coded into the individual components.

Emergent properties in nature[1]

In philosophy an emergent property of a system is a property of the system as a whole that is not a property of any component of that system. Emergent structures are patterns that emerge via collective actions of many individual entities. Weak emergence is a type of emergence in which the emergent property can be simulated by a computer. For strong emergence instead, the emergent property cannot be provided in this manner. An example of "weak emergent property" is the shortest path built by ants when for foraging, while none of the ants is actively

[1] https://en.wikipedia.org/wiki/Emergence

trying to build a shortest path. An example of strong emergent property is provided by the progressive and collective buildings of snow-flakes and their distinctive 6 branches patterns, or the creation of sand dunes and wind-shaped patterns from the collective movement of sand grains.

Emergent properties in engineered systems
Following these principles observed in nature, (REF TFG Meeting Lisbon report - 2006) defined an emergent phenomenon as a functionality, structure/organisation, characteristics or property of a system not explicitly coded in the individual local components, visible by an observer at the macro-level but not necessarily at the micro-level. Weak emergent phenomena are those phenomena we can define as the result of a (possibly complex) operational function of the components. Strong emergent phenomena are those phenomena we cannot define as the result of such an operational function.

Challenge: **Designing / engineering the individual components so that the system as a collective exhibits the expected emergent behaviour or property.**
Self-organising mechanisms/patterns are one of the frameworks for designing self-organising systems proposed in the recent years. Indeed, self-organisation is achieved by the use of *self-organising mechanisms*, i.e., rules that individual components employ to coordinate their behaviour, usually following information gathered from their local environment. In order to make the self-organizing mechanisms applicable more systematically, (REF: Fernandez et al.) proposed a catalogue of self-organising, design patterns, detailing the problem they solve and the solution they propose, discussing as well their relationships and their respective boundaries. This catalogue discusses some of the bio-inspired self-organizing mechanisms available from the literature, such as gradient, gossip, or digital pheromone. These mechanisms provide autonomy, decentralized coordination, and pro-activity, and lets the entities employing those mechanisms locally interact among themselves or with their environment.

Challenge: Questions and challenges arising from these patterns relate to the **universality and completeness of mechanisms and patterns**, in particular: Is it possible to find a flexible and universal mechanism of adaptation that can support all these diverse mechanisms? From all possible mechanisms that we can borrow from nature (from biology, human behaviour, etc.) is it possible to identify a minimum set of mechanisms sufficient to design and build self-organisation and emergent properties in a vast variety of systems?

Verifying Emergent Behaviour / Properties
Challenge: Verifying from the design (or engineering) of the individual components that the expected emergent behaviour or property will eventually actually happen or, if the property must be avoided, that it will actually be prevented.

The properties that are of interest to both system developers and system users, are global properties relating to emergent behaviour and properties of the system. The implementation is viewed in terms of the composition of a number of local behaviours, verification is more easily tackled at the level of local components and their individual behaviour.

Challenge: A challenge for the verification is to develop ways in which assessment of global/emergent properties may be built upon verification of local properties of individual or small collections of components. **This encompasses offline (design time), online (run-time) verification and involves new types of properties that are spatio-temporal in nature. There is then a need for formal languages capturing all these aspects, and a compositional approach to verification.**

Emergent properties have a spatio-temporal nature, since the system evolves with time and progresses physically in the environment in which it is grounded (e.g. spatial services spreading in geographic space). Specific attention need to be given to these properties particularly in relation to well identified self-organising design patterns that demonstrate those properties (e.g. spatial gradient), in particular to develop spatio-temporal model-checking. Spatio-temporal properties address both the evolution in time and the spatial properties of an element of the model. In each possible world there is a different valuation of atomic propositions, inducing a different snapshot of the spatial situation which evolves over time. This is an area of on-going work and there are a number of open issues for this line of research: what are the necessary and sufficient spatio-temporal operators to express the spatio-temporal properties relevant to the self-organising patterns used to design and implement CAS? What are the operators that address performance aspects (e.g. to express the probability that 90% of the people in an emergency exit case have been reached within a certain time-bound) or collective aspects such as that a set of points satisfies a certain property?

The above challenges clearly link to other challenges on local-to-global, verification and prediction issues: from design of local components we engineer/verify emergent properties at the global level.

Control

Control and trade-off between autonomy and controllability (by external operators)

A typical contradiction in self-organising systems lies in the trade-off between full autonomy and the industry-level usage of those systems requiring some level of control. A typical question is: how can a system administrator stop/change the goal/retrieve a swarm of robots/drones once they are deployed in the wild? The more external control is allowed, the less the system is actually self-organising. Should self-organising be circumscribed to a specific functionality, while control dedicated to other tasks (meta-level?).

Section 2.1.2 Autonomy

In CAS, *autonomy* concerns at least two levels: the system (macro) level and the component (micro) level. Since different sorts of autonomy (such as execution vs. motivational autonomy, or autonomicity) are possible in general, components of a CAS can in principle feature diverse sorts of autonomy. So, while a CAS as a whole typically features *execution autonomy* (as the ability to autonomously pursue some given goals) without *motivational autonomy* (as the ability to choose its own goals), components of a CAS may generally exhibit any sorts of autonomy. For instance, socio-technical systems like social networks may be seen as CAS whose human components have full motivational autonomy; at the same time, many biological CAS does not require much more than a limited execution autonomy to its components.

When observed from the viewpoint of the components, execution autonomy requires first of all the *situatedness* of the components, as well as the ability to select the most suitable course of actions to achieve the individual goal. Whenever physical systems are involved, autonomy is also strongly related to the problem of *energy*, or, more generally, to the problem of accessing all the resources required to sustain the component's activities. Effectiveness and efficiency of the activity of CAS components largely depend on their ability to deal with the available resources: in the case of energy, for instance, on the ability to obtain (energy supply) and to preserve (energy storage) the energy necessary for sustaining the life and activity of the components. When observed from the viewpoint of the overall CAS, and of the environment where it works, the same problem translates onto the issue of *sustainability*, and may involve not just a CAS and its components, but also any other system sharing the same environment and the same resources.

A special sort of resource, requiring to be mentioned separately, is *knowledge*. Many of the foreseeable scenarios for CAS involve knowledge-intensive environments (KIE), where both autonomous tasks and adaptation may require the ability of handle huge amounts of data and information. The ability of making sense of available knowledge is then critical for autonomy in KIE, and requires some cognitive skills (such as the ability to make correct inferences over raw data–for instance, as in the case of logic engines), or even some cognitive architecture (such as the capability to reason over articulated scenarios–for instance, as in the case of intelligent agents).

Interference with other systems sharing the same environment may affect autonomy of a CAS and of its components. Interference may occur indirectly – for instance, in terms of limited availability of shared resources – or directly–for instance, by attacking CAS components in order to maliciously change their behaviour. This is why *security,* and *safety* as well, are relevant issues for autonomy in CAS.

With great autonomy comes great responsibility–and liability, and the like. While automatic systems can easily trace back responsibility of their actions to designers, builders, deployers, maintainers, and users, the conceptual and technical distance between the engineering of a CAS (and of its components) and its running in the execution environment is too huge, due to autonomy, to make the same operation easy again. This is why a ever-growing effort is currently ongoing, as a joint effort by the communities of experts of law, technologies, politics, and social sciences, in order to understand the possible dimensions of responsibility and liability of autonomous systems–including CAS.

Next we address some specific aspects related to autonomy.

<u>Independence</u>

One of the most challenging aspects concerning autonomy of distributed technical systems is **energy supply and energy-storage**. Concerning this aspect, engineering is still several orders of magnitude behind the efficiency exhibited by natural CAS (organisms). This shortcoming of the current state-of-the-art is also linked to **sustainability** and is also a **maintenance** issue (cost). In addition to that, autonomy of distributed and adaptive systems is a **security** issue as well as a **liability** issue, requiring sophisticated governance and legislation, as well as a high level of informedness of the end-users and other stakeholders (**transparency**). In addition to battery supply there is also an issue of **longevity** of technical devices: When it comes to long-term running physically embodied CAS (in contrast to pure-software CAS), one would first think that the collectiveness allows to replace broken units in the system and thus extend the

collective lifetime of such CAS. However, the **individual lifetime** of the components is a critical factor in the feasibility of such systems, as the supply with material can quickly get a limiting factor, associated also with cost and sustainability issues. This holds especially for autonomous systems, as they are inherently not always under control/monitoring by human supervisors.

Human-in-the-loop

The **human-in-the-loop** approach seems to mitigate some of the risks involved with autonomous systems as liability, **responsibility** and control might be shifted to this human agent. However, there is an issue of **fairness** in this approach, as this human might be already misinformed by filtered, algorithmically processed data/information. Thus, ethical aspects pose a challenge here, as well as **usability** aspects of the system (user interface, visualization and filtering of relevant information, etc.). In addition to that, human cognitive information processing, **ethics**, **plasticity of behaviour** and other **social factors** and knowledge can help a human-in-the-loop system to perform beneficial for human society better than a system without human in the loop.

Biology

Besides human society, CAS are embedded in a whole **ecology**, thus they offer the chance to act beneficial concerning their embedding ecosystems and **sustainability**. However, ecosystems represent CAS on their own, thus they pose the challenge of integration of artificial CAS into them. Principally, **artificial CAS should not go into conflict with natural CAS**, except there is a clearly understood and **societally agreed** benefit for human society, e.g. when fighting diseases spreading or antibiotics resistance.
Besides this integration of artificial CAS into living ones, there is also the approach of integrating existing natural CAS (animals, plants) into artificial CAS or even to build novel artificial CAS from living CAS building blocks (**bio-hybrid approach**): Those approaches pose a plethora of challenges, as they require not only the **understanding of the fundamental mechanisms** in the living CAS and in the artificial CAS, but also a deep understanding of phenomena that can **emerge** at the interface between those layers due to **self-organization** and **complexity**. **Modelling** self-organizing adaptive distributed heterogeneous systems on the **microscopic** and on the **macroscopic** level is a key strategy to achieve the fundamental understanding required to build reliable, predictable, secure and trustworthy bio-hybrid CAS.

Reliability/robustness

To extend the **robustness** and **reliability** of CAS by exploiting **adaptive** mechanisms **homeostasis** and **self-regulation** are often mentioned as a design target. Such self-stabilizing systems can be achieved by an outside designer who applies an outside model (control theory) to construct the required feedback loops to **self-regulate.** This requires a-priori knowledge of the system structure (including feedback loops and delays in those feedbacks) as well as knowledge on the level and characteristics of noise in the embedding environment. To prevent such a demand for a-priori knowledge, there is the approach of **self-modelling**, which is still at a very early stage is allowing the system to model itself and frequently re-validate this self-model, occasionally detecting mismatch of model-derived predictions and real-world measurements indicating an unexpected change in the system itself. By applying methods of **automated model-refinement** or just re-iterating **self-calibrating parameters**, the system can detect and thus also react to such changes, ultimately allowing it to gain **self-regulation** again.

Section 2.1.3 Adaptation

Models for adaptation

- CAS should be **adaptable by design**; this means that each entity in the system must be able to adapt its behaviour taking into account the current context/situation.
- The **model should be flexible and extensible**, fusing a priori and learn knowledge. The local knowledge of an entity should be extended during its life thanks to collaboration with other entities and learning about the current context of the adaptation.
- The **model should consider the heterogeneity and diversity** of the entities and the same time the specific roles that they play in the collective.

Techniques for adaptation: **To find a flexible and universal mechanism for adaptation** that unifies different mechanism and provide a way to compose them in adaptation strategies.

- reflection, learning, reasoning and inference, planning, evolutionary

Types of Adaptations: new coordination mechanisms, new communication protocols, changes of internal behaviours, adaptation of adaptation mechanisms and strategies, interpretation of messages.

Techniques for collective adaptation

- co-adaptation is needed every time that an adaptation at the level of a single entity in the CAS impact other entities. The overall adaptation resolution can be done using different mechanisms like **collective learning** taking into account different assumptions (i.e., selfish vs cooperative, partial observability, coordination that respects stabilization goals).
- co-evolution is a form of long-term adaptation
- collective learning (sharing knowledge, observability, joint decision making)

Collective Adaptation Needs

- **Adaptation to changed requirements/goals, changing environment, user preferences, new or leaving resources, changing topologies.**
- Challenge w.r.t. adaptation techniques / models: open-ended evolution, innovation in adaptation, adapting to / learning in the presence of unforeseen changes...

- Each entity can have multiple goals simultaneously and the adaptation need could be also multi-objective.
- **Characteristics of goals: heterogeneous, multi-objective, dynamic (the goal can be revised),** measurement of progress towards the goal is partially observable: each entity knows only part of the collective goal from its own perspective (no need to express/specify collective goals but satisfy a collective goal means to be single entity goals satisfied)

Section 2.1.4 Challenges in collectiveness

Scale

Within a distributed system, scale is not a one-dimensional facet but a multi-dimensional one: *size*, *space*, *time* and *organisational* scale should be considered, at least. Probably the most intuitive dimensions are size and space, often collectively branded as *scalability*: our perception of a distributed system, in fact, usually assumes a huge amount of entities somehow distributed in a geographical / virtual space. Probably less trivial to recognise are the time and organisational dimensions: distributed systems may in fact be structured (or, at least, interpreted) according to a *hierarchy* of domains / layers / views operating on a different scale w.r.t. the organisational structure---e.g., as happens with living organisms > organs > tissues > cells. More, these views may operate on different *time scales*---e.g., as happens within an ecosystem > population > individuals.

Scalability, in particular, refers to the ability of a system to handle a growing amount of distributed entities (e.g., devices, users, software agents) in terms of performance and computational complexity. Being able to do so is a strict requirement for CAS, as word "collective" points out. Challenges in scalability are both in software and hardware, in particular: experimenting with a collective of -- potentially, costly -- heterogeneous devices with different operating systems, software, demanding maintenance, etc., keeping reasonable performance (almost) independently of the scale of the system, relying on decentralized algorithms / mechanisms / knowledge only.

Heterogeneity

Heterogeneity can be interpreted, at least, along two dimensions: the *nature* ("structure") of entities participating the system (devices / software agents / human users), and their *goals* (purpose, attitude, etc.).

In nowadays systems we have many different devices work together. For example, we have powerful workstations, laptops and tablets, but we also have small devices like mobile phones, watches and wearables. If we want them all to collaborate we have to deal with / take advantage of their *heterogeneous* nature (different computational model, different communication protocols, etc.).

Also non "heterogeneous-in-nature" entities may be still heterogeneous in their goal / purpose / functionality, which then have to be coordinated into a system-wide meaningful behaviour.

Security, privacy, trust

When we have so many different devices working together in a distributed way, it is extra important that the system is *secure*. When you can break into one device, not all connected devices must be able to get in. (challenge: Mitigating security threats in open systems Identification of building block components that can be predictable)

Dependability and resilience

A collective system being dependable and resilient means for it to be *robust*, *reliable*, *maintainable*, *safe*, *incorruptible*, etc. How to achieve this and how to integrate such aspects raises many challenges: ideally, we want a collective system that can self-maintain forever despite (thus, adapting) ever-changing circumstances, possibly on different time scales. On the other hand, we want to prevent the system to go out of hand [130,144], thus recognise and accommodate individual failure at the collective level of behaviour.

Coordination and interaction

Being CAS collectives, the *interaction dimension* of computation is reasonably expected to play a fundamental role in both their modelling and engineering. Managing interactions -- as the comprehensive term including *communication*, *collaboration*, *competition*, *interference*, *compositions*, etc., basically any action leading to a *dependency* between different activities -- is the foremost responsibility of coordination models and technologies. Challenges in this sense regards: *interoperability* between heterogeneous entities (hardware, software, human) with the goal of meaningful exchange of knowledge, achieving (potentially) global communication supported by *local-only* interactions (either constrained by software for scalability purpose, or by hardware as a physical world constraint), taming complexity of *multi-scale* interactions both within different layers of the same system and within different coordinated systems, effectively and efficiently supporting establishment and change of structural and volatile relationships (e.g. *composition*), possibly changing over time as well as according to contextual (environmental) conditions, tolerating *uncertainty* both in the knowledge exchanged, in the knowledge needed to meaningfully communicate as well as in the outcome of interaction processes (knowledge produced), designing sufficiently *expressive* yet manageable -- from the complexity point of view -- interaction / coordination mechanisms supporting a reasonably wide range of distributed algorithms.

Section 2.2 Transversal challenges

In this section we address two "transversal" sets of challenged: one related to modelling CAS and the other related to governing CAS.

Section 2.2.1 Modelling CAS

Lack of theory and foundation

Basic research question here is how to create at design-time and maintain at run-time a shared world model for a dynamic CAS in order to enable collaboration, possibly in unanticipated situations. Available approaches include:

- formal foundations and modelling techniques for concurrent systems that deal with problems such as enabling and limiting concurrency, access to shared resources, avoidance of anomalies, inter-process communication, performance estimation;
- models and formal languages that incorporate general features of CAS and are open to the integration of different specialization aspects in a modular way;
- approaches originating from different communities, ranging from research on concurrent systems to adaptive/multi-agent systems, also addressing modelling of bio-inspired mechanisms;
- use behavioural model based on graphs (a la process algebra) or mathematical models based on PDE, and establish correspondence between them;
- different communication models (varying from local to global reachability) and collaboration models (varying from collaborative to competitive).
- interconnections (possibly dynamic) between the entities composing CAS via higher-level, declarative (name- or property-based) specification or via spontaneous emergent coordination patterns;
- modelling mechanisms that more easily allow to fill the gap between emergent global properties and single-component behaviour;
- models capturing collections (or aggregate) of components as first-class abstraction.

Formal languages and expressiveness

A key modelling issue to support a sound engineering process for CASs is to develop the right formal computational model or language that can ground the construction of provably correct and increasingly complex CASs. This needs to start from the identification of interaction primitives forming a universal set of low-level mechanisms of collective adaptation. On top of it, expressive and reusable building blocks should be developed that can be proven resilient (robust to changes and fault, self-adaptive, and self-organising); typically, such building blocks will support known patterns of collective adaptation (collective spreading, aggregation, and so on). Ideally, a proper theory of composability and substitutability can be introduced that can (i) allow such building blocks to be increasingly composed to form libraries and systems enjoying the same resiliency properties of building blocks; and (ii) allow to delay the task of finding efficient implementations of building blocks, possibly replacing (either statically or dynamically) the existing one.

Bridging the gap between individual/micro/local (bottom up) and collective/macro/global (top down) models

Collective adaptive systems are complex systems, and such a main theme of research lies in the ability of properly filling the gap between micro-level specification (namely, the behaviour of individuals) and macro-level specification (namely, the resulting behaviour of the collection, and its properties). This problem can be addressed in different ways, each providing different challenges to system modelling and engineering. First of all, one can try to provide some support for engineering emergence, namely, identifying those conditions on individuals' behaviour that can lead to globally functionally correct behaviours; this approaches is most suitable when there's some lack of knowledge about what the intend global behaviour should be. On the other side of the spectrum, a viable approach is to synthesis individual behaviour directly out of the intended global behaviour, e.g., via direct programming the self-organising system at the aggregate level. In between, practically an intermediate approach can be suitable that models local as well as global levels, balancing the design of both of them. More realistically, complex CASs may need to resort to multiscale models, where the concepts of locality and globality blur into multiple levels.

Verification and validation

Various techniques can be used to verify and validate a CAS design:

- techniques for the analysis of concurrent systems that typically cluster equivalent systems to reason on their stochastic properties, suitably enhanced to deal with large-scale systems, and dynamic architectures;
- compositional reasoning to support scalable analysis techniques that address qualitative and quantitative aspects of systems;
- techniques for providing the basis for the development and runtime control of CAS while providing assurance of the desired properties;
- techniques to simplify verification of properties by abstracting away from a single-device behaviour, focussing instead on how ensembles/aggregates work together in achieving collective tasks;
- service based approaches for dealing with diversity, hence with legacy system (only interfaces are specified), models of humans, animal, robots (all seen as receptors and activators)
- assigning a role to knowledge models to set up learning and feedback loops

Additionally, when dealing with CAS one has to consider additional mechanisms to deal with:

- lack of predictability of system behaviour, caused either by the presence of noise or for the difficulty in observing all the variables of interest;
- frequently, system monitoring and analysis has to be done on-the-fly on a large-scale open system ();
- analysis with the embedded aid of simulation might be needed to properly connect with the real world.

Integrating diversity or extending modelling approaches (e.g. agent-based models) and theories integration (e.g. game theory), dealing with legacy

CAS can be of different nature: they can be natural systems at different levels (physics, chemistry, biology, ecology), legacy computational systems, robotic swarms, cyber-physical systems in which humans and digital systems coexist, and so on. A challenge for research on the foundations of CASs is hence to deal with such diversity, migrating legacy systems and devices into working CASs, finding approaches suitable integrating into computational systems animals and robots, ensuring that controlled emergence can be achieved in spite of individuals belonging to a heterogeneous set, and make diversity of behaviour a desired feature of emergent systems.

Section 2.2.2 Governance of CAS

"Governance" concerns "the processes of interaction and decision-making among the actors involved in a collective problem that lead to the creation, reinforcement, or reproduction of social norms and institutions."[2]

Thus for a CAS *governance* involves bringing together many, **potentially conflicting**, **stakeholder collectives** in order to **resolve conflict** and develop the social institution around the CAS that regulates its operation and ensures it satisfies the needs of the participants as far as this can be achieved. The recognition of the need for structures that reconcile conflicting goals is well developed in the economic literature. In particular Ostrom's notion of **"Polycentric Governance"**[3] recognizes the need for inclusive approaches take account of relevant stakeholders despite potentially irreconcilable differences. A particular challenge arises from the *hybrid* nature of CASs that involve diverse human collectives together with assemblages of hardware, software and potentially other biological systems. This challenge is in determining how to change the systems to ensure particular goals of governance are achieved. For example, in a system that takes decisions on **the allocation of resources**, we may want to ensure that these are allocated "transparently" in the sense that it is clear that no individual or collective is being deprived resources they can justify they should receive. But, for many CASs involving resources there may be groups that take a strong perspective on privacy of individuals and groups and so this may be in conflict with other requirements. Resolving such issues will require new arrangements such as independent auditors or other aggregated ways of surfacing resource allocation without violating the **privacy demands** of some stakeholder collectives.

Policies

Policies are one tool to reinforce and reproduce **social norms**. Often they are used to restrain the scope of action of particular actors or collectives in a system. For example, an electricity supply company might have a legal right to terminate power supply to a household on the basis of non-payment but they may have a policy not to terminate the power supply if there are children living in the house. Policies might also include rules to forbid data linkage across datasets that might identify individuals or vulnerable groups through linkage.

Sustainability of CAS

On the one hand, CAS can provide **tools to improve the sustainability of existing large infrastructure** such as smart grids, smart cities and smart transportation systems. CAS can monitor and act to **optimize energy consumption**. The aim can be the creation of CAS that would develop services and applications making use of information generated by energy consumers or captured from sensors (e.g. smart meters, smart plugs, connected devices and the internet of things). These CAS could **empower consumers** and stimulate collaboration to enable full participation in more efficient energy use. (Franco Zambonelli, Engineering self-organizing urban superorganisms, Engineering Applications of Artificial Intelligence, Volume 41, May 2015, Pages 325-332). Underlined in the Pillar VII: ICT-enabled benefits for EU society, the Digital Agenda focuses on ICTs capability to reduce energy consumption.

On the other hand, **sustainability issues** have to be taken into account in the design of CAS in terms of **energy and material consumption**. Devices making the CAS should be energy efficient or **harvest their energy** from their close environment. The design of the components should avoid rapid obsolescence and make a rational use of non-sustainable and **critical materials**. In the past decade,

---

[2] Hufty, Marc (2011). "Investigating Policy Processes: The Governance Analytical Framework (GAF). In: Wiesmann, U., Hurni, H., et al. editors. Research for Sustainable Development: Foundations, Experiences, and Perspectives.". Bern: Geographica Bernensia: 403–424.

[3] E. Ostrom, Beyond Markets and States: Polycentric Governance of Complex Economic Systems. The American Economic Review, 100(3) 641-672, 2010.

sporadic shortages of metals and metalloids crucial to ICT technologies have demonstrated the relative criticality of various materials.

Technological innovation for long term duration:

CASs will often persist over **long timespans** so there is the need for mechanisms to identify and incorporate new technologies into the CAS and adapt the surrounding technologies, **practices and norms** to take account of the technical change. For example, issues such as historical measurements, standards, and interpretive practice all may need to change as a new technology is adopted. Any CAS with long lifespan will need to take account of these aspects. Paul Edwards account of the social organisation around climate data[4] exemplify some of these aspects.

Innovative social practices

Innovation will inevitably involve the adoption of **new social practices** by collectives[5]. Some of these will arise from changes in the environment and some will be necessary to ensure the objectives of the CAS are maintained.

Observability, measurement, perspective, goal

A key aspect of the governance of a CAS is the extent to which it can be instrumented **to allow the observation** (some aspects will be unobservable), measurement, and the interpretation of data from different stakeholder perspectives. This concrete aspect of measurement and observation will often be heavily contested because it is the source of evidence to support different stakeholder perspectives in the governance of the CAS.

Incentives and decision-making

Incentives are a key aspect of the shaping of decision taking in CASs. The design of **incentive systems** is very challenging and there are many examples of the creation of "perverse incentives"[6] that fail to achieve a global objective through a system of rules based on local performance incentives for individual collectives in a CAS. Incentive design is a key aspect of the design of CASs.

[4] Paul N. Edwards. 2010. *A Vast Machine: Computer Models, Climate Data, and the Politics of Global Warming*. The MIT Press.

[5] Peter J. Denning and Robert P. Dunham. 2010. *The Innovator's Way: Essential Practices for Successful Innovation*. The MIT Press.

[6] Seddon, John. *Systems Thinking in the Public Sector: The Failure to Reform Regime-- and a Manifesto for a Better Way*. Axminster: Triarchy Press, 2008.

Chapter 3 Engineering challenges in CAS

In the previous chapter we have introduced and discussed many issues that have to be faced towards building solid foundations of a science of CAS, which includes for instance clarifying issues related to qualitative and quantitative modelling, harnessing self-organization and scale, understanding the trade-offs between autonomy and collectiveness.

However, emerging socio-technical systems bringing together a large-number of humans and/or ICT devices, and engaging them in a variety of purposeful activities, decentralized and without any well-defined locus of control, are de facto CAS. These systems can take many forms: systems composed of both human and ICT components, and involving different types of interactions, namely human-human, computer-mediated human-human, human-ICT ones (sensor, device, software agent, robot, etc.), and ICT-ICT ones. The scenarios in which we will see such systems at work will include smart cities, mobility management, smart homes, logistics, rescue and disaster management scenarios, just to mention a few.

Given that such systems will be so strongly intertwined with the everyday life of everyone, and likely to strongly affect it, we cannot limit ourselves to passively study and understand them, but must be ready to control and engineer their behaviour for the good of the society. In other words, CAS must become subject to the efforts of people and engineers, who have to intervene (e.g. through assistive mobile applications, transport scheduling systems, smart home control software, etc., and both at design time and at run-time) in order to attempt to influence the behaviour of both collectives and the individuals entities who comprise them, and direct them towards the desired behaviours and configurations.

Such efforts occur neither in a vacuum nor a greenfield site, and their success will be dependent on complex interactions between the deployed technologies and the already existing CAS. Moreover, when humans enter the picture as participants of CAS the impact becomes even more unpredictable and uncertain. In order that engineering efforts in CAS are both successful and acceptable to the humans who inhabit them, there is a need for new development approaches, which embrace uncertainty, unpredictability and interaction of behaviours, while simultaneously taking into account human preferences at different scales (i.e. individuals, collective, CAS). While engineers cannot always program or directly control the individual components (especially if humans), they may “shape” their interactions or the environment in which individuals live, in order to influence the way they behave and interact.

To ensure CAS sustainability in an efficient way it is necessary to conceive, design and deploy test environments that “mirror” the world possibly through several rounds of experiments to reliably deduce the impact of certain interventions on the global behaviour the socio-technical system exhibits. The long term vision is the development of a "general intervention framework", to be used at design-time, run-time or even both, which deals naturally with the complexities associated with socio-technical CAS. The framework will have to rely on methods for the automatic discovery of the drivers of behavioural change, on the automation of the formation of effective collaborations in heterogeneous CAS, on the opportunities for inspection and governance of the decisions that drive interventions.

Tackling these challenges will require contributions from engineering-related fields including statistical modelling, data mining, multi-agent modelling, and software engineering. However it is clear that the design, deployment and evaluation of socio-technical CAS are not purely engineering endeavours, and will rely on interdisciplinarity.

In the above context, we identify the key challenges discussed in the following.

Challenge 1: Shaping and Controlling Socio-technical CAS.

The main objective here is to develop an understanding of how the design and the operating principles for CAS composed purely of "technical" components can be successfully transferred to resolve corresponding problems in socio-technical CAS, where directly programmable influence is only partially possible [1].

Unlike in many engineering endeavours, the members of large-scale socio-technical CAS are often neither directly shapable by design nor directly controllable during the operational life of a CAS. On the one hand, people are not equivalent to programmable components. On the other hand, it may be the case that one has to deal with ICT components whose design and behaviour is not under the direct control of the engineer.

As a consequence, while engineers cannot neither shape nor control all the individuals (especially if humans), they may try to adopt different solutions to: **Shape** the collective adaptive behaviour of the system, that is assuring at the design level that the system will be able to serve its purpose at the global macro level, despite the impossibility of controlling each individual components and of accurately predicting the dynamics of the environment in which they situates; **Control** the dynamic collective behaviour of collective systems and its impact on the socio-physical environment in which it situates, to make it possible to enforce constraints (e.g., safety rules) in its behaviours and in its interactions with other systems, and possibly to tune its collective behaviour in order to dynamically meet emerging requirements.

Challenge 2: Transparency and Polycentric Governance of CAS Development

Since CAS impact upon many aspects of our lives, it will be desirable for the decision-making processes behind interventions (which may be automated) in CAS to be transparent to and understood by users, and incorporated into the system according to principles of good governance. Such integration will require systems to explain themselves, including their authority and decision-making rationale, in order to engender trust. Lack of transparency and trust are known problems in adaptive systems, and socio-technical CAS face this problem at a much higher level of complexity. Often, this lack of trust leads to users disengaging from adaptive systems, however with applications on the city-scale, for many this may not be an option.

One approach to this is to attempt a mapping between algorithms used to make or support decisions and a cognitive model, so that decisions can be intuitively understood and trusted. However, a challenge for cognitive modelling [2] is how to present vast amounts of data in a format that helps decision makers understand the reasons why one decision is made or supported rather than another, especially when humans are themselves required to make judgements under uncertainty about the efficacy of each decision.

It will be important to recognize however that whenever a given technology is offered to people, it likely simultaneously enables and constrains (shaping) their capabilities of behaving and interacting within the CAS. A key challenge is therefore how to inspect and intervene in a way which is both effective and acceptable to the inhabitants of the CAS. The overall challenge here is to incorporate both governance and self-explanation [3, 4] into CAS development. This will require the integration of dispute resolution mechanisms, social ergonomics and models of participation and engagement, acknowledging that “attention” is a limited resource [5].

The concept of transparency is particularly important in the presence of multiple centre of control for CAS. Indeed, future CAS will be governed by a variety of different stakeholders and will interact with a variety of other systems. For instance, consider the different aspects to be managed in smart cities, from traffic to energy to waste, aspects that are strictly intertwined yet are typically managed by different organizations. Such issue of “polycentric governance” for CAS raises the issue of understanding how it is possible to guarantee specific purposes in face of multiple and possibly conflicting goals within interacting systems and within a single system.

Challenge 3: Second-order emergent behaviour.

Most studies related to collective adaptive systems assume that the individual components are not aware of the global dynamic of the overall system, and that the resulting collective behaviour is an emergent process, resulting from the specific patterns of interactions of the components, and of their agnostic individual behaviours.

However, when individual components of a system are situation-aware (as it is the case of humans or of cognitive agents/robots), they can be made perceiving the global dynamics of the system and the resulting emergent purposeful schemes. Such awareness can in turn possibly influence their individual behaviour, affecting in turn the global behaviours. For instance, this is what happens in urban traffic due to the presence of real-time information systems, which enable drivers to reconsider their choice based on the global traffic patterns they become aware of.

Such phenomena of second-order (higher-order in general) emergence have not been so far investigated in the area of collective adaptive systems, and are of fundamental importance to properly control the behaviour of complex socio-technical CAS.

Challenge 4: Shaping with Unpredictable and Unknown Interventions

Socio-technical CAS are complex systems [6]. As participants engage in micro-level behaviours, actions and decision-making produces potentially unexpected macro-level outcomes. In many cases, the `levers' available to engineers, and their effects, will be unknown. Furthermore, CAS are highly uncertain, dynamic and heterogeneous: for example populations, attitudes and cultures change over time, and people have different and changing access to, perception of, and skills with technology. Failure, error, unforeseen emergent phenomena and misbehaviour (such as incivility) are all intrinsic parts of the system. Moreover, people innovate themselves, and in particular utilise generative technology in unexpected ways [7].

Due to unavoidable uncertainty in modelling CAS and limited and uncertain intervention opportunities and capabilities for engineers, simulation and model checking are often insufficient to ensure CAS sustainability. Thus, one promising approach is the use of live testing environments, running in a ``mirror world'' reflecting real-world data, behaviours and interactions while protecting real individuals, continuously providing feedbacks on CAS behaviour and absorbing patches, upgrades, intervention of engineers. The main challenge here is how to conceive, design and deploy these test environments. Several rounds of such experiments will need to be carried out in order to reliably deduce the impact of certain interventions on the global behaviour the socio-technical system exhibits. Despite the existence of structured approaches for model-building, developing good multi-agent models is still something of an art.

Challenge 5: A General Intervention Framework for Shaping CAS

Designing interventions necessary to govern CAS so as to realize system-wide goals, either manually or offline in advance, is extremely challenging, due to the complexities described in the challenges above. Our long term vision is that -- by attacking the above challenges, will lead to a "general intervention framework", that might be used at design-time (for shaping) as well as run-time (for controlling), which can help naturally dealing with the complexities associated with governing socio-technical CAS. This will require:

- An ontology for the specific “shaping mechanisms” which instantiate the general framework.
- Methods for the automatic discovery of the drivers of behavioural change, taking advantage of the wealth of data available within such systems, to facilitate the design of such interventions.
- The automation of the formation of effective collaborations in heterogeneous CAS, consisting of both human and engineered components. Heterogeneous contacts can be more creative and effective than homogeneous ones if the right people are matched to the right task, which is a non-trivial problem [8].
- The integration of those aspects of human behaviour and psychology which distinguish socio-technical CAS from “purely technical” CAS.
- Opportunities for inspection (supported by system self-explanation) and governance of the decisions (which may be automated) that drive interventions.
- Innovative tools that enable to validate the behaviour of CAS, e.g., in the form of live testing environments or “mirror worlds”.
- Mechanisms by which humans can be included in adaptation loops, and principled methods for deciding the degree to which decisions are made by humans or machines, given the trade-off between the quality, trustworthiness and efficiency of decisions.
- A rigorous methodology for on-the-fly creation of and experimentation in “mirror CAS”.

Tackling these challenges effectively will require contributions from engineering-related fields including statistical modelling, data mining, agent-based modelling, and software engineering. However, it is clear that the design, deployment and evaluation of socio-technical CAS are not purely engineering endeavours, and will rely on interdisciplinarity. Psychology, sociology, legal anthropology and business and stakeholder analysis will all make essential contributions to the creation of principled approaches for the development and shaping of socio-technical CAS.

[Refs for Chapter 3]

[1] Miorandi, D. & Maggi, L.: “Programming” social collective intelligence. Technology & Society Magazine 33(3), 55–61 (2014)

[2] Farrell, S. & Lewandowsky, S. (2011). Computational Modeling in Cognition: Principles and Practice. Sage.

[3] "Self-explanation in Adaptive Systems based on Runtime Goal-based Models", Nelly Bencomo, Kris welsh, Pete Sawyer, Jon Whittle, Transactions on Computational Collective Intelligence, 2014

[4] "Requirements-Aware Systems A research agenda for RE for self-adaptive systems", Pete Sawyer, Nelly Bencomo, Jon Whittle, Emmanuel Letier, Anthony Finkelstein, International Requirements Engineering Conference, RE 2010, Sydney, Australia

[5] Ferscha, A., Paradiso, J., Whitaker, R.: Attention management in pervasive computing. IEEE Pervasive Computing 13(1), 19–21 (2014)

[6] Gell-Mann, M.: What is complexity? Complexity 1(1), 16–19 (1995)

[7] Zittrain, J.: The Future of the Internet – And How to Stop It. New Haven, CT: Yale University Press (2008)

[8] Geiger, D. & Schader, M. "Personalized task recommendation in crowdsourcing information systems - Current state of the art." Decision Support Systems 65: 3-16 (2014

Chapter 4 A Synthesis: Governing Collective Adaptive Systems

To conclude this document, we now summarize in a concise way the key issues the key concepts behind CAS, and what we consider the most urgent and potentially highly impactful research directions.

As stated earlier in the document, we are assisting to a massive diffusion of networked ICT devices increasingly entangled with our physical and social world. This is making our everyday life and economic activities increasingly – if not fully – dependent on the functionalities of large-scale distributed software-intensive (i.e., heavily relying on software) systems.

Our streets will be soon populated by myriads of self-driving cars, interacting with each other and with traffic control infrastructures to globally improve urban mobility and its sustainability. Most urban and agricultural activities will be delegated to teams of robots, typically interacting with each other and with the surrounding physical work to perform their tasks in a collectively coordinate way. The Internet of things vision, to unfold its potentials, will require coordinating the sensing and actuating activities of millions of networked physical objects and their interactions with the physical work. Finally, the vision of "smart matter" consider the future possibility of defining novel materials made up of smart, software-defined particles, that will make it possible to produce adaptive self-shaping and self-repairing artefacts other than possibly pave the way for smart drugs and nano-level in-body therapies.

The kind of systems is indeed exemplary instantiations of CAS, collective adaptive systems. However, the above classes of CAS exhibit distinguishing characteristics with respect to the simpler classes of CAS that have been so far investigated in mainstream CAS researches (which typically deal with CAS composed of homogeneous components with limited autonomy, situated in rather controllable artificial or biological environments) described above exhibit common characteristics that make them substantially different from the classes of systems typically addressed by traditional programming and software engineering technique. In fact:

- They are inherently *heterogeneous* and *socio-technical*, since they require orchestrating the activities of components as diverse as software agents, a variety of sensors and actuators, robots, and – last but not least – they involve the active contributions of humans with their peculiar capabilities and competences.
- They are *situated* in dynamic and unpredictable physical and social environments, where their components are required to be *context-aware* and *socially-aware* in their interaction, and where consequently any coordination scheme has to be adaptive to the context.
- They require the capability of effectively coordinating the activities of a *huge number* (up to the millions) of decentralized *autonomous components*. This implies the impossibility of enacting some centralized scheme of coordination of the activities, as well as the impossibility of enforcing full control over the activities and interactions of some (if not most) of the components.

The above characteristics suggest novel challenges for researches in CAS, to build more ***solid foundations*** for the understanding and harnessing of the emerging classes of CAS of which we will be components and in which, in some way, we will be forced to live and interact.

However, other than understanding the foundations of these novel classes of CAS, this document has emphasized that there is need of ***novel engineering approaches*** to pave the way for the *systematic development* of systems that, despite the limited controllability and dependability of the individuals and their situations, can exhibit predictable and dependable collective behaviour, capable of serving specific purposes. That is, engineering such systems implies the capability of *governing their collective behaviour*, where such governance has a twofold meaning:

- **Shaping** the collective adaptive behaviour of the system, that is assuring at the design level that the system will be able to serve its purpose at the global macro level, despite the impossibility of controlling each individual components and of accurately predicting the dynamics of the environment in which they situate;
- **Controlling** the dynamic collective behaviour of collective systems and its impact on the socio-physical environment in which it situates, to make it possible to enforce constraints (e.g., safety rules) in its behaviours and in its interactions with other systems, and possibly to tune its collective behaviour in order to dynamically meet emerging requirements.

Engineering challenges for governing the behaviour of collective systems imply the definition of an overall framework that should include:

- The definition of novel models for expressing required global behaviours, where functional requirements should express the overall mission of a system, and non-functional requirements can be possibly expressed as constraints over the way such mission is accomplished.
- The definition of qualitative and quantitative models for understanding collective behaviours (e.g., formal models, model checkers, simulators), and tools (e.g., programming languages) facilitating the shaping of specific global behaviours, and the definition of control models and associated decentralized control tools to make it possible to dynamically tune or steer the behaviour of deployed systems.

Specific attention in the definition of the above models and tools should be placed on a number of challenging issues that are peculiar of collective systems dived in complex socio-physical environment and that are particularly critical for the proper functioning and social acceptance. In particular:

1) *Transparency and polycentric governance*. Collective systems will be called to support our everyday activity in a variety of situations, and will be managed by a variety of stakeholders and will interact with a variety of other systems. This raises the issue of understanding how it is possible to guarantee specific purposes in a transparent (understandable and appreciable by humans) way, and in face of multiple and possibly conflicting goals within interacting systems and within a single system.

2) *Higher order emergence*. When individual components of a system are situation-aware they can be made perceiving the global behaviour of the system, and this can possibly influence their individual behaviour, affecting in turn the global behaviours. Such phenomena of second-order (higher-order in general) emergence have not been so far investigated in the area of collective adaptive systems.

3) *Handling dynamic and uncontrollable environments.* Due to unavoidable uncertainty in modelling CAS, there will be need to build live testing environments, running in a ``mirror world'' reflecting real-world data, behaviours and interactions while protecting real individuals, continuously providing feedbacks on CAS behaviour and absorbing patches, upgrades, intervention of engineers.

The results of attacking the above challenges should eventually be a comprehensive "intervention framework" for governing complex socio-technical CAS.

At the time this document is being released, we noticed that the above engineering issues have some relations with FET-PROACTIVE in the 2016-2017 Workprogramme, and in particular with some topics associated to call "FETPROACT-01-2016: FET Proactive: emerging themes and communities". In particular, the call refers to a number of issues and applications/societal scenarios (being human in a technological word, bio-electronic medicines and therapies, new computing paradigms and their technologies, complex bottom-up construction) that are strictly related to the issue of governing collective adaptive systems. That is, all of the systems/technologies of interest to the call somehow involve collective systems and implicitly involve some means to govern some sort of collective systems. Yet, we consider that a specific funding action addressing the engineering and governance foundations of the novel identified classes of CAS is needed and urgent.

Appendix A: challenges references

In this appendix, we report the references to the challenges listed at:
http://www.focas.eu/research-landscape/challenges/

Fundamentals (Science)

- predictability 151
- modelling 157, 124
- expressiveness 150, 149, 148
- analysis 154, 151, 138, 136, 135, 132

(**o**f)

Collective

- coordination (interaction) 156, 148
- heterogeneity 137, 131
- autonomy 143, 131, 126, 125
- governance (incentives, gamification): 12, 15

Adaptive

- adaptation 155, 134
- emergence 150, 129
- reflection 93, 100, 119
- awareness 145, 140, 127, 121

Systems (Engineering)

- distribution 152, 146, 144, 142, 141, 139, 135, 130, 128, 124
- engineering 153, 150, 143
- control 143
- knowledge 156, 145, 140, 133, 122
- security 147
- interdisciplinarity: 11, 15, 17, 30, 37
- adaptation: 101, 22, 90.1, 90.2, 90.3, 112, 103, 140,
    - models for adaptation
    - techniques for adaptation (reflection 93, 100, 119 learning: 134, reasoning and inference: 133, 122, planning)
    - observability, measurement, perspective
    - adaptation to changed requirements, changing environment
    - co-adaptation (distributed adaptation to other aspects of the system itself)
    - adaptation patterns: (local, global, selfish vs collective): 155, 148, 105
- autonomous systems
    - autonomy: 131,
    - autonomicity: 126,
    - control and trade-off between autonomy and controllability (by external operators)
    - socio-technical systems (human in the loop): 125, 126, 8, 27
    - interfacing natural (living systems, from cells to ecosystems) and artificial CAS: 1,8,27,31
    - awareness & self-awareness: 156, 127, 125,
        - uncertainty: 145, 140, 122, 121,
        - *bridge to knowledge engineering: 133,* 10
- distributed systems
    - spatial issues: 141,
    - scale 141,7, 124,11, 124,11,98
    - scalability, scalable analysis techniques: 152, 5,7,20, 146, 142, 139, 135, 124, 92, 120 11, 98, 64, 152, 57,128
    - dependability and resilience: 151, 23, 98, 12,98,104
        - robustness: 23, 107.1, 115, 104, 113, 114, 105, 140, 145
        - availability
        - reliability: 132,

        - maintainability 12,
        - safety: 144, 130,
        - integrity
    - efficiency, performance, optimisation: 26, 38 123,
        - observability, measurement, perspective 95.2, 111, 107.2, 107.1, 119
    - security, privacy, trust: 29, 147, 106
    - heterogeneity / diversity: 137, 131, 127, 6
    - interaction, communication & coordination, interoperability: 156, 7, 19
    - composability: 4,6, 90.3, 118, 113, 114
    - Coordination and interaction 156, 19,113,114, 7, 90,4,113,114,118, 145,144, 148
    - interfacing natural (living systems, from cells to ecosystems) and artificial CAS: 1,8,27,31
- self-organising systems
    - self-organisation 105, 12, 15, 103,90.3
    - emergence: 16,24,25,32,33,34,40: 150, 95.1, 129,
    - designing Emergent Behaviour / Properties 16, 24, 25,129
    - control and trade-off between autonomy and controllability (by external operators)
    - self-organising mechanisms/patterns: 155, 148,
    - verification: 154, 151, 150, 144, 138, 109
    - verifying Emergent Behaviour / Properties 32, 33, 34, 95, 154, 151, 150, 144, 138, 109
- Modelling CAS: 157, 96, 116, 107.2, 107.1, 94, 82, 124,
    - keep it simple: 9
    - integrating diversity of or extending modelling approaches (e.g. agent-based models) and theories integration (e.g. game theory), dealing with legacy: 1,6,13,21,22,39
    - lack of theory and foundation: 14, 30, 35, 157, 30, 82,116, 94
    - formal languages and expressiveness: 149
    - bridging the gap between individual/micro/local (bottom up) and collective/macro/global (top down) models 18, 40, 89, 108, 82, 97, 3, 124
    - verification and validation: 154, 151, 138, 136, 135, 132, 107, 95, 111, 119
    - observability, measurement, perspective 95.2, 111, 107.2, 107.1, 119
    - designing, requirement specification, goals and engineering CAS, model driven methods for CAS, adaptable "by design" 85,87, 95.3, 95.4, 110, 118, 96, 89, 108, 93,100
    - Programming languages: 143
- Governance of CAS
    - **- sustai**nability of CAS: 36
    - - observability, measurement, perspective, goal 95.2, 111, 107.2, 107.1, 119
    - - incentives and decision-making: 2,25,28
    - - deploying and operating CAS 101

- unclassified/meta: 81, 84, 86, 88, 99, 102, 117, 128