Towards the development of Socio-Technical Systems within the C-K design theory

Stefano Borgo
Laboratory for Applied Ontology
Institute of Cognitive Sciences and Technologies, CNR
Trento, Italy

(8^{th} Design Theory SIG, Paris 26-27 January 2015)

Socio-technical systems (STSs), understood as systems that are crucially based on the interaction among a technological and a human or social component like city subways, hospital services or mobile communication systems, are particularly interesting due to their complex structure and dynamics, the typically large number of objects involved, the different roles they allow, the variety of interactions they activate, the interfering dynamics of the parts, the chances of unexpected evolution as in case of new public regulations or the development of new social and cultural behaviours. The design of different aspects of STSs has attracted attention from the 1950s and has led to a variety of guiding principles that have their roots, at least in the western world, in the general viewpoint of the different socio-cultural periods [7]. While important outcomes have been obtained from perspectival modeling of STSs, a methodology for their general modeling is still lacking since it requires conceptual tools to model human agents and organisations as well as methodologies for coherently integrate behaviours and norms for both the technical and social elements.

Today things seem to be ripen for moving forward on the development of more general and comprehensive models of STSs. Indeed, from the 1990s there has been an important development of formal theories for multi-agent systems. This research area has introduced logical frameworks to distinguish different types of agents and agentive groups, has shed light on how to model types of coordination and power across agents, and has provided ways to distinguish agents' behaviour depending on aims and generic attitudes.[1]
During the same period, applied ontology has been introduced as the approach for analysing properties of objects and processes, and to build integrated information systems where knowledge can be reliably shared.[2] With the method-

[1]See for example the International Foundation for Autonomous Agents and Multiagent Systems (AMAAS) activities at `http://www.aamas-conference.org/`.

[2]See for example the International Association for Ontology and its Applications (IAOA) activities at `www.iaoa.org`.

ologies developed within applied ontology, one can make explicit and compare approaches on basic engineering notions like that of artefact [1] or of functionality [2]. The methodologies allow to:

- analyse and (formally) characterise objects and their relationships; and
- model the variety of interactions across (groups of) agents, natural entities, devices, software, and organisations (broadly understood).

One important advantage of research in multi-agent systems and applied ontology, beside the flexibility and generality of these approaches, is the possibility to formalise the models in formal logical. This makes possible to exploit them with automatic reasoners for different tasks like, e.g., verify consistency and conceptual coherence. In principle, one can use them also to foresee how the behaviour of a STS would change when changing the knowledge of (human or artificial) agents or their general attitudes. Of course, we are just at the beginning and there are many open problems in the actual development of these models [5]. Yet, the fundamental elements have been laid out already. The vision is that these models are now at our reach and are expressible in formal representation languages so that they can be verified and updated by software systems, maintained in information systems as knowledge repositories, and exploited to check general properties like liveness and safety using the same techniques one would apply to prove a theorem in logic.

The main problem to face now is the integration of the approaches we have mentioned. In order to develop and coherently model the different parts of a STS and their interactions, we need a general design framework. Among the design theories available today, the C-K design theory [4] proposes an interesting distinction between the component reserved to knowledge and belief (the K space) and the component devoted to the study of creative and speculative ideas (the C space). If we want to take advantage of this structure, we first need to understand how to isolate, use and manage STS information within the K space. We suggest to rely on three general pillars, namely, context, information layer, and function, which now we briefly present.

Context is a largely used term but, unfortunately, lacks a precise meaning [6]. Depending on the domain, it can be understood as information that characterises the state of an entity like a person, a location or a (possibly complex) object. This notion is interesting in our approach since it allows to carve out the information that needs to be taken into account in a specific design problem. Contexts, when properly characterised, are like filters: they identify the knowledge needed (or even compatible) within a perspective. This feature needs to be exploited to make design and reasoning in STS manageable.

The notion of information layer is also quite general and is connected to a philosophical view known as 'level of reality' (see, e.g., [3]). We are interested in an application oriented view of these layers which are introduced to separate homogeneous types of knowledge as well as forms of interdependencies and interactions. While context are entity dependent, information layers provide a fixed setting to distinguish, for instance, the knowledge related to the physical, the biological and the social domains. It is still unclear which are the best ways to implement information layers, especially if one aims to include forms of knowledge granularity, but the usefulness of this notion is quite clear.

Finally, functions help to model and track the transformations of (and within) a system making possible to describe and explain the internal and general dynamics as well as to foresee evolutions of the systems as a whole and those of its parts. There is a large literature on functionality in engineering design. What is missing is the understanding and formal modeling of social functions, including how they interact with engineering functions.

The notions of context, information layer and function are general and this should be seen as a positive feature: we need general conceptual tools to create models suitable for disparate views and goals. These three elements, that are to all effects framing tools for modeling, provide multiple ways to cluster, interpret, and integrate different types of knowledge. To use the C-K design theory as a framework to design and monitor STSs we suggest to first integrate these framing tools in the C-K structure, and then exploit them for modeling the knowledge relative to the technical and social components of STSs. This opens a series of questions which, fortunately, are not completely new and, in some cases, solutions are already available, see, for instance, the studies on belief revision [8]. Yet, there are issues that are largely unexplored. Here we initiate a discussion on one of them: the modeling of contexts as knowledge filters for STS models.

We propose to see contexts as descriptions, that is, as information entities. This leads to isolate two types of context that we can roughly distinguish according to their connection to the real world. On the one hand, we have *abstract* contexts like, for instance, the transportation system or, at a more specific level, the train transportation system. The context here is the description of (generic) entities that are relevant for a transportation system, that is for the moving of physical objects (perhaps including people); thus it lists entities like the means, equipment, personnel, organisation and dynamics that are usually applied to perform the service including their roles and associated capacities, powers, duties and norms as well as their relationships. On the other hand, some contexts are somehow present (materialised) in the actual world, i.e., refer to a specific spatio-temporal location and entities in it like, for instance, the French transportation system in 2014 or, at a more specific level, the RER transportation system in Paris in 2014. Besides the information relative to transportation as selected by the corresponding abstract context, the *materialised* context lists also the specific real entities that are part of the system in the intended spatio-temporal location possibly including, in the given example, the actual regulations on safety and labor unions, cultural habits and client expectations. By applying the ontological perspective to identify and classify both abstract and materialised contexts, one can use them to divide the K space into islands of 'contextual' knowledge. The mechanism is not trivial but in a logical framework from the entities and relationship provided by the chosen contest, it becomes possible to extract the relative information from the K space. The possibility to partition knowledge relatively to a context is just a first, yet important, step towards a methodology to design and manage STS models within the C-K theory.

References

[1] S. Borgo, M. Franssen, P. Garbacz, Y. Kitamura, R. Mizoguchi, and P. E. Vermaas. Technical artifacts: An integrated perspective. *Applied Ontology Journal*, (to appear), 2015.

[2] P. Garbacz, S. Borgo, M. Carrara, and P. E. Vermaas. Two ontology-driven formalisations of functions and their comparison. *Journal of Engineering Design*, 22(11-12):733–764, 2011.

[3] C. Gnoli and R. Poli. Levels of reality and levels of representation. *Knowledge organization*, 31(3):151–160, 2004.

[4] A. Hatchuel and B. Weil. C-K design theory: an advanced formulation. *Research in Engineering Design*, 19(4):181–192, 2009.

[5] A. J. Jones, A. Artikis, and J. Pitt. The design of intelligent socio-technical systems. *Artificial Intelligence Review*, 39(1):5–20, 2013.

[6] H. Koç, E. Hennig, S. Jastram, and C. Starke. State of the art in context modelling–a systematic literature review. In *Advanced Information Systems Engineering Workshops*, pages 53–64. Springer, 2014.

[7] E. Mumford. The story of socio-technical design: reflections on its successes, failures and potential. *Information Systems Journal*, 16(4):317–342, 2006.

[8] P. Peppas. Belief revision. *Handbook of knowledge representation*, 3:317–359, 2008.