The Design of Intelligent Socio-Technical Systems

Andrew J I Jones[1], Alexander Artikis[2] and Jeremy Pitt[3]

[1]*Department of Informatics, King's College London, UK*

andrewji.jones@kcl.ac.uk

[2]*National Centre for Scientific Research 'Demokritos', Athens, Greece*

a.artikis@iit.demokritos.gr

[3]*Department of Electrical & Electronic Engineering, Imperial College London, UK*

j.pitt@imperial.ac.uk

Abstract The design of intelligent socio-technical systems calls for careful examination of relevant social and organizational concepts. We present a method for supporting this design process, placing emphasis on different levels of formal characterization, with equal attention to both the analysis of concepts in a formal calculus independent of computational concerns, and the representation of concepts in a machine-processable form, fully cognizant of implementation issues – a step in the method we refer to as *principled operationalization*. There are many tools (i.e. formal languages) that can be used to support the design method; we define and discuss criteria for evaluating such tools. We believe that, were the method proposed to be adopted, it would enhance the state-of-the-art in the systematic design and engineering of socio-technical systems, respecting the fundamentally interdisciplinary nature of those tasks, in both their theoretical and practical dimensions.

Keywords*: conceptual and computational models, socio-technical systems, multi-agent systems, synthetic method*

1 Introduction

………there is no such thing as philosophy-free science; there is only science whose philosophical baggage is taken on board without examination. [Dennett 1995, p. 21]

To a significant extent, research in Computer Science that aims to develop socio-technical systems has to address issues pertaining to the interpretation of social and organizational concepts. The components of socio-technical systems, be they artefacts or humans, carry out their work by interacting with each other against a social, organizational or legal background. The field of Autonomous Agents and Multi-agent Systems has for some time represented an obvious example of this work, but the important part played by social concepts extends into other parts of Computer Science too. Consider – to mention just three further domains – Computer Security, where the notions of *trust*, *reputation* and *role* have figured prominently; E-commerce, where the representation, formation and fulfillment of contracts is fundamental; and E-government, where representing and reasoning about policies and norms are essential.

In Biology and Social Science, in Jurisprudence, and in Analytical Philosophy, among other disciplines, we find examples of conceptual models designed to enhance our understanding of the nature of organized interaction. In writing this paper, our initial question was this: in their construction of so-called *computational* models of social concepts, such as those mentioned in the previous paragraph, have computer scientists been sufficiently informed by *conceptual* models of social phenomena, the construction of which was not motivated by computational considerations, but aimed primarily to reveal, in a systematic fashion, the structure and interconnections of the concepts themselves? Through its attempt to answer that question, the principal contribution of this paper is a proposed approach to the engineering of socio-technical systems that respects the interdisciplinary nature of the task, in regard to both its theoretical and practical dimensions.

We proceed in Section 2 by giving some examples of work that would justify a negative answer to our initial question, and we explain their shortcomings. Against that background, Section 3 describes an approach to the engineering of socio-technical systems in which rich, conceptual-analytical models and computational frameworks are combined, providing a basis for *principled operationalization*, observing that similar methodological concerns have arisen in the field of biologically-inspired computing. We describe the approach in terms of a sequence of steps and, accordingly, in Section 4 we formulate and illustrate adequacy criteria that, ideally, the key steps should satisfy. In the concluding section we suggest, in particular, that if our general methodological proposals were to be adopted, they should have significant consequences for the ways in which researchers are trained, not least in the area of Autonomous Agents and Multi-agent Systems.

We ask the reader to consider this paper as an invitation to enter a discussion of how the different elements involved in the design of socio-technical systems should best be pulled together. We have attempted here to provide the beginnings of a coherent framework within which conceptual and computational work can be effectively combined. But these are controversial and difficult matters. While we are prepared to concede that a more mature method might take a form rather different from the one proposed herein, we are nevertheless convinced of the necessity of a genuinely interdisciplinary synthetic method.

2 Motivating Examples

In this section we consider three examples of work on the engineering of socio-technical systems in which social concepts – specifically *trust*, *role* and *normative power* – have figured prominently.

2.1 Normative Power[1]

Any reasonably comprehensive model, formal or informal, of norm-governed multi-agent systems must be able to accommodate norms pertaining to institutionalized normative power, in addition to those that express obligations and permissions. It is a commonplace feature of organizations that particular agents, individually or collectively, are empowered to carry out actions, the consequences of which have a significant bearing on the way the organization is governed or administered. For instance, some public officials/bodies will be empowered to create a state of marriage between two individuals, or to validate wills, or to appoint some other persons to particular roles (including roles that themselves involve the possession of powers), or to create or modify laws and regulations. Powers of this sort are types of rights, or entitlements, that some agents have, and others lack. There is a substantial body of literature, stemming from Hohfeld (1913), that focuses on the systematic characterization of types of rights-relations, including in some cases formal analyses of these relations expressed in terms of a small set of basic operators drawn from modal logic (Kanger 1957,1972, Pörn 1970, Lindahl 1977, Jones and Sergot 1993, 1996).

Oren et al. (2010) present a model of what they call ‘normative power’, which they associate with the power to create and/or modify norms. While they refer to the Hohfeldian tradition, they make no use of the analyses offered therein, preferring instead to characterize normative power by means of a first-order logic tuple, the key element of which is called ‘mandators’. “Mandators is a set of predicates identifying agents” (op. cit., p. 817), and “A mandator of the form *professor(x)* means that any agent in the professor role is able to exercise the power”, for instance the power to place a student under an obligation to write a conference paper (op. cit., p. 819). Note that the interpretation of what it means for an agent to be able to exercise a normative power is not here

[1] An earlier version of Section 2.1 appeared in Jones et al. 2011.

explicated; rather, it remains implicit in the natural-language reading the authors assign to the ‘mandator’ predicate. This attempt at modelling jumps straight from an *informal* description of the concept of normative power to a first-order logic representation – a transition that is presumably motivated primarily by considerations of computational tractability.

The practice of giving a rather simple, but computationally convenient, representation of complex social concepts is quite widespread in Computer Science – the areas discussed in sections 2.2 and 2.3, to follow, provide further examples of it. But it is a problematic practice because it provides no clear picture of the nature of the simplifications made, and thus also no proper framework for assessing whether a system implemented on the basis of such a computational model behaves in a way that adequately reflects the properties of the social concept itself.

2.2 Role-based Access Control

The NIST model for role-based access control (RBAC) (Sandhu et al. 2000) formed the basis for the ANSI RBAC standard. The model comprised a four-step sequence of increasing capabilities, with each step containing the previous one: Flat RBAC (step 1); Hierarchical RBAC (step 2); Constrained RBAC (step 3); Symmetric RBAC (step 4).

“Flat RBAC embodies the essential aspects of RBAC. The basic concept of RBAC is that users are assigned to roles, permissions are assigned to roles and users acquire permissions by being members of roles” (op. cit., pp. 48-49). Step 2 then adds role hierarchies, either general or restricted, by which senior roles will acquire permissions assigned to roles below theirs in the hierarchical ordering. Step 3 adds constraints that enforce conflict of interest policies, to deal with conflicts that may arise when a user belongs to more than one role, and at step 4 there is the further requirement that permission-role assignments can be reviewed.

In their introductory section, the authors maintain that “….the basic role concept is simple: establish permissions based on the functional roles in the enterprise, and then appropriately assign users to a role or set of roles” (op. cit., p.47). But very soon thereafter they allude to a structure that is considerably more complex: “Roles could represent the tasks, responsibilities and qualifications associated with an enterprise”. It is revealing that the latter description of roles is by no means confined to mere permissions, since it appears that some key aspects of the overall NIST model are motivated by the largely unexplicated assumption that agents get assigned to particular roles in virtue of their qualifications, and that – as role-holders – they also acquire obligations associated with the organizational tasks for which they are deemed to be responsible. (Note also the remark: “A role is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of a role” (op. cit., p. 51).)

It is instructive to consider how a formal-logical analysis of the structure of the role-concept could have usefully served the development of practical applications of the NIST model. For instance, it could have supported Constrained RBAC by providing a means for formally testing the consistency of the package of norms associated with a given role, thereby facilitating the systematic investigation of the role conflicts that can arise when an agent is assigned to more than one role. Secondly, it could have supported the investigation of different ways of designing role hierarchies and their inheritance properties. Thirdly, consider the claim that “A permission is an approval of a particular mode of access to one or more objects in the system. The terms authorization, access right and privilege are also used in the literature to denote a permission” (op. cit., p.51). For a number of practical purposes, the distinctions between those concepts can be safely ignored. But in some contexts it would be important to distinguish between what an agent is permitted to do and what he is authorised to do in the sense of being *empowered* to do it. (This is a distinction, recognized by Hohfeld, that has also been made explicit in the formal theory of rights.) One example (op. cit., p. 53*ff*) concerns the roles of Test Engineer and Project Supervisor, where the authors suggest that even though the Project Supervisor role is above that of the Test Engineer in the hierarchy, it might be important not to allow the Project Supervisor to inherit all the permissions of the Test Engineer, since the former might well lack the technical competence of the latter. However, one might nevertheless want to insist that any powers held by the Test Engineer, for instance in regard to the hiring and firing of technical assistants, should also be held by the Project Supervisor – who should perhaps also be empowered to overturn any hiring/firing proposal made by the Test Engineer.[2]

In some very helpful comments on an earlier draft of this section, the late Steve Barker (personal communication, October 2011) indicated that some (but not all) of the issues raised in the previous paragraph have been addressed in the later RBAC literature. However, our view is that – even if those later developments deal successfully with some of the shortcomings of the original RBAC model – the process of piecemeal enhancement would have benefited by being informed and directed, from the outset, by a comprehensive, precise model of the role concept itself.

2.3 Trust

A third source of motivating examples is provided by the literature on the design of socio-technical systems addressing issues of trust in agent interaction. A very useful survey of that literature has recently appeared (Pinyol and Sabater-Mir 2011), in which the authors present a classification of a range of models in terms of several dimensions. Among the latter is what they choose to call the 'trust' aspect, which they describe as follows: "Trust can be seen as a process of practical reasoning that leads to the decision to interact with somebody. Regarding this aspect, some models provide evaluations, rates, scores etc. for each agent to help the decision maker with a final decision. Instead, others specify how the actual decision should be made. From our point of view, only the latter cases can be considered trust models" (op. cit., Section 3.1). It is noticeable that seven of the eighteen models in the authors' survey do not unequivocally embody a trust model in the above sense, but focus on the calculation of measures that are supposed to facilitate the decision-making process, without providing an explicit mechanism showing how decisions are to be made. Furthermore, the authors maintain, all but three of the remaining eleven models fail to provide an explicit representation of "….the epistemic and motivational attitudes that are necessary for the agents to have *trust* or to hold social evaluations" (op. cit., Section 3.2). The three exceptions here are the cases that satisfy the authors' *cognitive* dimension, which – as they clearly indicate – derives from the approach of Castelfranchi & Falcone (1998), later developed by Herzig et al. (2008).

Some of the authors' remarks about the cognitive dimension are of particular interest from the standpoint of the present paper. For instance, concerning models that satisfy the cognitive aspect they say: "From a software agents perspective, this endows the agents with a clear capacity to *explain* their decisions and to reason about the trust structure itself………..In this sense, for the models that achieve a cognitive representation, final values of trust and reputation are as important as the structure that supports them" (op. cit., Section 3.2). But then, revealingly, they add: "These models are usually very clear at the conceptual level, but lack in computational aspects." Immediately thereafter they give the opposing picture by adding: "…..models that are not endowed with this [cognitive] property consider the model as a black box that receives inputs and issues trust and reputation *values*……….the internal calculation process cannot be considered by the agent, only the final values. Moreover, the integration with the other elements of the agent remains unclear because motivational attitudes are assumed or mix with the calculus. However, their computational aspects are usually quite well defined…." (op. cit., Section 3.2).

These comments very strongly suggest the need for a methodology that brings together *both* conceptual modelling *and* a computational framework informed by it. Just one of the eighteen approaches considered in the survey achieves this synthesis, according to the authors; concerning that one model they say, in their concluding remarks, that it "…..summarizes one of the most prominent future research lines in trust and reputation models: *implementable cognitive models*" (op. cit., Section 5). In our view, the key point about those cognitive models is that they are *conceptual* models, designed primarily to clarify the *trust* concept itself; non-cognitive analyses of

[2] One of the reviewers of this paper pointed out that the NASA Challenger Space Shuttle disaster could provide an interesting real-world example illustrating the respective roles of Test Engineers and Project Supervisors.

trust might also be possible, but the essential methodological requirement emerging from the survey pertains to the need to *integrate* the conceptual and computational aspects.

3 Towards a Method for Designing Intelligent Socio-Technical Systems

This section outlines the structure of an approach to engineering intelligent socio-technical systems in which an abstract analysis of social concepts informs the development of a computational framework, providing a suitable platform for system implementation.

We are here, in part, building on the synthetic method underlying some research in artificial societies and artificial life (Steels and Brooks 1994). The main steps of the synthetic method involve generalizing from some observations of phenomena to produce a theory, on the basis of which an artificial system can be constructed and then used to test predictions deriving from the theory. The outcome of applying the synthetic method is to engineer an artificial system, with the resulting animation, experiments or performance serving to support or refute the theory. Several other attempts to apply ideas from the social sciences to the design of computational systems (see, e.g., Edmonds et al. 2005) have followed a similar pattern. Furthermore, researchers in biologically-inspired computing, notably those concerned with artificial immune systems (Andrews et al. 2010), have developed a comparable approach.[3]

3.1 Structure of the method

The root of the concerns we highlighted in Section 2 may be expressed in the following way: we fully accept that, in the design of socio-technical systems, the need for computational tractability makes it probable that there will have to be some degree of simplification of the principal social concepts involved; but in the interests of good scientific practice – and thus, also, good engineering practice – it is essential to achieve as clear a picture as possible of just what it is that is being simplified. We need first to have a clear characterization of the phenomena, before we set about simplifying them. Any computationally motivated simplifications should be carried out against the background of, and should be properly informed by, precise models of the social concepts themselves. And, crucially, the construction of those conceptual models should not itself be constrained by considerations of computational tractability.

We present our proposals in terms of different steps pertaining to the description and analysis of the members of a set *S* of observed social phenomena, as illustrated in Figure 1. The principal steps are theory construction, formal characterization, and principled operationalization.

Step1 representations of the members of *S* are characterized by the natural-language terms that are used to denote the social phenomena concerned – terms such as *empowerment*, *role*, *trust*, and so on.

The process of *formal characterization* is the process leading from *Step1* representations to *Step2* representations. A *Step2* representation must be expressed in a formal language or ‘calculus’ of some kind, where by ‘calculus’ we mean any system of calculation or computation based on the manipulation of symbolic representations.

However, there are *Step2* representations of various sorts, which for our purposes are appropriately divided into two sub-steps, or phases. *Step2-Phase1* representations define a conceptual framework for the phenomena in *S*, in which conceptual analyses are expressed in terms of, for instance, a formal-logical language; the key point about *Step2-Phase1* representations is that they aim to provide an analysis of conceptual structure, identifying the fundamental elements of which complex concepts are composed, and articulating the principles governing their

[3] We note that it has been observed in biologically-inspired computing that some programmed systems have ‘drifted’ from the original source of inspiration, and so become inadequate either as a simulation for answering questions about the nature of real immune systems, or as a foundation for other related problems. This observation provides an interesting parallel to our concerns about trends in the design of socio-technical systems.

composition and inter-relations. Crucially, *Step2-Phase1* representations are constrained primarily by considerations of expressive capacity, *not* those of computational tractability.

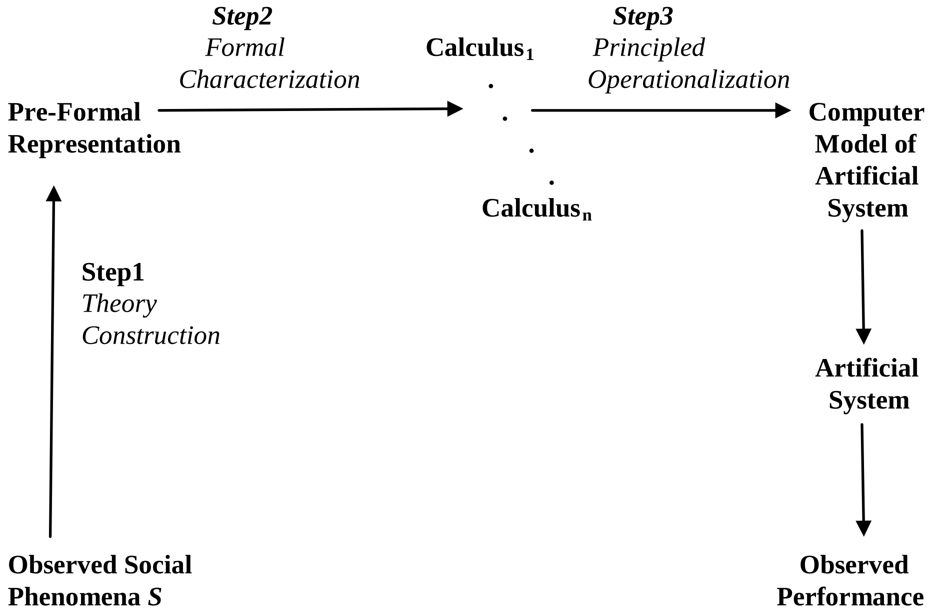

Fig. 1. Simplified Diagram Representing the Proposed Method for Engineering Socio-Technical Systems[4]

By contrast, it is at the *Step2-Phase2* stage that issues of computational tractability begin to come into play. A *Step2-Phase2* computational framework models the conceptual framework of *Step2-Phase1* in terms of a language, or languages, that are themselves amenable to the development of software implementations; the key points to note about *Step2-Phase2* computational frameworks are that the principles governing their composition are informed and guided by the conceptual characterizations of *Step2-Phase1*, but that they may well involve some degree of simplification, or approximation. Crucially, however, on this approach the designer of a computational framework will have a very clear picture, from *Step2-Phase1*, of the nature of the simplifications or approximations that may have been made.

Step2-Phase1 representations are essentially theory-facing, whereas *Step2-Phase2* representations are essentially implementation-facing. As we shall see below in Section 4, the orientations of the two phases will determine their respective adequacy criteria. However, nothing we have said so far should be taken as suggesting that a formal conceptual characterization could *never* itself *also* be a computational framework: it will all depend on the nature of the concepts to be analysed, and on the available formal and computational tools. The recommendation to adopt two *Step2* phases is motivated by the need to *guard against* trying to force subtle societal concepts into the straitjacket of some particular computationally tractable language.

One further observation should be made about the relationship between the two phases of *Step2*. We have emphasized that some of the conceptual detail that is captured in the *Step2-Phase1* model might be omitted from the *Step2-Phase2* computational framework; but we should also point out that there may well be abstractions that can be tolerated at the *Step2-Phase1* level that cannot be ignored in an implementation-facing framework, for example the representation of *time*, and of *the means by which* a particular state of affairs is to be brought about. We return to this point in Section 4.2.

[4] The diagram is simplified in that it depicts the method as unidirectional. However, there are important aspects of two-way interplay between the key steps. We describe some of these in Section 3.3.

Step 3 representations are exemplified not so much by formalisms but by tools that are employed in moving from the computational framework to a *model* of the artificial system, with algorithmic intelligence of the agents embedded in identifiable system processes. This is the transition that we call *principled operationalization*. Operationalization may well be selective, vis-à-vis the computational framework; but it is *principled* operationalization in that it is conducted in the full knowledge of which selections have been made, and why.

3.2 Step2 exemplified

3.2.1 Formal characterization with modal logic: Step2-Phase1

The focus here will be on some formal-logical tools, drawn from modal logic, that have been used in the analysis of the group of concepts discussed in Section 2. We do not of course mean to suggest by this choice that modal logic is the *only* tool suited to the analysis of social concepts: in some cases other logics, such as first-order logic, may be adequate to the task; or it may be more appropriate to use quite different sorts of formal model – consider, to give just one example, the application of games-theory to the characterization of signalling (Lewis 1969, Skyrms 2010).

In the case of *trust*, we referred in Section 2.3 to the work of Castefranchi and Falcone, later refined by Herzig and Lorini, who relied essentially on the modalities contained in the BDI (belief, desire, intention) framework, widely used in theoretical AI, supplemented by logical formalisms for representing agency and time. The authors' aim in those works was to provide a clear explication of their intuitions regarding the nature of the *trust* concept, by giving a formal-logical specification of the conditions under which it is true to say that one agent trusts another to do something.[5] We refer the reader to the original sources for details; our main point is to emphasize that their primary goal was formal conceptual analysis, driven not by computational considerations, but by recognition of the need to achieve a clear understanding of a complex social concept. (Even a cursory look at the papers contained in, for instance, Castelfranchi and Tan 2001 reveals the extent of that complexity.)

As indicated in Section 2.1, the literature on the formal theory of norms provides evidence that appropriate combinations of modalities can serve to represent many *rights*, as these are interpreted in the tradition of Hohfeld's *fundamental legal conceptions* (Hohfeld 1913, Kanger 1957, Kanger and Kanger 1966, Pörn 1970, Lindahl 1977, Jones & Sergot 1993), in addition to representing the *obligations* and *permissions* that in part define any norm-governed system of interacting agents. However, as Hohfeld himself recognised, rights pertaining to normative *empowerment*, as expressed by, for example, 'The Head of Department, but not the Director of Administration, is entitled to assign teaching duties', cannot be properly analysed in terms of *permission*. So a modal-logical language containing modalities for *agency*, *obligation* and *permission* will need the addition of some further elements, to reflect the idea that the Head of Department's assignment *counts as* a valid assignment, whereas any attempt by the Director of Administration to assign teaching duties would not so count. A modal conditional connective (the so-called *counts-as* conditional) was first formally defined in Jones & Sergot 1996, in order to represent the idea that, relative to some institution or organization, the performance by some designated agent of a particular type of action, often in a specified context, counts as a means of creating a particular state of affairs.[6] With this supplementary 'building-block' in place, the resulting multi-modal language is expressively rich enough to facilitate the representation of a range of the different types of policies and regulations used to govern agent interaction. And, since the component modal building-blocks are all elements of well-defined logics, they also facilitate the systematic investigation of relationships between different types of norms, the consistency of sets of norms,

[5] Herzig and Lorini elaborate the analysis in terms of two sets of conditions, for what they call *occurrent* and *dispositional* trust.

[6] Various alternative accounts of *counts-as* conditionals appeared later on, and these are summarized and critically compared in Grossi and Jones, in press.

and so on. But the construction of the formal, analytical models – at least in the cited works – was not itself constrained by considerations of computational tractability.

As regards the role concept, a clear account of a formal-logical model was first provided in Pörn 1977, pp.61-63.[7] In summary, Pörn identified two main components of what he called a role structure: the *role condition*, which specifies the properties that an individual must possess in order to qualify as a member of a given role category; and the *role norms*, which specify the norms that apply to any individual occupant of that role. So, for instance, only persons having particular qualifications and abilities satisfy the conditions for membership of the role of *medical doctor*; and any medical doctor is subject to a range of directive norms requiring certain standards of conduct, and permitting actions that, in many instances, would be forbidden for most non-physicians.

While it is clear that normative and action modalities, of the kind Pörn proposed in his 1977 book, would be key components of a formal-logical model of role structures, we can with hindsight suggest that the *counts-as* connective could further enhance that model, in two ways.[8] First, the conditionals that themselves represent the role condition say that an individual possessing such-and-such properties counts as a member of the given role. Secondly, the norms associated with a role are often not exclusively of the kind that impose obligations and grant permissions, but may also specify *powers* conferred on the role-holder – for instance a medical doctor will ordinarily be empowered to write valid drug prescriptions, to sign off death certificates, and so on.

Once again it is worth stressing the advantages that accrue from adoption of a *formal-logical* conceptual model. For instance, it will facilitate the systematic investigation of *role conflict*, within a given role, between different roles in a given organization, between roles in different organizations, and so on. It can also provide a platform for clear specification of role hierarchies. In short, the employment of a formal conceptual characterization of this kind at the *Step2-Phase1* level would provide rather precise guidelines for the construction of a coherent and suitably flexible computational framework for dealing with roles at *Step2-Phase2*.

3.2.2 Formal characterization with action languages: Step2-Phase2

Several approaches have been proposed in the literature that may be classified under *Step2-Phase2* of our approach. A notable line of research concerns action languages from the field of Artificial Intelligence. Fox and colleagues, for example, have used the Situation Calculus (Pinto and Reiter 1993, McCarthy 1963, Reiter 1993, Levesque et al. 1998) for enterprise modeling (Fox et al. 1998, Grüninger and Fox 1994), while Brewka has used this language for formalizing dynamic argumentation systems (Brewka 2001). The Event Calculus (Kowalski and Sergot 1986) has been very frequently used for norm-governed system specification and execution – (Marín and Sartor 1999, Yolum and Singh 2004, Fornara and Colombetti 2009, Artikis and Sergot 2010) are but a few examples. *C*+ (Giunchiglia et al. 2004, Akman et al. 2004), an action language with transition system semantics, has been used by Chopra and Singh (2006) to formalize ‘commitment protocols’, while Artikis et al. (2007) have used this language to develop executable MAS specifications in terms of *institutionalized power*, *permission* and *sanction*.

Craven and Sergot (2008) have presented a formal framework, called a ‘coloured agent-stranded transition system’, which adds two components to a labeled transition system. The first component partitions states and transitions according to various ‘colourings’, used to represent norms of two different kinds. *System norms* express a system designer’s point of view regarding which system states and system transitions are legal, permitted, desirable, etc. A second set of individual *agent-specific* norms are intended to be taken into account in an agent’s implementation. The second component of a ‘coloured agent-stranded transition system’ is a way of selecting, from a global system transition representing many concurrent actions by multiple

[7] Pörn’s book is full of valuable insights regarding the formal modelling of various aspects of social interaction. Regrettably, his work has been largely overlooked by researchers in the field of Agents and Multi-agent Systems.

[8] A suggestion of this sort was first put forward in the description of the multi-modal logical framework developed as part of the EC-financed ALFEBIITE project (IST-1999-10298, 01-02-2000 to 30-11-2003).

agents, an individual agent’s actions, or ‘strand’, in that transition. This allows one to say that in a particular transition it is specifically one agent’s actions, rather than those of some other agent, that are in compliance, or non-compliance with a system or agent-specific norm. This framework supports the characterization of several different categories of non-compliant behaviour, distinguishing between various forms of unavoidable or inadvertent non-compliance – behaviour where an agent does ‘the best that it can’ to comply with its individual norms, but nevertheless fails to do so because of the actions of some other agents – and behaviour where an agent could have complied with its individual norms, but did not.

Sergot (2008) has presented a further development of the aforementioned framework. The ‘colourings’ used to represent norms are separated from the more general structure of an agent-stranded transition system. Sergot presented a formal language for talking about properties of states and transitions, including but not restricted to their ‘colourings’, and for talking about agent strands of transitions. The language has operators for expressing that a particular agent, or group of agents, *brings it about* that such-and-such is the case, in the sense that it is responsible for, or its actions are the cause of, such-and-such being the case. The resulting logic has a strong resemblance to the logic of action/agency in Pörn (1977), except that instead of talking about an agent’s bringing about a certain state of affairs, one talks about an agent’s bringing it about that a transition has a certain property. In general, Sergot’s framework has been informed by the kinds of abstract conceptual models of (aspects of) normative systems discussed above in Section 3.2.1.

Sergot’s formal language has been implemented in the form of the ICCALC model-checker, which may be used to evaluate formulas on a given transition system. ICCALC is a re-implementation of the ‘Causal Calculator’ (Akman et al. 2004), which was developed as a means of performing computational tasks using the action language *C*+. ICCALC also supports *nC*+ (Sergot and Craven 2006), an extended form of *C*+ designed specifically for representing normative and institutional concepts. An action description in *nC*+ defines a coloured (agent-stranded) transition system of a certain kind.

3.3 A note on the interplay between steps

As mentioned above, Figure 1 is simplified, and fails to bring out the fact that the design process is frequently two-way, not unidirectional. For instance, regarding *Step1* and *Step2-Phase1*, it is evident that the process of formalizing natural-language sentences – consider legal rules and policy statements – can often reveal ambiguities.[9] Similarly, as indicated in Section 2.2, a formal-logical model of roles, in which it would be possible to test the consistency of sets of role assignments, could be used to detect potential incoherence in a *Step1*, informal specification of the role structure of a given organization. Furthermore, the transition from *Step2-Phase1* via the computational framework to implementation might reveal shortcomings in the underlying conceptualization of the phenomena; for instance, an attempt to implement an Agent Communication Language for which deceitful informing is a genuine possibility would expose the shortcomings of a formal conceptualization of communication protocols of the type presented in the FIPA standard (FIPA 2002), which appears to be so closely wedded to contexts in which the relationship between communicator and audience is assumed to be helpful and cooperative that it is not clear whether room is left for deceitful communication. So what becomes clear at *Step3* could well lead to recognition of the need to revise aspects of *Step2* and, indeed, in the FIPA case, to revise the underlying *Step1* informal theory of communication.

4 Adequacy criteria

In this section we specify, discuss and exemplify adequacy criteria for formalisms in the two phases of Step2 of the proposed method.

[9] Over fifty years ago, Layman Allen began arguing the case for applying logic as a tool in legal drafting; see, e.g., Allen 1957.

4.1 Adequacy criteria for Step2-Phase1

Inevitably, the analysis of a concept *C* begins from a number of intuitions about the meaning and content of *C* – intuitions that may initially be quite vague and less than well-structured. The application of formal tools to the analysis of those intuitions is intended to serve the purpose of facilitating a more perspicuous articulation of the structure and content of *C*, expressed in terms of well-defined component elements. Accordingly, the dominant consideration in assessing the degree of adequacy of a proposed conceptual characterization is *expressive capacity*.

Under the broad criterion of expressive capacity, we can identify a number of sub-criteria:

(1) the capacity to identify the principal elements;
(2) the capacity to test for consistency;
(3) the capacity to articulate specific, characteristic aspects of the concept;
(4) the capacity to 'place' the concept in relation to its near relatives.

The first capacity is to identify the basic 'building-blocks' out of which concept *C* is composed, and exhibit the structure of the composition of *C* from those elements. So, for example, the analyses of occurrent and dispositional trust in Herzig and Lorini 2010 show how those compound concepts are built up from modal building-blocks representing *belief*, *action*, *choice*, among others. Likewise, the analysis of institutionalised power in Jones and Sergot 1996 exhibits the structure of the concept in terms of, essentially, modalities for action and the *counts-as* conditional.

The second capacity is to test for consistency sets of sentences in which the concept figures, and thus to test for inferences that may be validly drawn from such sets. The specification of a formal semantics to give both truth conditions for the component modalities and conditions defining how the modalities are inter-connected, as in the examples referred to in section 3.2.1, provides the basis for this capacity. In most cases, a proposed formal conceptual analysis will have to show how it copes with a range of test-cases, pertaining to consistency and inference, that have previously been proposed in the literature as benchmarks. To give just two of many possible illustrations here, Delgrande 1988 and Prakken and Sergot 1996 describe benchmarks that challenge the inferential capacities of formal theories of, respectively, default conditionals and so-called 'contrary-to-duty' conditionals.[10]

The third capacity is to articulate specific, characteristic aspects of the concept. This sub-criterion is similar to the second, in that it concerns how a proposed analysis deals with particular problematic test-cases – which may well have been previously described in the literature – but with the primary focus on matters other than consistency and inference. One illustration concerns the *role* concept, for which one of many pertinent questions to ask, of a proposed analysis, is whether it enables expression of the difference between an agent *y*'s performance of (an instance of) an act of type *A* whilst *in* a role *R* to which he belongs, and *y*'s performance of (an instance of) *A* whilst *not in R*. (Consider, for instance, *y*'s giving a warning to *x* in his (*y*'s) role as policeman, and *y*'s performance of the same type of act in his role as *x*'s next-door-neighbour.) A second illustration concerns the concepts of *obligation* and *permission*, and more particularly the fact that they (and their normative relatives) quite commonly occur in rules, regulations and policy-formulations in *nested* sequences. Consider, for example, the sentences "The Head of Department is obliged to permit technical staff to attend relevant training courses", and "The police ought to be permitted to forbid Nazi rallies". If one treats these normative concepts as sentential modal operators, the formal representation of such nested sequences seems to be unproblematic. But they present an interesting challenge to the expressive capacity of a first-order-logic approach in which *obligation* and *permission* are treated as predicates of named actions.

The fourth capacity is to 'place' the concept in relation to its near relatives. For instance, in developing a formal theory of the central normative notions, it is important to articulate (among many other relations), the differences and similarities between permissions and rights, and

[10] Essentially, these are conditional obligations that come into force when some other obligation has been violated.

between permissions and empowerments. And if that theory is embedded in a broader, action-theoretic framework, then it should show how *obligation* is related to, respectively, *ability* and *responsibility*. Furthermore, in the characterization of *communication*, it would be expected that the formal analyses bring out clearly the differences and similarities between different types of communication (*informing*, *ordering*, *requesting*, and so on). As a final illustration, consider again *trust*; it might be suggested that *trust* belongs to a spectrum of concepts, ranging from *full trust*, at one end, to *complete distrust*, at the other, with such notions as *hope*, *uncertainty*, *suspicion* and *fear* falling between them. Would *one* set of basic modal building-blocks be expressively adequate for the task of describing each of the different points along that spectrum?

4.2 Adequacy criteria for Step2-Phase2

In this section we propose a set of adequacy criteria that are relevant to the design and use of *Step2-Phase2* calculi, i.e., formalisms that put primary emphasis on a computational orientation (rather than on a conceptual orientation) with a view to supporting the process of principled operationalization.

The criteria we propose are:

(1) a formal semantics;
(2) a declarative semantics;
(3) expressive capacity;
(4) support for computational tasks;
(5) efficient execution.

A *formal semantics* is sine qua non for a language serving the roles we assign to *Step2-Phase2* calculi. Informal semantics constitute a serious limitation for many applications, where validation and execution traceability are crucial. Without a formal semantics, the *Step2-Phase2* computational framework cannot act as a bridge between the *Step2-Phase1* conceptual representation and the *Step3* system platform implementation. The absence of formal semantics would not allow one to determine precisely the correspondence between the conceptual analysis and the executable specification – for example, we might not be able to determine the extent to which the conceptual analysis is being simplified in order to develop a software implementation.

A *declarative semantics* states *what* is to be computed, not necessarily *how* it is to be computed. Consequently, declarative semantics can be more easily applied to a variety of settings, not just those that satisfy some low-level operational criteria.

A *Step2-Phase2* calculus should additionally have the necessary expressive capacity to represent at least simplified versions of the concepts formalized in *Step2-Phase1*. Ideally, the *Phase2* language will stay 'close' to that of *Phase1*, simplifying as little as possible, and where that is not possible for reasons of computational efficiency, making it clear just what is being simplified, why, and how.

Note that, apart from expressing (possibly simplified versions of) the concepts formalized in Phase1, a Phase2 calculus may often be required to represent features not captured in Phase1. For example, Jones, Sergot and their collaborators employ in a series of papers (Jones and Sergot 1993, 1996, Sergot and Richards 2001, Santos et al. 1997, Santos 2002) a logic of action that stems from work on legal theory – see, for example, (Pörn 1970, Kanger 1972, Pörn 1977). This logic of action includes a relativized (to an agent) monadic action modality E_a – expressions of the form 'E_aF' are read 'agent *a* brings it about that *F*', 'agent *a* sees to it that *F*', or 'agent *a* is responsible for it being the case that *F*'. Central to this logic of action is the concept of *agency*, that is, the state of affairs *F* is caused by or is the result of actions by agent *a*. However, while this level of abstraction makes the approach highly effective for certain explanatory purposes (*Step2-Phase1* analysis), since in some cases it may be desirable to avoid specifying the exact *means by which* a state of affairs was brought about (see, for instance, the exposition of institutionalized power in Jones and Sergot 1996), it has proved resistant to computational application precisely because of that abstraction; that is, the absence from the abstraction of a representation of *state-changes* and *time* is problematic in *Step2-Phase2*. A *Step2-Phase2* calculus, therefore, should have the expressive capacity to represent (possibly simplified versions of) the concepts of a *Step2-Phase1* analysis, *as well as* the necessary features of a computational specification, such as time, even if such features are absent at *Step2-Phase1*.

As regards the criterion of *support for computational tasks*, the main requirements of the computational framework concern narrative understanding/assimilation, i.e. the ability to compute the current state of a system given the (communicative and physical) events that have taken place, as well as proving properties of a computational specification, such as safety, fairness, 'normative consistency', etc.

Narrative understanding, at the very least, is a service offered at run-time, informing, for example, agents about their current institutionalized powers, permissions, obligations, rights and possibly other normative relations; (such a service may be offered in various distributed settings – these considerations are a part of *Step3*). Clearly, run-time services should be offered in real-time, i.e. under certain application-dependent time constraints. In other words, the computational frameworks of *Step2-Phase2* have to support *efficient execution*.

5 Summary and conclusions

This paper has addressed the issue of formalizing concepts from social theory in the design of intelligent socio-technical systems. We reviewed some cases in which the operational model of the concept seemed to rely more heavily on pre-formal intuition than a conceptual model rooted in philosophical, psychological or sociological studies of the concept itself, and discussed the resultant limitations. We proposed instead a method which aims to leverage the best of interdisciplinary research within a consistent, coherent framework, integrating conceptual studies with principled operationalization, i.e. without neglecting the application requirements. To support such a method we need *tools*, primarily formal languages, and we argued that the process of formal characterization requires, in effect, a *toolbox*, depending on the purpose; we presented and discussed specific criteria for evaluating a given tool's fitness-for-purpose.

It has been pointed out to us that historical practice in Computer Science, ordinarily, has been first to provide practical implementations, only later (if ever) formulating underlying theories and models. In this regard, it seems to us, Computer Science has been imitating what previously happened elsewhere in Engineering. Bridges, for instance, were first built in the absence of any comprehensive theory explaining why – in some cases but not all! – the bridges functioned effectively and safely. But just as more reliable, more elaborate bridges were later designed and constructed in ways informed by the relevant physical theories, so – we suggest – could improved socio-technical systems be produced were currently available theories of the fundamental social concepts to be taken seriously in the design process.[11]

The immediate contribution of this paper is therefore the method presented in Section 3. In order to explain our position we have frequently described aspects of our own research, and that of our closest collaborators, since it is that research experience that has led us to reflect on the need for a closer examination of the way things are done. But we emphasize that we used examples drawn from our own research merely for the purpose of illustrating the various steps of the proposed method. It is that approach itself which is the principal contribution of this paper; we are not suggesting that its adoption would require use of just the same sorts of formal tools as we have employed in our own work on designing conceptual and computational frameworks.

Our central thesis does not depend on the actual existence of examples exhibiting application of our proposed method in its entirety; perhaps there are none at all.[12] Nevertheless, we have described examples where parts of the methodology have been adopted, and we offered – in Section 2 – some examples of work (on the development of working systems in which social

[11] In future work, it would be interesting to explore comparisons between the History of Computer Science and the History of (other parts of) Engineering. Furthermore, as one of the reviewers has observed, we might usefully consider discussions of modeling in the Philosophy of Social Science, with a view to developing our position regarding the respective roles of conceptual and computational models.

[12] However, in some interesting comments, Davide Grossi (personal communication) has suggested to us that aspects of the history of BDI in Artificial Intelligence do in fact provide such an example.

concepts have figured prominently) in relation to which we argued that improvements *could have* been gained *had* our proposals been followed.

In conclusion, we would argue that if the methodological proposals made above were to be adopted, there would clearly be consequences for the ways in which researchers need to be trained, not least in the area of Agents and Multi-agent Systems. In particular, they would need to become familiar with techniques corresponding to the two phases of Step2, for the formal modeling of relevant concepts, and for the design of computational frameworks, together with an understanding of research examples in which those two types of model interact in appropriate ways. This would require not only a depth of knowledge of Artificial Intelligence, but also an appreciation of the essential role of interdisciplinary enquiry. It takes time to develop such skills, so it would be necessary to re-think the ways in which PhD programmes in this area should be structured and funded.

Acknowledgements The preparation of this paper benefited from the comments of Gunnar Coll, Robert Demolombe, Davide Grossi and the late Steve Barker. We thank also the reviewers for their valuable insights and suggestions.

References

Akman, V., Erdogan, S., Lee, J., Lifschitz, V., and Turner, H. (2004), "Representing the Zoo World and the Traffic World in the language of the Causal Calculator," Artificial Intelligence, 153(1-2), 105-140.

Allen, L. (1957), "A razor-edged tool for drafting and interpreting legal documents," The Yale Law Journal, 66, 833-879.

Andrews, P., Polack, F., Sampson, A., Stepney, S., and Timmis. (2010), "The CoSMoS Process, Version 0.1: A Process for the Modelling and Simulation of Complex Systems," echnical Report YS-2010-453, University of York.

Artikis, A., and Sergot, M. (2010), "Executable Specification of Open Multi-Agent Systems," Logic Journal of the IGPL, 18(1), 31-65.

Artikis, A., Sergot, M., and Pitt, J. (2007), "Executable Specification of a Formal Argumentation Protocol," Artificial Intelligence, 171(10-15), 776-804.

Brewka, G. (2001), "Dynamic Argument Systems: a Formal Model of Argumentation Processes Based on Situation Calculus, Journal of Logic and Computation, 11(2), 257-282.

Castelfranchi, C., and Falcone, R. (1998), "Social Trust," in Proceedings of the first workshop on deception, fraud and trust in agent societies, pp. 35-49.

Castelfranchi, C., and Tan, Y.H. (eds.), (2001), Trust and Deception in Virtual Societies, Kluwer, Dordrecht, Holland.

Chopra, A., and Singh, M. (2006), "Contextualising commitment protocols," in Proceedings of the Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), ACM, pp.1345-1352.

Craven, R., and Sergot, M. (2008), "Agent strands in the action language nC+," Journal of Applied Logic, 6(2), 172-191.

Delgrande, J. (1988), "An approach to default reasoning based on a first-order conditional logic: revised report," Artificial Intelligence, 62-90.

Dennett, D. (1995), Darwin's Dangerous Idea: Evolution and the Meanings of Life, London: Penguin Books.

Edmonds, B., Gilbert, N., Gustafson, S., Hales, D., and Krasnogor, N. (eds.), Socially Inspired Computing. Proceedings of the Joint Symposium on Socially Inspired Computing, AISB (2005).

FIPA (2002), "FIPA Communicative Act Library Specification," http://www.fipa.org/specs/fipa00037/index.html.

Fornara, N., and Colombetti, M. (2009), "Formal specification of artificial institutions using the Event Calculus," Multi-Agent Systems: Semantics and Dynamics of Organizational Models, IGI Global.

Fox, M., Barbuceanu, M., Grüninger, M., and Lin, J. (1998), "An Organizational Ontology for Enterprise Modeling," in Simulating Organizations: Computational Models for Institutions and Groups, eds M. Prietula, K. Carley and L. Gasser, AAAI Press/MIT Press, pp. 131-152.

Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., and Turner, H. (2004), “Nonmonotonic causal theories,” Artificial Intelligence, 153(1-2), 49-104.

Grossi, D., and Jones, A. (in press), “Constitutive Norms and Counts-as Conditionals,” in D. Gabbay, J. Horty, R. van der Meyden and L. van der Torre, eds., *Handbook on Logic of Normative Systems*, vol. 1, College Publications, UK.

M., Grüninger, M., and Fox, M. (1994), “The Role of Competency Questions in Enterprise Engineering,” in Proceedings of the IFIP WG5.7 Workshop on Benchmarking-Theory and Practice.

Herzig, A., Lorini, E., Hübner, J. F., Ben-Naim, J., Castelfranchi, C., Demolombe, R., Longin, D., and Vercouter, L. (2008), “Prolegomena for a Logic of Trust and Reputation,” in Proceedings of NORMAS, pp. 143-157.

Herzig, A., Lorini, E., Hübner, J. F., and Vercouter, L. (2010), “A Logic of Trust and Reputation,” Logic Journal of the IGPL, 18(1), 214-244.

Hohfeld, W. (1913), “Some Fundamental Legal Conceptions as Applied in Judicial Reasoning,” Yale Law Journal, 23(16).

Jones, A., and Sergot, M. (1993), “On the Characterization of Law and Computer Systems: The Normative Systems Perspective,” in Deontic Logic in Computer Science eds. J.-J. Meyer and R. Wieringa, Chichester: John Wiley and Sons.

Jones, A., and Sergot, M. (1996), “A formal characterization of institutionalised power,” Journal of the IGPL, 4(3), 429-445.

Jones, A., Pitt, J., and Artikis, A. (2011) “On the Analysis and Implementation of Normative Systems – Towards a Methodology,” in Proceedings of the Workshop on Coordination, Organisation, Institutions and Norms (COIN), at the 10th International Conference on Autonomous Agents and Multi-agent Systems, Taipei, Taiwan, pp. 47-56.

Kanger, S. (1957), New Foundations for Ethical Theory, University of Stockholm, Department of Philosophy. Also in R. Hilpinen (ed.), Deontic Logic: Introductory and Systematic Readings, Dordrecht: Reidel, 1971.

Kanger, S. (1972), “Law and Logic,” Theoria 38, 105-132.

Kanger, S., and Kanger, H. (1966), “Rights and Parliamentarism,” Theoria 32, 85-115.

Kowalski, R., and Sergot, M. (1986), “A Logic-Based Calculus of Events,” New Generation Computing, 4(1), 67-96.

Levesque, H., Pirri, F., and Reiter, R. (1998), “Foundations for the Situation Calculus,” Linköping Electronic Articles in Computer and Information Science, 3.

Lewis, D. (1969), Convention – A Philosophical Study, Harvard University Press, Cambridge, Mass., USA.

Lindahl, L. (1977), Position and Change: A Study in Law and Logic, Dordrecht: Reidel.

Marín, R., and Sartor, G. (1999), “Time and Norms; a Formalisation in the Event Calculus,” in Proceedings of the Conference on Artificial Intelligence and Law (ICAIL), ACM Press, pp. 90-100.

McCarthy, J. (1963), “A Basis for a Mathematical Theory of Computation,” in Computer Programming and Formal Systems, P. Braffort and D. Hirschberg, (eds.), Amsterdam: North-Holland, 33-70.

Oren, N., Luck, M., and Miles, S. (2010), “A Model of Normative Power,” in Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 815-822.

Pinto, J., and Reiter, R. (1993), “Temporal Reasoning in Logic Programming: a case for the Situation Calculus,” in Proceedings of Conference on Logic Programming, D. Warren, (ed.), Cambridge, Mass.: MIT Press, 203-221.

Pinyol, I., and Sabater-Mir, J. (2011), “Computational trust and reputation models for open multi-agent systems: a review,” Artificial Intelligence Review (Springer Online-First).

Pörn, I. (1970), The Logic of Power, Oxford: Blackwell.

Pörn, I. (1977), Action Theory and Social Science – Some Formal Models, Synthese Library vol. 120, Reidel, Dordrecht, Holland.

Prakken, H., and Sergot, M. (1996), “Contrary-to-duty obligations,” Studia Logica 57, 91-115.

Reiter, R. (1993), “Proving Properties of States in the Situation Calculus,” Artificial Intelligence, 64, 337-351.

Sandhu, R., Ferraiolo, D., and Kuhn, R. (2000), “The NIST Model for Role-Based Action Control: Toward a Unified Standard,” in the 5th ACM Workshop on Role-Based Access Control, RAC ’00, 47-63.

Santos, F. (2002), “A Modal Logic Framework for rganization Analysis and Design,” in Proceedings of the Workshop on Deontic Logic in Computer Science (DEON), eds. J. Horty and A. Jones, pp. 279-299.

Santos, F., Jones, A., and Carmo, J. (1997), “Action Concepts for Describing Organized Interaction,” in HICCS ’97: Proceedings of the 30th Hawaii Conference on System Sciences, ed. R. A. Sprague, IEEE Computer Society, pp. 373-382.

Sergot, M. (2008), “Action and Agency in Norm-Governed Multi-Agent Systems,” in Proceedings of ESAW VIII, A. Artikis, G. O’Hare, K. Stathis and G. Vouros, (eds.), LNAI 4995, Springer, 1-54.

Sergot, M., and Craven, R. (2006), “The Deontic Component of Action Language nC+,” in Deontic Logic in Computer Science (DEON’06), L. Goble and J.-J. Meyer, (eds.), LNAI 4048, Springer, 222-237.

Sergot, M., and Richards, F. (2001), “On the Representation of Action and Agency in the Theory of Normative Positions,” Fundamenta Informaticae 48(2-3), 273-293.

Skyrms, B. (2010), Signals: Evolution, Learning and Information, Oxford, UK: Oxford University Press.

Steels, L., and Brooks, R. (1994), The Artificial Life Route to Artificial Intelligence: Building Situated Embodied Agents, New Haven: Lawrence Erlbaum Ass.

Yolum, P., and Singh, M. (2004), “Reasoning About Commitments in the Event Calculus: An Approach for Specifying and Executing Protocols,” Annals of Mathematics and Artificial Intelligence 42(1-3), 227-253.