Softening electronic institutions to support natural interaction

Citation for published version:
Murray-Rust, D, Papapanagiotou, P & Robertson, D 2016, 'Softening electronic institutions to support natural interaction' Human Computation, vol. 2, no. 2. DOI: 10.15346/hc.v2i2.3

Digital Object Identifier (DOI):
10.15346/hc.v2i2.3

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Human Computation

Human Computation (2015) 2:2:155-188

ISSN: 2330-8001, DOI: 10.15346/hc.v2i2.3

Softening electronic institutions to support natural interaction

DAVE MURRAY-RUST, UNIVERSITY OF EDINBURGH

PETROS PAPAPANAGIOTOU, UNIVERSITY OF EDINBURGH

DAVE ROBERTSON, UNIVERSITY OF EDINBURGH

ABSTRACT

A large amount of human interaction is prosecuted along the rambunctious pathways of social networks, producing vibrant and chaotic streams of communication. Utterances are channelled along ever more complex pathways to an increasingly fuzzily defined audience of humans and machines. These receivers are eager to find meaning and structure, to coordinate and support social endeavours. However, extracting interactional structures from these outpourings is a complex task, as we deal with overlapping conversations, between many actors, spread across multiple networks.

A well developed method of coordinating activity exists in the form of Electronic Institutions (EI). These institutions provide an architecture which allows agents to carry out complex patterns of interaction, based on shared protocols, while assuming little knowledge about their compatriots, and providing guarantees about the outcomes of interactions. While EIs are a powerful tool for coordinating computational agents, they are less widely used to support human activity. A key reason for this is what one must give up in order to join an EI: one must first understand the language used to define the protocols, and then commit to carrying out interactions through the machinery of the EI, sacrificing control and leaving the openness of the interconnected online world. These barriers to entry have meant that traditional EIs have not become relevant to the vast surge of data, or the potential for interaction centred around socially driven systems such as Twitter.

Here, we connect the power of EIs to describe and formalise interaction with the open social systems which are currently supporting such a wide range of human interaction. This allows for the modelling of behaviour, to extract patterns of interest from data for summarisation and exploration. It also provides a framework by which computational intelligence can be harnessed in support of informal human interaction.

The cost of making this connection is the creation of an additional layer which *binds* freeform interaction streams into appropriate hooks and levers within EIs, matching loose social discourse with crisp institutional structures. While the general case matching utterances to formal semantics is extremely difficult, the presence of an interaction protocol allows us to concentrate on only the possible actions which would make sense for the institution in its current state, reducing the range of possibilities and simplifying the task of translation such that simple approaches give sufficient discriminatory power. This is not appropriate for every situation, but for well chosen models of interaction, a few matching rules can be enough to create a formal skeleton *alongside* free discourse.

In this paper, we describe how this translation can be carried out, starting from an account of how institutions can still have power when they are no longer the gatekeepers of action. We detail the formal machinery necessary, and describe our implementation using a process calculus, with examples.

1. INTRODUCTION

Our paper describes "shadow institutions", a mechanism for combining electronic institutions with self-organised online interaction. This allows access to these new areas of engagement at the cost of supporting this new form of "softer" institution.

Just as human institutions provide a high level of support for collaborative activity, electronic institutions support collaboration amongst computational agents. However, this comes at the cost of requiring the interaction to be formalised, and participants to submit themselves to the institutions control, which is not appropriate for the bulk of human interaction online.

In the broader context of creating environments for human interaction, Alexander (1975) maintained that planning in a completely top down manner does not work. Rather, one should observe the traces people leave as they go about their lives, and create infrastructure which follows these *desire lines*, a notion which is being taken up in human computed interaction (Myhill, 2004). Similarly, (Parikh, 2002) uses "social software" to describe the informal notions of process which communities have: norms arise, gradually become explicit, are encoded textually and eventually implemented in software. This incremental formalism allows the community greater flexibility about their operation, and allows parts of the interaction which are not suitable for formalisation to be left unconstrained.

Instead of creating formal institutions for people to adapt to their use, shadow institutions are built on top of existing infrastructure—living spaces, already populated with interacting people, with paths well worn through social convention. The permanence of online interaction makes it possible to follow the digital traces of thousands of humans going about their business and decide which parts would benefit from some assistance. This leverages the effort social networks have put into creating convivial, community spaces (Illich, 1973; Lamizet, 2004; Caire, 2009).

In summary, the principle is to create the most lightweight, most open institutional framework which still brings the benefits of formalisation and computational intelligence to bear on the interaction at hand. By relying on constitutive rather than regulative mechanisms (Section 5), our shadow institutions can be incrementally applied to existing, semi-structured interactions.

In order to do this, we use LSC, a process calculus (Murray-Rust and Robertson, 2014; Robertson, 2004) to define protocols for interaction. This is a departure from traditional EIs such as ISLANDER (Esteva et al., 2002) that use *scenes* as the basic metaphor for institution design, with agents moving between scenes and taking different roles based on constraints on their behaviours. LSC, despite having equivalent generality and power to ISLANDER[1], does not require as much structuring at a scene level, instead relying simply on

[1] As evidenced by the existence of translators from the parent language LCC to ISLANDER specifications

having roles which the actors may play, allowing state to be distributed among the actors without a central authority. This decentralisation of state is a crucial feature which allows the institution to work without taking complete control of the actions allowable for each participant. Instead, the institution matches its structures to the observed outputs, *following along* with what people are doing naturally. Producing matched structures can be used summarise and classify human behaviour, but the institution can also contribute to the interaction, nudging the participants, feeding back the results of calculations and importing knowledge from external sources.

This way of thinking about institutions expands the range of situations in which they can be applied, providing a route to engagement with human communities at scale. However, there is a necessary change in approach to engineering such institutions: rather than assuming that every possible action must be accounted for, interaction models are chosen to represent only that which is of interest, allowing other events to pass untranslated. Since there is little control over the actions of the participants, protocols and supporting mechanics must be chosen so as to reduce ambiguity over which events are of interest and how they should be interpreted.

The rest of this paper explores the machinery necessary to create lightweight, contextually appropriate, ‘soft’ institutions, which pick out the relevant utterances in a stream of discourse to create a formal skeleton on which to hang computational intelligence.

The paper is structured as follows. Section 2 describes uses of Electronic Institutions (EIs), and sets out a model of online activity on which our analysis is built, while Section 4 gives a definition of EIs, and artifacts which they can use to engage with a stream of events. Section 5 discusses the means by which institutions derive their power. Section 6 lays out our proposal for *shadow institutions*, and Section 7 illustrates the construction of an SI. Further sections discuss the implications of the proposal and its relation to other work.

1.1. Example Applications

The application domain for our work is social situations where adding some lightweight computational support can help with coordination, planning or other aspects of a shared endeavour. We describe three interactions for which we have built functional shadow institutions (although no field trials are reported here).

Organising group events: using unstructured media to organise events can be a headache: as email threads get longer, it becomes difficult to know who is currently involved, what their preferences or constraints are and so on. Supporting this is described more fully in Section 7, but the general intent is to create an institution which listens in to the discussion, keeping track of participants and their preferences, and which can be called upon to summarise, aggregate or optimise group choices.

Taxi Sharing: in the UK, social norms prevent people in taxi queues from enquiring about adjacent destinations in order to make more efficient use of the taxi infrastructure. At transport hubs such as airports, where many people are trying to get a taxi at the same time, sharing taxis is economically efficient, and reduces waiting times. The institution is set up to be almost zero barrier to entry—anyone can tweet where they are going, using a hashtag for the point of departure, and if there is someone else going somewhere nearby, the institution will notify them both. The only additional infrastructure needed on site is some way to discover the hashtag, such as stickers or posters near the taxi rank. This is in contrast to more elaborate ride-sharing sites, which require signing up or registering, and typically have to provide reputation metrics or other safeguards, rather than relying on existing mechanics.

Distributed hypothesis testing: making the assumption that people are carrying out lifelogging, it would be useful to be able to test hypotheses about correlations between different variables, such as exercise and mood, in a way which does not require users to share every aspect of their lift. To this end, we have an interaction model which integrates access to a personal data store (INDX) with the shadow institution. Users can submit hypotheses using Twitter, and their followers can discover the test and decide if they are happy for their data to be user. The institution then passes queries to be executed to the participants data stores, aggregates the responses and sends out the results. This interaction illustrates the use of institutions in interactions to perform trusted functions such as aggregation and anonymisation of personally sensitive data.

2. BACKGROUND

2.1. Uses of Electronic Institutions

The inspiration for electronic institutions in multi-agent systems comes from the observation that human social systems gain coherence through observing social norms (Esteva et al., 2001; Dignum, 2002; García-Camino et al., 2005). These are rules and conventions that are accepted by a given social group and upon which membership of the group depends. This, and other incentives applied when interacting within the group, ensures that the social norms of an agent group are upheld. The aim is to define conventions for the actors to follow, supporting cooperation and reducing uncertainty (North, 1990, p.4)(Savarimuthu and Cranefield, 2009). This, in turn, is intended to make perennial issues of communication and coordination more manageable, particularly in open systems where we cannot standardize individual agents (Aldewereld et al., 2007; Luck et al., 2013; Artikis and Pitt, 2009).

The means of engagement between agents and electronic institutions is normally by subscribing, in some way, to the protocol for interaction. Depending on the system, this can happen in different ways: by downloading the protocol from within an execution environment (Islander Esteva et al., 2002); by discovering the protocol on a peer-to-peer network (Siebes et al., 2007; Robertson et al., 2008); or by accessing the protocol via a Web service (De Roure et al., 2010). Regardless of the means of subscription, agents always make a decision to engage with these systems. For example, to be part of an on-line auction and agent might subscribe to a protocol for a particular style of auction and, as a consequence, commit to the social norms of that protocol. This sort of up-front commitment is convenient for artificial agent societies but is not natural for many human social contexts.

2.2. Humans and EIs

Participation in an electronic institution can be difficult for humans, as their actions are then mediated by the institution. In order for a human to join in with an interaction, they must:

i. discover that the interaction is taking place;

ii. commit to taking part in the interaction as specified;

iii. have a facility with the institutional language and infrastructure;

iv. fit their utterances to the current state of the interaction, either through understanding the protocol and the current state, or through being offered choices of possible actions by some interface to the institution.

Any kind of infrastructural engagement is a barrier to entry; users are bound to their convivial networks of friends and habits, and are reluctant to move to new websites for interaction, let alone download and install architectures or even desktop programs. This issue increases as the number of people involved increases—the group cost of commitment quickly becomes unreasonably high.

Similary, forcing interaction to follow a model requires a commitment from people to the particular model, and does not allow flexibility. This is a strong constraint on their behaviour, which is not necessary in every situation. Additionally, participants must spend time and effort on understanding the model in order to match their desires to possible institutional pathways. This is a cognitive overhead, requiring conformance of human behaviour to computational systems.

2.3. Online Interaction

The commitment and formalisation demmanded by EIs is at odds with existing human social networking systems. In social networks, a wide variety of social norms are in play constantly, and humans are adept at identifying, combining and adapting to these. In contrast to traditional broadcast or many-to-one models (Walther et al., 2010), many fluidly defined channels of communication exist (Wu et al., 2011). As an example, on Twitter, messages can be sent directly to a single person, publicly but flagged for the attention of certain people, or to a particular ad-hoc public denoted by a ‘hashtag’(Bruns and Burgess, 2011). The environment is noisy; messages in a single interaction are often non-adjacent (Honeycutt and Herring, 2009); and conventions around structure are constantly emerging (Boyd et al., 2010).

Here, we are interested in situations where *some* events within the global stream can be analysed in terms of interaction protocols (Figure 1). We consider a spectrum of formality that has at one end completely formalisable interactions, where protocols can perfectly cover the actions humans naturally take, and at the other situations where interaction is essentially schema-less, with no underlying models which can explain the relations of events to each other.

Reality lies between these extremes. For example, recent work on Twitter has shown how a portion of utterances can be modelled in terms of speech acts (Zhang et al., 2013) or conversational structures (Ritter et al., 2010). However, the proportion of events which surrender meaning to this analysis is small—37% of tweets are ‘conversational’ according to a 2009 study (Risman, 2009). Many interactions are single utterances, or have such simple structures that adding any kind of dialogical framework on top of them brings little enlightenment.

There are increasing numbers of communities and applications which build islands of structure in the stream of interaction. As two examples, WeDo runs an ideation and voting protocol over Twitter (Zhang et al., 2014) and `#icanhazpdf` is a community built around a hashtag for exchanging scholarly literature, with clear behavioural guidelines (Liu, 2013).

Not all events will be covered by a protocol, so it is not possible to model the entire event stream as a state machine, and it is not possible to strongly constrain user behaviour, but there is some desire for the kind of structured interactions supported by an electronic institution. The approach outlined here is aimed at creating a low barrier to entry interaction model which can be flexibly applied to online discourse in order to draw on the benefits of computational intelligence.

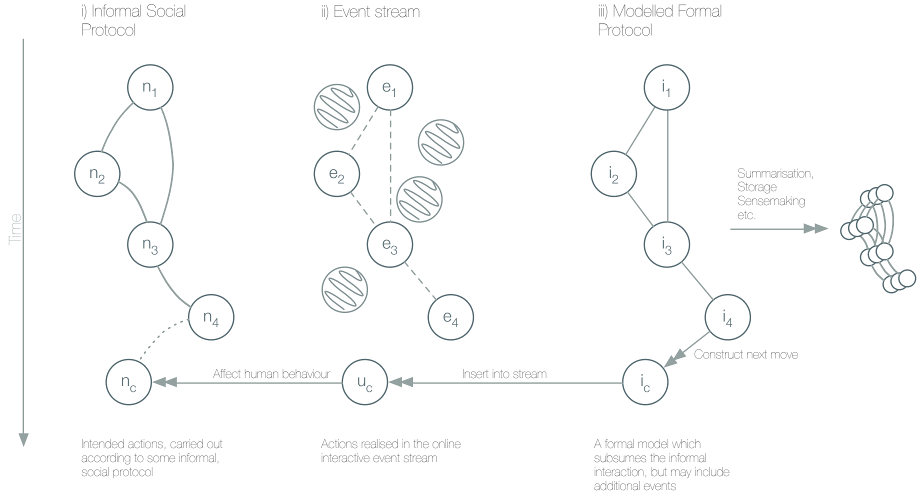

i) Informal Social Protocol
ii) Event stream
iii) Modelled Formal Protocol
Time
n_1
n_2
n_3
n_4
n_c
e_1
e_2
e_3
e_4
u_c
i_1
i_2
i_3
i_4
i_c
Summarisation, Storage Sensemaking etc.
Construct next move
Insert into stream
Affect human behaviour
Intended actions, carried out according to some informal, social protocol
Actions realised in the online interactive event stream
A formal model which subsumes the informal interaction, but may include additional events

***Figure 1.* Modelling interaction streams.**
i) Informal or semi-formal social protocols cover some portion of human interaction, and it's these that we're interested in. ii) The desired actions are expressed as utterances by the participants, in the interaction stream. There are other messages in there, and the structure are only partially specified. iii) A formal protocol then mirrors the social protocol, allowing computational agents to come up with the next move within a superset of the social protocol, affecting change in the human behaviour.

3. RELATED WORK

Shadow institutions seek to soften EIs in order to allow them to be used in the the less structured environments where people spend much of their time. Here we review three separate approaches to achieve similar ends: i) allowing humans participatory access to EIs; ii) human oriented structures with similar constructions such as BPMN or BPEL iii) other systems currently used which achieve the kinds of coordination provided by EIs.

3.1. Interfaces to Electronic Institutions

One approach is to make EIs more accessible to humans by providing interfaces which they can use. Campos et al. (2009) discusses tools which can support decision making in EIs, and Esteva et al. (2011) looks at using minimal lightweight infrastructure with assistance to create hybrid human/agent MAS. Bogdanovych used the term *virtual institution* (VI) to refer to electronic institutions embedded in 3D virtual worlds (Bogdanovych, 2007). By creating a ‘normative environment that offers immersive experience’, virtual institutions support social interaction and collaboration among other qualities. Early uses of VIs in E-Commerce (Bogdanovych et al., 2005) created a new 3D world for people to interact in, separate from their existing online habitus, although subsequent development used Second Life as the infrastructure, and developed the idea of a *causal connection layer* between the virtual world and the institution (Trescak et al., 2013). VIs have also been integrated into the fabric of game worlds for Massively Multiplayer Online Games (Aranda et al., 2012). Here the VI is used to set up scenes and conditions for transitions between them to define role-based interactions; this can then be used to e.g. set up quests for players to carry out within the gameworld. These are compelling examples of the utility of EIs, as there is a clear correspondance between the needs of game designers and the properties of EIs. They also share many components with shadow institutions: there are *avatar agents* which represent humans, a performative structure and some connection to a world outside the institution. Ananda et. al suggest that: "[s]pecifically, those actions that require institutional verification are those mapped to scene messages. The rest of the actions provided by the virtual world software can be freely executed."—(Aranda et al., 2011, p.194). This is similar to the idea that only the parts of the interaction which benefit from formalisation need be formalised. However there are two key differences: firstly, the locutions which make up the performative structure are in terms of the bounded ontology of items, locations and agents within the gameworld, so no translation or integration is needed. Secondly, the institution is still in a position of control over what happens in the world—the institutional fact that "X has not completed quest Y" means that the door to the lair of the space pirates remains resolutely resistant to ingress.

de Jonge et al. (2013) develop web-based GUIs to EIs. This has a similar goal to our shadow institution, of allowing institutions to become part of online interaction. The construction of the interface layer is rooted in particpatory design, allowing for the deployment of incomplete institutions with iterative testing and development, in order to be responsive to community needs. The technologies used are also appropriate—RESTful interfaces and HTML5—to ensure that there is no barrier to entry. Again, the difference is the the EIs in MusicCircle are the arbiters of action, and have a regulatory power that shadow institutions do not. Additionally, interaction happens on infrastructure which is driven by the institution rather than on existing infrastructure with its attendant population and practice.

A similar approach is taken in the Dialogue Game Execution Platform (Bex et al., 2014): a formal model of the dialogical structure of an argument—of similar structure to an EI's protocol—is used to drive a web interface which can be used to carry out argumentation with participant's responses constrained by the protocol. The strong semantics encoded in the protocol then allow the extraction and visualisation of the resulting argument.

Finally, there are strong similarities with the A+A model of agents and artifacts (Omicini et al., 2006, 2008). In particular, the use of artifacts to bridge the divide between the physical and digital worlds (Castelfranchi and Piunti, 2012) informed the artifactual presentation of constitutive rules outlined here.

3.2. Process workflows and online coordination

Workflow modelling employs a different point of view to (human and computer) agent coordination. In this, agents are viewed as *processes* that achieve tasks or goals. Workflow models are used to describe the control and information flow between these processes, which is commonly represented diagrammatically. The workflow structure defines the appropriate *process choreography*. Within this context, workflows can be viewed as parallel constructs to EIs.

Modelling and managing process workflows is the primary focus of Business Process Modelling/Management (BPM) (Williams, 1967). BPM has been used as an industrial scale business management approach by a rapidly increasing number of companies over the past 15 years. Its practices aim to increase business effectiveness, efficiency, flexibility, and integration with technology, by automating the coordination between human and computer systems. It is most commonly seen within the context of Enterprise Architecture (EA) (Ross et al., 2006) and as part of business optimisation.

There are two core languages that are most commonly used in the context of BPM, namely the Business Process Model and Notation (BPMN) (Object Management Group, 2011) and the Business Process Execution Language (BPEL) (OASIS, 2007):

- BPMN is the most commonly used language for the design of business process workflows based on a large variety of concepts. This variety maximizes the flexibility when modelling different business scenarios. The existing graphical notation makes BPMN models accessible and understandable by both technical developers and business analysts, therefore helping towards bridging the communication gap between the two.
- BPEL is an execution language used to map abstract business processes to concrete implementations. Its aim is to provide concrete, executable models of abstract workflows based on web services standards and tackle the various implementation challenges.

In both BPMN and BPEL, human stakeholders are modelled as roles that are responsible for a number of processes. Swimlane notation is commonly used to clearly separate the different roles. Similarly to EIs, the workflow-based coordination between the various human and computer components is strict and regulatory. Even though BPMN is general enough to model social interactions (Brambilla et al., 2012), the corresponding structures remain rigid and require a formal basis of interaction through specifically deployed platforms.

3.3. Other online coordination

The recognition of the importance of collective intelligence (Malone et al., 2009; Bernstein et al., 2012; Malone et al., 2010) has led to an increasing number of systems which support collective action. Many systems support some form of crowdsourcing or collective working (Ahmad et al., 2011; Franklin et al., 2011; Schall et al., 2012), however these systems tend to lack the mixed initiative and open ended structure which EIs make possible.

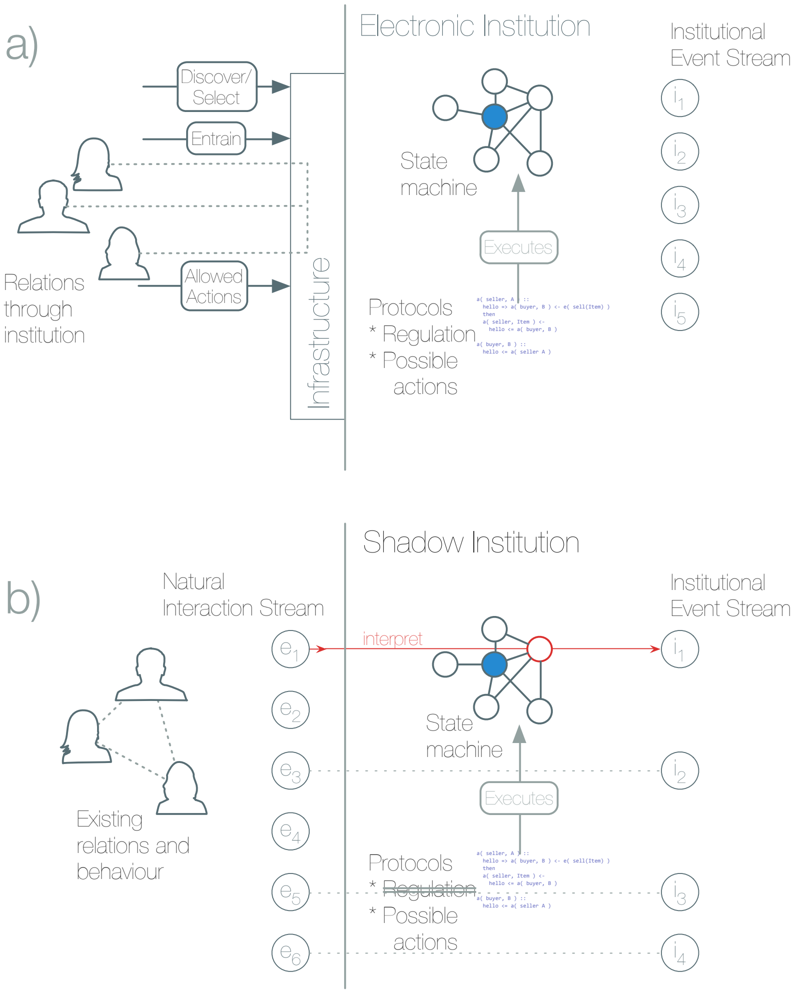
Possibly the most similar system is WeDo (Zhang et al., 2014). which supports ideation[2] and execution of collective actions. Participants suggest ideas by tweeting, and then the favouriting and retweeting infrastructure built into Twitter is used for voting and decision making. This perfectly captures the lightweight approach to interaction which shadow institutions are aimed at, but lacks the flexibility of a protocol based system for specifying patterns of interaction.

Finally, work is progressing on identifying conversational and interactive structures on Twitter. The forthcoming E^2 platform is seeking to generate argumentation based structures from user opinions in order to present summaries to decision makers (Chesñevar and Maguitman, 2013). Finally, being able to i) extract dialogue structures through unsupervised induction (Ritter et al., 2010) and ii) to annotate tweets with performative labels (Zhang et al., 2013) demonstrate the plausibility of extracting rich, meaningful interaction data to inform institutional behaviour.

4. ARTIFACTS AND INSTITUTIONS

Engineering of electronic institutions is deeply rooted in formal specification of the framework for interaction (Sierra et al., 2004; D'Inverno et al., 2012). A popular style of specification is by means of an explicit protocol, crafted to enforce a set of social norms. This protocol can be inspected by an agent as a means of deciding whether to join a particular social group. The protocol also acts as a guide for the

[2] The generation, development and communication of ideas



```
a( seller, A ) ::
  hello => a( buyer, B ) <- e( sell(Item) )
  then
  a( seller, Item ) <-
    hello <= a( buyer, B )

a( buyer, B ) ::
  hello <= a( seller A )
```

```
a( seller, A ) ::
  hello => a( buyer, B ) <- e( sell(Item) )
  then
  a( seller, Item ) <-
    hello <= a( buyer, B )

a( buyer, B ) ::
  hello <= a( seller A )
```

***Figure 2.* Diagram illustrating moving the state machine outside the interaction stream. In a), people interact directly with the electronic institution; they must understand and obey its protocols, and their allowed utterances at any point are constrained by the state machine. In b) people interact without constraint, and the institution interprets and processes those utterances it can according to the protocols it understands.**

agents to follow once they have joined a social group. Crucially, the scope of such protocols is limited to communication and coordination between agents. This simplifies the task of specification because most of the detail of individual agent reasoning is encapsulated by decision procedures external to the protocol. This sort of protocol-based approach to agent communication and coordination has been adopted in a variety of formal systems, for example in the Islander system using finite-state automata (Esteva, 2003) or in the LCC process calculus (Robertson, 2004, 2012).

4.1. Electronic Institution Definition

For conciseness, here we use a simplified, abstracted definition of an EI[3], as seen in Figure 3. Our simplified electronic institution is defined in terms of two languages:

– A language for representing the *domain* of discourse, which we will refer to as *AL*
– Languages for the *attributes* of agents and institutional state, *constraints* on actions and *updates* to institutional state which occur as a result, collectively termed *IL* here.

In what follows, we use the LSC process calculus (Murray-Rust and Robertson (2014), based on Robertson (2004, 2012)) for our formal specifications but our arguments are independent of this choice of specification language.

The following core components are defined:

Agents are going to engage with the institution dialogically, using it to exchange messages with other agents. The internal behaviour of agents is not specified, but the institution represents facts about the agents, e.g. their credit or their current position within an interaction (*AgState*) and updates this state based on the messages which they send.

Protocols contain statements in *IL* and *AL*, covering the ways in which agents can engage with the institution: the roles they can take, transitions between states, the order in which actions may be carried out, and the effects of actions on the institutional state and the agents' states.

Interactions are instantiations of protocols, with actual agent states. As the interaction progresses, agents exchange messages and their state transitions according to the protocol. This combination of agent states, along with any associated institutional state represents the interaction at a given point in time.

Institutions are a set of protocols representing the interactions which the institution knows how to carry out, and a set of interactions which are currently running, along with their associated state.

This roughly describes a traditional EI: an agent can discover an institution, and either select a protocol to use for an interaction, or join an existing interaction. This engagement is carried out by selecting a protocol for the interaction, creating any state necessary, and then using the protocol and state to define its next set of possible moves. As this activity progresses, the institution updates roles and knowledge about agents to open up new possibilities for actions and state transition.

The institution works as a state machine, and controls what actions are possible according to the current state. Agents are constrained in their actions by the options offered by the institution according to the the protocol being executed.

4.2. Simple Artifacts for Electronic Institutions

In many cases, the effects of institutional actions are contained within the institution's state. However, there is often a need to interact with something outside this state— for instance a human, or some computational resource. A means to represent and contain non-local state is the use of *artifacts* to provide an environment for the agents to operate within (Omicini et al., 2006, 2008) by encapsulating state and providing a visible

[3] A rough mapping to the schema in D'Inverno et al. (2012) is as follows: *AL* is the *domain* language; *IL* is a combination of *attribute*, *constraint* and *update* languages; Interactions are roughly equivalent to *scenes*, *roles* are roughly equivalent, although LSC roles carry information as well as labels to reflect the decentralised nature of the language.

$$\begin{aligned}
Institution &= \langle P*, Interaction* \rangle \\
Interaction &= \langle P, AgState* \rangle \\
Ag &= \langle aid \rangle \\
AgState &= \langle aid, Role, IL* \rangle
\end{aligned}$$

- *AL* is some language in which the domain of the interaction is described, while *IL* is a language used to specify the mechanics of institutions. *Role* and *Pt* are both part of *IL*.
- *P* is a set of statements in *IL*, defining a particular interaction.
- the Institution contains a set of protocols, and interactions which are running according to those protocols. Each interaction is defined by a protocol and a set of agent states representing the position of the agent in the interaction, in terms of a role and some statements in *IL*.
- Agents are separated from their state here—the only constraint we put on them is that they have an ID which can be used to find their state.

Figure 3. Specification of Electronic Institutions.

$$\begin{aligned}
Institution &= \langle P*, Interaction*, IArt* \rangle \\
Interaction &= \langle P, AgState*, Art* \rangle \\
Ag &= \langle aid, Art* \rangle \\
AgState &= \langle aid, Role, IL* \rangle \\
Art &= \langle E \rightarrow bool, elicit*, toRole*, Diss* \rangle \\
IArt &= \langle E \rightarrow bool, toInteraction* \rangle
\end{aligned}$$

Bindings:

$$\begin{aligned}
toInteraction &= E \nrightarrow Interaction \\
elicit &= E \nrightarrow (aid, Role, AL*) \\
toRole &= E \nrightarrow (aid, Role, AL*) \\
Diss &= k(aid, AL) \nrightarrow E
\end{aligned}$$

Additions relative to Figure 3 deal with the addition of communication artifacts that map events on the social network (E) into formal structures. These use artifact predicates (Section 4.2) as constitutive rules (Section 6.1). Matching functions ($E \rightarrow bool$) are used select a subset of all events for consideration—for example, following a particular hashtag on twitter, or watching messages sent to a certain account. Constitutive rules are represented as partial functions from events to institutional structure ($E \nrightarrow X$). Within this

- *Art* is a communications artifact. Agents have communications artifacts (*Art*) which they use to follow the activities of a human. Each communications artifact:
  - filters a stream of utterances to produce as set for consideration ($E \rightarrow bool$)
  - uses constitutive rules to turn them into statements in institutional terms attributed to an agent, i.e. the result of *elicit* is used to satisfy $e()$ or the result of *toRole* gives the agent a new role and associated knowledge in the interaction.
  - Maps institutional facts onto utterances (*Diss*) to relate the state of the institution back to the participants.
- Institutional artifacts (*IArt*) are similar, but contain rules that map utterances onto interaction definitions: a protocol to use and some starting agent states (*toInteraction*).

Figure 4. Specification of shadow institutions.

interface for action. Using this formulation, everything in the institution can be assumed side-effect free, unless it explicitly interacts with an *artifact*, which is then responsible for containing the state that has changed.

Here, we take a particular view on artifacts and their use. We assume that any agent in the system has access to a set of artifacts which define their computational environment outside of that provided by the institutional machinery. This could include semi-permanent state which the agent uses across multiple interactions, access to shared state with other agents, or drawing on computational resources which are not appropriately defined in institutional terms such as databases or optimisation algorithms. These artifacts are wired into the protocols by assuming that the institutional language (*IL*) contains three special predicates: *e*,*k* and *i*. These are intended as general purpose mechanisms for accessing computational artifacts, but *e* and *k* have a special use within the *communication artifacts* described in Section 6.1:

$i(t)$**:** attempt to computationally satisfy term t, whether by searching a data store for compatible values or running some complex computation which is outside the scope of the interaction protocol, e.g. to construct an optimal set of value assignments.

$k(t)$**:** attempt to store t in an artifact. This is typically used to write state out for persistence, re-use, summarisation or to remove the need to thread it through the interaction protocol. $k(t)$ indicates that the agent knows t, but this knowledge is not necessarily part of any commitment mechanism. In shadow institutions, $k(t)$ is satisified by constructing an utterance which communicates t to an appropriate audience via the interaction stream.

$e(t)$ attempt to *elicit* a value for term t. This is a point at which human intervention is required—or at least some recourse to the goals of some agency outside the interaction protocol. Formally, this is similar to $i(t)$ in that new information is brought into the interaction. The implication however is that this is based on human intervention rather than computational processes. In communication artifacts, the term is satisfied when the value t can be derived from an utterance in the interaction stream.

Why this particular set of predicates? It would be entirely possible to rewrite the same operations using a single “artifact-access” predicate that subsumes this functionality. The strong distinction between input—$i()$,$e()$—and output—$k()$—is intended to help both with the construction of protocols and their analysis. The split between computation and elicitation is more subtle but attempts to capture the difference between value judgements and operations carried out in service of those value judgements. In both the *i* and *e* cases, the special predicate functions to separate the implementation of of how a particular step is carried out from the protocol which defines the shape of the interaction to be carried out.

5. CONSTITUTIVE AND REGULATIVE RULES - RELATING INSTITUTIONS TO THE WORLD

Before defining shadow institutions, it is useful to explore some of the theory behind the ways in which institutions derive and exercise power. Recent work in open systems has analysed the difference between *i*) *constitutive* rules that define activities and assign social meaning to actions, and *ii*) *regulative* rules that define when actions may be carried out. In particular, Baldoni et al. (2011, 2013) address the constitutive/regulative distinction by using two different languages, one for describing the constitutive model of activity, and a separate regulative policy to specify which actions are allowable in a given situation.

In this work, we are interested in what happens when we do not make much use of regulation, but instead, institutions derive their effective power from constitutive rules.

Constitutive rules define and give meaning to actions; they take the form of "*X counts as Y in context C*", where *Y* is some form of institutional fact. By defining institutional facts, constitutive rules allow for the ascription of status functions, forming the basis of collective intentionality. Without the constitutive rules making up the institution of football, while people can kick a ball around, it is impossible to engage in the activity of *playing football*, or to be *offside*, and the white paint on the field does not gain special status as a dividing line between in- and out-of-play.

Constitutive rules can be broken down into two types (Cherry, 1973), based on whether *X* is a 'brute' fact about the world, such as "That stone weighs 4 kilograms[4]" or an *institutional* fact, which only exists in the context of a human institution, such as "Bob and Sue are married".

A1 rules relate brute facts to institutional ones: *X* is a brute, physical fact: "the ball is in the net", "John said 'I do'". These rules make sense of messy physical reality in terms amenable to formalisation within an institution, and give rise to rules such as "Saying 'I do' *counts as* an agreement to marry".

A2 rules relate institutional facts to each other, through entailment and other relations: "A checkmate is made when a king is attacked in such a way that no move will leave it unattacked". Here, relations are drawn between institutional facts—the attackedness of the king—and institutional activities—different moves—to define a new institutional state of checkmate.

While Cherry uses this distinction to attack Searle's conceptualisation of constitutive rules as conflating two different types of definition, both types of definition are necessary. Within the system, the network of compositions, entailments and subsumptions which $A2$ rules create between institutional facts gives terms their meaning and power, allows for inference, and forms the foundation of formalised actions. However, at some point it is necessary to ground the terms in events, utterances, actions or states which are outside the system, and $A1$ rules do this.

$A1$ rules necessarily have one foot outside of the formal specification, working with a messy, non-symbolic world of human action and utterance, physical objects and situations. This allows raising a hand to *count as* commitment at the next price level in an auction, or a particular community to decide that the piles of schoolbags *count as* goalposts for the institution of football. The translation of brute facts is a key component of human institutions, as it allows for their use and re-use in different situations and context[5].

In contrast, electronic institutions generally work within a world of formalised action; even those which are embedded into virtual worlds (see Section 3.1) have access to a propositional view of the world which is closely related to institutional knowledge. If institutions are to be bound to the more general realm of human discourse, this level of matching, of saying that fragments of text and semi-structured communication *count as* an institutional fact becomes vital. If an institution has severely limited power to regulate, as in the case of shadow institutions, a large portion of their efficacy must stem from their ability to *constitute*, and lay some form of meaning over activity.

[4] Although the *statement* of this fact requires the institution of language (Searle, 2005, p.3)

[5] This is distinct from the ontology matching sometimes used between models, as here we restrict ourselves to a single institution relating to the world, without concerns about matching concepts between different institutions

6. SHADOW INSTITUTIONS

Shadow Institutions[6] are a response to the points outlined above, namely that:

- Electronic Institutions provide a powerful structuring technique for interaction where it can be formalised;
- There are barriers to entry for humans engaging with these institutions, in particular i) learning formal protocols and ii) ceding control;
- Much of human online interaction is not currently formalisable;
- Community is an important part of social networks, so it is desirable to fit with current practice rather than demanding people change their habits.[]

With respect to openness, two different qualities may be identified: openness in terms of *who may* participate, and openness in terms of the *means by which* participation occurs. In the traditional Electronic Institutions view, the institution is the gatekeeper of action: only institutionally valid actions can be carried out by the participants (Figure 2a). While EIs are open in the sense that any agent may join in (Arcos et al., 2005), they constrain interaction in that it must happen *through* the institution. In contrast, a shadow institution is separated from the flow of activity, attempting to match events to institutional structures. This allows the SI to be attached to an existing social network, where people act in whatever manner they choose, with a partial matching of events to protocol elements. Figure 1 shows the intuition behind this move:

- People carry out some kind of interaction, according to some protocol. The protocol may be entirely informal and implicit, the only assumption is that there is some kind of structure to their activity.
- The interaction is carried out through means of utterances in the stream of events, although there are also other events present.
- Institutional machinery can select some of the events as belonging to the interaction, and reconstruct their relationships in terms of a formal protocol.

By separating the means of interaction from the institution, we allow openness around the ways in which interaction takes place.

Figure 5 illustrates a shadow institution in operation. On the left of the diagram is a normal social network, which people are using to communicate and interact. The shadow institution sits alongside the network, and has three main components:

communication artifacts give the institution a view onto the social network, by filtering the stream to present a view of events, and giving the ability to translate utterances in the social network into formal language, and translate formal language into utterances to be sent out onto the network.

shadow agents represent individual users, by presenting their behaviour in an institutional context. They each have a communications artifact which gives them the utterances of their user, and any utterances which can be understood formally are then enacted by the shadow agent in the institution.

institutional agents are used to thread computational support through the interaction, such as computation, coordination and integration with other systems. They have a communications artifact which allows them to convert formal language into utterances to send out on the social network.

Figure 4 provides a simple model of how these components fit together.

[6] The term *shadow* is taken from the discussion in Castelfranchi and Piunti (2012) of agents as *digital shadows* of physical systems in a *mirror world* (Gelernter, 1991), rather than a sense of hidden, crepuscular watchers.

6.1. Artifacts for social engagement

Artifacts are used to give agents their computational environment, and here they are used to convert the freeform “brute facts” of utterances into institutional facts suitable for consumption by agents. This binding must happen outside of the protocol—it concerns the relation of brute facts to institutional facts, and the protocol only deals with institutional facts. Additionally, the exact rules chosen will depend on the context in which the protocol is used.

There are four points where the communication artifacts create contact between the executing protocol and the outside world:

elicitation rules (*elicit*) of the form $E \nrightarrow e(AL,aid)$ are used to relate utterances in the interaction stream to formal statements which shadow agents commit to. The predicate $e(t)$ asks if there is an artifact able to satisfy t based on user input. This is an example of an $A1$ constitutive rule: it is saying “E_i counts as agent *aid* committing to t within this institution”, where e_i is a brute fact in natural language, and t is an institutional fact in IL.

role production (*toRole*) rules deal with something outside the formal specification of the protocols: how does an agent choose which role it would like to take at any given time. In most descriptions of electronic institutions this is generally left as being entirely up to an agent who is responsible for reviewing the available protocols and roles, analysing them in terms of desired outcomes and then choosing an appropriate role. In the present setup, agents are expected to act as avatars of humans, and so must look to them for role choices. Hence the artifacts provide a set of partial functions $E \nrightarrow Role$ which match up utterances to roles to be played.

interaction definition (*toInteraction*) rules correspond to the initiation of an interaction. They produce the protocol which is to be used, as well as initial agent states for the starting set of agents. In order for interaction definitions to be found, a shadow institution must have been instantiated, observing the particular communications channel, with a set of protocols which might be applied. These rules are then responsible for finding utterances, or sequences of utterances the *count as* the beginning of one of the interactions which the institution understands.

dissemination primitives (*Diss*) are the converse of elicitation primitives, dealing with the conversion of institutional facts into utterances. Without these, the institution is capable of matching its internal state to the progression of a human interaction, but doesn’t take any action or contribute to the event stream. $k()$ predicates indicate that something is now “formally” known, and has become an institutional fact, which can be disseminated. $k(t)$ succeeds on a communication artifact if it has a partial function $k(AL,aid) \nrightarrow E$ whose domain matches t, so that the formal term can be converted into a natural language utterance suitable for consumption by the humans in the system, and uttered on behalf agent *aid*. This allows for the diffusion of institutional facts outside of the institution, serving as a mechanism for engaging with the interaction stream to which the institution is attached.

When an institution is running, it (and its agents) attach these artifacts to one or more event streams, using them to connect the utterances made by humans into the agents’ internal state, and relate that state back to the relevant participants (Figure 5)

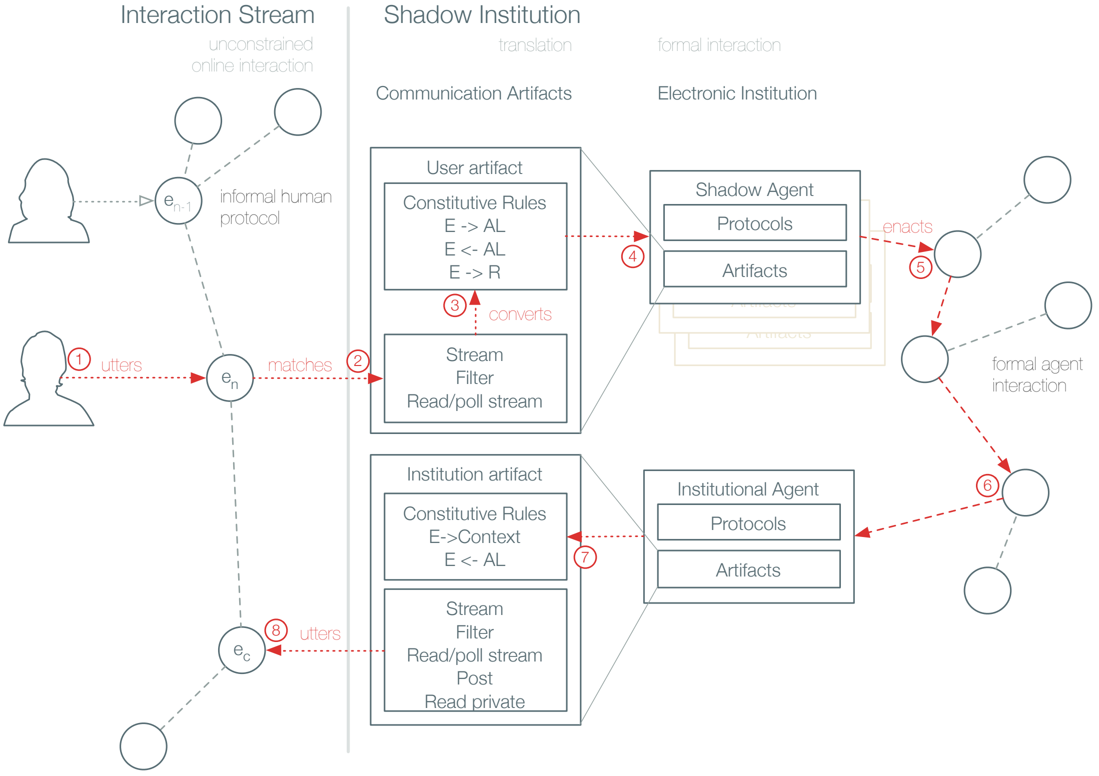


Figure 5. Shadow institution in operation. On the left of the diagram is the event stream of a social network. When a human creates an event (1), it is matched (2) by the communications artifact belonging to a **shadow agent** *representing that user. If possible, it is converted (3) using a constitutive rule into a formal description (4) which the agent then enacts (5) as part of an agent interaction, according to a formal protocol. At some point, the interaction state may produce output (6), which an institutional agent converts using constitutive rules (7) into an utterance which is then sent out into the social network (8).*

6.2. Mechanisms of action

Shadow institutions cannot forcibly regulate interaction; they are able to attach institutional meaning to actions, and communicate based on that view. Instead, however, there are many functions which can be provided which are of utility to the human participants. Much of this functionality is possible without requiring the humans cede a large amount of control to the institution.

Having a live model of state within a formal protocol allows the community to monitor what is happening in their interactions, and create histories of past interaction, with some degree of sense-making already carried out. The institution may also communicate, influencing human behaviour either using messages in the social network, or through some 'out-of-band' means, for example presenting aggregate information on a website. This allows the results of computational procedures, database lookups and interactions with online services to be seamlessly brought into the interaction.

Mechanisms of action hence include:

State: given a protocol, the shadow institution can maintain a state model of the interaction as it progresses. This allows users to interrogate the institution about the current state, options, history and so on. The institution can provide supporting context for human actions without requiring a strong commitment to institutional or even protocol level behaviours: presenting summaries, providing lists of 'available' actions and so on. This can be done without requiring that the humans come under institutional control, only that the institution carries a good enough model of the interaction to provide a useful state model, and that the humans agree with its representation.

Protocol maintenance and enforcement: as humans carry out interaction, there are frequent utterances which maintain the flow of interaction: clarifying meanings, pointing out violations, steering the direction of discussion. Some of this burden can be shared with shadow institutions, by setting them up to notice and respond to events outside the interaction model, at the risk of causing annoyance to the people involved.

History: if the institution can follow along with interactions, there is the potential for providing histories of multiple interactions. Due to the interpretation as institutional facts, these histories contain an implicit degree of sensemaking on top of simply recording the message stream.

Provenance: as an extension of recording history, the provenance of actions within the institution can be recorded: the message which *counted as* making a bet, or the web service which was used to get restaurant recommendations for a dinner reservation.

Computation and integration: one of the primary motivations for creating shadow institutions is the prospect of seamlessly integrating computational into interactions. This includes carrying out optimisation, database lookups, planning, searching and so on—making all of the capabilities of distributed AI systems available in the context of social interaction. If the participants are able to give the institution access to personal calendars or other accounts, their information can be integrated automatically into interactions, and the outcomes can be enacted on their behalf.

Cross-network interaction: one of the side benefits of moving the state model outside of the event stream is the possiblity of using multiple event streams simultaneously. Many people maintain some seamfullnes in their online lives (Barkhuus and Polichar, 2010), however this causes friction when interacting with those on other networks. Creating interactions which span the networks allows for easier interaction without causing 'context collapse'(Marwick and Boyd, 2010).

7. DEVELOPMENT OF A SHADOW INSTITUTION IN LSC

To illustrate how all of these concepts fit together, we will develop a simple model of organising a group of people to go out for dinner, using LSC to define the $A2$ constitutive rules, and then go through the $A1$ rules which are necessary to tie this into the real world of human action, via the interface of artifacts developed previously.

7.1. An interaction language for shadow institutions

Electronic institutions require two languages in our formalism—a domain language (*AL*) and an institutional language (*IL*). We use the Lightweight Social Calculus (LSC) (Murray-Rust and Robertson, 2014) as *IL*, and Prolog-style terms as *AL*.

$$\begin{array}{rcl}
Protocol\ (P) & := & \{Clause,...\} \\
Clause\ (Pt) & := & Actor::Def \\
Actor & := & a(Role,Id) \\
Def & := & E \,|\, Def\, then Def \,|\, Def\, or Def \,|\, c(Def) \\
E & := & Event \,|\, Event \leftarrow C \,|\, C \leftarrow Event \,|\, C \leftarrow Event \leftarrow C \\
Event & := & Actor \,|\, Communication \,|\, null \\
Communication & := & Content \Rightarrow Actor \,|\, Content \Leftarrow Actor \\
C & := & \langle ...Item... \rangle \\
Item & := & k(Term) \,|\, e(Term) \,|\, i(Term) \\
Role\ (Role) & := & Term \\
Content & := & Term
\end{array}$$

Where *null* denotes an event which does not involve communication; *Term* is a structured term in Prolog syntax and *Id* is either a variable or a unique identifier for the actor. $M \Rightarrow A$ denotes that content M is communicated to actor A. $M \Leftarrow A$ denotes that content M is received from actor A. Events can be associated with pre and post conditions, so $C_2 \leftarrow E \leftarrow C_1$, says that event E, can take place if C_1 is true and that C_2 must hold after E has occurred. All conditions are understood within the context of the actor following the clause in which the condition appears. Conditions of the form $k(Term)$ denote that data item *Term* is assumed to be believed by the appropriate actor. Conditions of the form $e(Term)$ describe how the appropriate actor engages with the social interaction. Conditions of the form $i(Term)$ are internal computations performed by the LSC interpreter. The *then* operator represents sequence. The *or* operator represents committed choice. *Term* is some term in an underlying representation language (*AL*) used to represent institutional facts. Names in parentheses are the identifiers used in the formalisation of shadow institutions given in Figure 4.

Figure 6. Syntax of LSC

LSC is an extension of the Lightweight Coordination Calculus (LCC) (Robertson, 2004, 2012) which includes the artifact predicates discussed in Section 4.2. The properties which motivate this choice are:

- it provides a concise specification of the institutional concepts which we are interested in, namely:
  - a nuanced view of actor roles within an interaction, allowing both nested roles and parameterisation of roles;
  - sequencing and choice over events;
  - message passing between actors inside the system.
- LSC is declarative, and clearly separates the structuring of protocols from their execution, but can also be run in an interpreter as part of a computational agent.
- Much of the design and use of LSC is rooted in constitutive rules. Although the distinction is not formally constructed in the language, its heritage as an open protocol and the community of practice around it has a bias towards constitutive specification, and there is no explicit support for deontic modelling.
- As a side effect of execution, traces of key activities are preserved, allowing for the collection of provenance information where desired (this is further discussed in Section 6.2).

Figure 6 details the syntax of LSC; broadly, protocols are composed of clauses, which have an actor role as a head, and some activities as the body. Activities are connected with sequencing (*then*) and committed choice (*or*) and given preconditions with implication. An activity can be sending or receiving a message, interacting with an artifact (see Section 4.2) or $null$. As well as the LSC syntax, some underlying language must be used to represent institutional facts; for this paper we use terms from first order logic.

LSC typically operates as an executable specification language, so engineers write interpreters for it—using the rewrite rules in Appendix A—under the assumption that these will operate in an environment where locutions are passed between message buffers, and are specified formally using the data language of the interaction. An agent working with LSC must, in general, have capabilities to carry out the following activities:

i. Discover a protocol through some mechanism, and decide to follow it;

ii. Establish a channel of communications for exchange of locutions in order to enact the protocol;

iii. Take on one of the roles defined by the protocol, and use that role to create an initial state

iv. Continually rewrite the state according to a set of rewrite rules. For completeness, these are included in Appendix A. This includes copying in the body of new clauses when new roles are called for, creating messages to send, responding to incoming messages and so on.

The states of agents may also be persisted, and an agent may engage in multiple interactions at once, and may play multiple simultaneous roles within a single interaction. The state of an agent for a role within an interaction is a tuple containing its current, unfolded version of the clause it is executing, along with any in- or out-going messages and a copy of the protocol itself.

7.2. Current implementation

The shadow institution framework has been implemented as a freely available open source system called LSCitter (Murray-Rust and Robertson, 2014). A Scala based implementation of LSC is used, which separates the logical core and rewrite rules from the scaffolding around creating and running agents. The Akka framework is used to allow for high performance, distributed execution, and artifactual bindings are provided which allow connection with Twitter, Sparql endpoints and INDX, a personal data store (van Kleek et al., 2012).

The system is designed to be as lightweight as possible in terms of deployment: only three pieces of configuration are required:

- a protocol file which defines the interaction using standard LSC syntax[7].
- a *bindings* file which details the mappings between institutional structures and utterances (see Figure 7) as well as configuration for any artifacts which the agents should have access to.
- a set of accounts on social networks which can send and receive natural language messages.

In this implementation we have used regular expressions to do both the matching and the extraction of information from tweets, although there is nothing in the infrastructure which prevents integrating a more intelligent method of utterance processing. Regular expressions were chosen due to their relative simplicity and clarity in this instance. The brittleness of the matching setup is also an illustration of one strength

[7] This is essentially structures akin to those in Figure 7 re-written using ASCII representations of the extended symbol set

of shadow institutions—by creating the interaction context, we need only worry about the narrow range of utterances which we can understand. Since tweets tend to be very short and to the point, there are relatively small sets of responses to match. Additionally, the text in tweets can be shaped by e.g. sending out tweets to be retweeted. There is currently no entity extraction or similar ontology alignment process, but no fundamental reason not to implement one.

7.3. **Organising Dinner**

Inspired by the discussion in (D'Inverno et al., 2012) of an informal group of people deciding to go an see a movie together, consider a group of people arranging to go for a meal together at a project meeting or a conference. While this is a similar task, there is a key difference—the set of people involved is not known at the start of the interaction. However, the assumption is made that they can be reached through social media channels, in this case Twitter. Additionally, since this is a single interaction, occurring in a relatively small timeframe, it is unlikely that the participants can devote any attention to learning the rules of a system or paying other coordination costs.

In the next paragraphs, we will develop the protocol and bindings file shown in Figure 7.

To begin with, an extremely minimal protocol for organising dinner is as follows:

$$\begin{array}{l} a(participant,A) :: \\ \quad vote(Time,Place) \Rightarrow a(coord(\_,\_),C) \\ \quad or \\ \quad call \Rightarrow a(coord(\_,\_),C) \\ \\ a(coordinator,C) :: \\ \quad vote(T,P) \Leftarrow a(participant,Vo)\; then\; a(coordinator,C) \\ \quad or \\ \quad call \Leftarrow a(participant,A) \end{array} \tag{1}$$

This states that within the current protocol, there are two roles:

- A `participant`, who sends one of two messages:
  - a vote for a particular time and place
  - a message saying that it's time to make a decision
- a `coordinator`, who listens for these messages, and keeps track of the votes.

These are A2 constitutive rules - they relate institutional facts to each other, specifying what *counts as* playing coordinator or participant in this interaction, in terms of messages, temporal ordering and institutional state.

Additionally, in this case they are underspecified—there is no specification for what the *Time* and *Place* variables should be bound to—and impotent, as nothing is done with the information collected.

7.4. Institutional representations of input, output and computation

To begin tying the protocol into the event stream, artifactual predicates (as detailed in Section 4.2) are introduced to indicate that something outside the formal machinery is involved. The first step is to extend the protocol as follows:

$$\begin{array}{l}
a(participant, A) :: \\
\qquad vote(Time, Place) \Rightarrow a(coordinator, C) \leftarrow e(vote(Time, Place)) \\
\qquad or \\
\qquad call \Rightarrow a(coordinator, C) \leftarrow e(enough) \\
\\
a(coordinator, C) :: \\
\qquad k(voted(T, P, Vo)) \leftarrow vote(T, P) \Leftarrow a(participant, Vo) \\
\qquad or \\
\qquad \left(\begin{array}{l} call \Leftarrow a(participant, A)\ then \\ k(decided(T, P, People)) \leftarrow i(decide(T, P, Votes, People)) \end{array} \right)
\end{array} \tag{2}$$

The updated protocol contains no more specification about what is allowed happen when—there is no more regulation—but artifacts are used to provide both state maintenance and external input and output in the following ways:

- $e(vote(Time, Place))$ and $e(enough)$ are used to look for evidence that the agent in question wants to vote for a particular time and place for dinner, or believes that there have been enough votes to select a final outcome.
- $k(voted(T, P, Vo))$ is used to store the knowledge that a person voted for their preference. As a result, the state can be threaded through the interaction without having to explicitly include it in the protocol; a database of votes can be maintained, rather than carrying around an expanding list.
- $i(decide(T, P, People))$ uses the previously stored to calculate the final choice of time and place and the list of attendees. The way this is calculated is not specified, and left up to the agent.
- $k(confirmed(T, P, People))$ is used to record the final outcome of the voting process.

Constituting the act of voting at this level separates it from the mechanics of making decisions. Here, we simply record the state of play by recording what was said by the participants and what decision was made by the computational intelligence.

7.5. Institutional representations of input, output and computation

7.5.1. Input

After the protocol has been expanded, binding are created in a bindings file (Figure 7) translate between institutional structures and utterances. If a participant posts "Lets meet at The Southern at nine thirty" it can be taken to mean that they committing to the proposition *vote("The Southern", "nine thirty")*. The desired conversion is

$$\textbf{@bill}: \textit{“Lets meet at The Southern at nine thirty”} \Rightarrow e(vote(The\ Southern, nine\ thirty)) \tag{3}$$

Regular expressions which covers this and the other elicitation primitives ($e(enough)$ and $e(interested)$) are illustrated in Figure 7*b*. In each case, named groups in the regular expression are translated into Prolog terms for use in the protocol.

7.5.2. Output

Dissemination rules (*Diss*) are used to convey information contained in $k()$ predicates to the appropriate humans. There are three uses of $k()$ in the protocol, (2), and a further one in the full protocol given in Figure 7:

$k(decided(T,P,People))$ occurs when the coordinator has run its decision making algorithm and selected a likely place to go for dinner. The desired effect of this is that people are notified of the decision, so the coordinator maps this to a broadcast tweet (Figure 7d).

$k(confirmed(T,P))$ occurs in the participant's part of the protocol, when the institution has made a decision. This is used to give an alternative communication pathway. Rather than the broadcast message just discussed, this is used to send a direct message to the particular user (Figure 7e)

$k(voted(T,P,Vo))$ **and** $k(subscribed(Vo))$ occur when the coordinator receives a vote or subscription message from a participant. This is only used to store this fact in an institutional database, to be used later when counting votes and confirming the attendees, so no binding is needed.

These are the options which are appropriate in this particular example; a fuller treatment is given in Section 6.2.

7.5.3. Initialisation

The next mechanism is the creation of new interactions. Here, utterances are mapped to institution definitions, consisting of a protocol ID and initial agent states (Figure 7b). In this case, the institution is set up to watch any tweets sent directly to its account in case they are relevant.

Finally, an announcement is given (Figure 7c) to advertise the newly started interactions. This uses a template to translate available variables into natural language, including both institutional information and information extracted from the initial message which was sent.

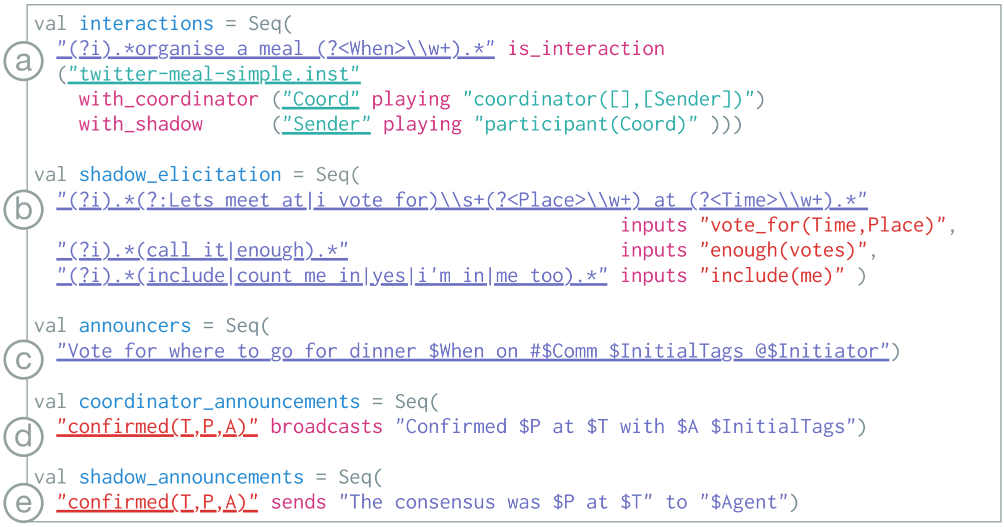
7.6. Example execution trace

Figure 7 gives an example protocol for how this might be carried out, along with constitutive rules for binding this into a stream of natural interaction. Figure 8 details a potential use of this setup. Taken step by step:

1) First, *@ade* sends a message to *@mealbot* indicating a desire to start an interaction. In this case, this is done explicitly, and requires that *@ade* knows of the existence of *@mealbot*, and its ability to support this kind of interaction. Alternative formulations could *i*) allow more specificity about *which* protocol to use; *ii*) allow more paramterisation such as when the meal is, or when a decision should be confirmed by; *iii*) not require an explicit specification, but watch some subset of the interaction stream for meal-organising activities.

2) The *@mealbot* then responds, with a hashtag identifying this particular interaction. Depending on the underlying communications infrastructure, this may not be necessary. For example, messages on twitter can be in reply to other messages, allowing an entire conversation to be recovered. In this case, an explicit channel foregrounds the fact that some set of messages is selected.

$$
\begin{array}{l}
a(participant,A) :: \\
\quad \left(\begin{array}{l} vote(Time,Place) \Rightarrow a(coord(\_,\_),C) \leftarrow e(vote(Time,Place))\ or \\ call \Rightarrow a(coordinator,C) \leftarrow e(enough))\ or \\ subscribe \Rightarrow a(coordinator,C) \leftarrow e(interested) \end{array} \right) \\
\quad then \\
\quad k(confirmed(Time,Place,People)) \\
\qquad \leftarrow confirmed(Time,Place,People) \Leftarrow a(confirmer(\_,\_),\_). \\
\\
a(coordinator,C) :: \\
\quad \left(\begin{array}{l} \left(\begin{array}{l} k(voted(T,P,Vo)) \leftarrow vote(T,P) \Leftarrow a(participant,Vo) \\ or \\ k(subscribed(Vo)) \leftarrow subscribe \Leftarrow a(participant,Vo) \end{array} \right) \\ then\, a(coordinator,C) \end{array} \right) \\
\quad or \\
\quad \left(\begin{array}{l} call \Leftarrow a(participant,A)\ then \\ k(decided(T,P,People)) \leftarrow i(decide(T,P,Votes,People))\ then \\ a(confirmer(confirmed(T,P,People),People),C) \end{array} \right) \\
\\
a(confirmer(X,List),C) :: \\
\quad \left(\begin{array}{l} X \Rightarrow a(participant(C),P) \leftarrow List = [P|Tail]\ then \\ a(confirmer(X,Tail),C) \end{array} \right) \\
\quad or\, null \leftarrow List = [].
\end{array}
\tag{4}
$$



```
val interactions = Seq(
 "(?i).*organise a meal (?<When>\\w+).*" is_interaction
 ("twitter-meal-simple.inst"
   with_coordinator ("Coord" playing "coordinator([],[Sender])")
   with_shadow      ("Sender" playing "participant(Coord)" )))

val shadow_elicitation = Seq(
 "(?i).*(?:Lets meet at|i vote for)\\s+(?<Place>\\w+) at (?<Time>\\w+).*"
                                                  inputs "vote_for(Time,Place)",
 "(?i).*(call it|enough).*"                       inputs "enough(votes)",
 "(?i).*(include|count me in|yes|i'm in|me too).*" inputs "include(me)" )

val announcers = Seq(
 "Vote for where to go for dinner $When on #$Comm $InitialTags @$Initiator")

val coordinator_announcements = Seq(
 "confirmed(T,P,A)" broadcasts "Confirmed $P at $T with $A $InitialTags")

val shadow_announcements = Seq(
 "confirmed(T,P,A)" sends "The consensus was $P at $T" to "$Agent")
```

***Figure 7.* An LSC protocol**

and bindings file for informal dinner organisation. Bindings are: a)* interaction rules*, mapping utterances to protocols and agent definitions; b)* elicitation rules *mapping utterances to agent roles; c)* announcements *to help users find the interaction; and d,e)* dissemination rules *mapping institutional facts to utterances. Named groups in regular expressions are substituted for similarly named variables in the corresponding LSC definitions. In all cases, multiple rules may be present, allowing for e.g. different forms of words to be matched, or different interaction structures to be initiated.

	Mapping	Other Effects	
1	**@ade**: *"@mealbot: I need to organise a meal tonight for #sociam"*	Start interaction in this protocol with *coord*_345 playing the role *coordinator*([],[*ade*]) and *ade* playing the role *participant*	Creates a special communication channel (#345) for this interaction
2	**@mealbot**: *"@ade: Here's a hashtag #345"*		Direct message, just to @ade
3	**@ade**: *"Who wants to join for dinner tonight? #sociam #345"*	Ignored	Uses informal channel to reach fuzzy group
4	**@bri**: *"I'm in! #345"*	*a*(*subscriber*(345),@*bub*)	Reply to (3), also uses channel
5	**@charly**: *"Lets go to Spoons Cafe #345"*	*a*(*voter*(345,*SpoonsCafe*), @*charly*)	Reply to (4)
		(More responses…)	
7	**@ade**: *"@mealbot: lets make a decision #345"*	Direct message, reply to (2)	
8	**@mealbot**: *"We've chosen Wedgewoods #345 #sociam"*	*k*(*dinner*(345,*Wedgewoods*))	
9	**@mealbot**: *"@ade The consensus was Wedgewoods #345"*	*k*(*dinner*(*Wedgewoods*))	…and similar messages to the others

***Figure 8.* Effects of running the LSC shadow institution described in Figure 7**

3) @*ade* then broadcasts a message inviting people to go for dinner.The message has two hastags attached: a specific tag generated for this particular interaction, and a tag referencing the project or conference. This makes use of the social network to address an *ad-hoc public*(Bruns and Burgess, 2011): a soft-edged group of people with interest in a particular topic.

4-6) Replies come in from @*bri*, @*charly* and others, which are interpreted as either subscribing to find out the results of the process, or voting for a particular restaurant to visit. The messages are identified by the interaction hashtag attached (or by their status as descendants in the reply-tree to @*ade*'s message).

7-8) At a certain point, @*ade* then privately messages @*mealbot* to bring the solicitation phase to a close. Some kind of automatic decision procedure is run, which results in @*mealbot* sending out a message with the decision. This message contains the group's hashtag, as included in the original message, and is hence potentially visible to anyone potentially interested.

9) Twitter streams are noisy and lossy: users receive a lot of messages, and not all messages which match a users filters are received. Hence, the protocol is set up to individually message each person who has expressed an interest in the interaction to ensure that the message gets through. Depending on the socio-technical context, this stage may not be necessary, or could be extended to a more complex confirmation procedure to derive a good estimate of how many people will arrive at the restaurant.

At the end of this interaction @*abe* may be the only person who knows that an EI—or any kind of computational support has been involved. If present, an arcane tag identifying the interaction may tip people off, but there is no fundamental requirement for them to know that they are participating in a structured interaction.

However, given more commitment, it is possible to better tailor the interaction. If users provide the institution with dietary requirements and preferences—for example by allowing it read access to their account on a restaurant review website—these can be factored into the automatic decision process. Providing read access to calendars would allow for compatible times to be chosen and so on. Similarly, the decision procedure alluded to here could be a simple tallying of votes, or it could be a complex process integrating reviews and opinions along with the location of participants to optimise restaurant choice.

In this example, the interaction protocol acts as the minimum necessary formal backbone which allows for the integration of any computational intelligence or user specific data, but does not *require* that such data is provided, or any other up-front commitment from participants. Existing communications infrastructure is used, and more importantly, the interaction attaches to the place where people are already conversing, rather than requiring the people to come to the institution.

8. **DISCUSSION**

Shadow institutions are not a replacement for electronic institutions, rather a means by which to make use of some of their most useful features—open, transparent protocols which can be formally reasoned over—while letting go of the constraints which make them difficult to deploy at large: since the state machine is outside the flow of interaction, the institution is no longer in a position to control human behaviour. It cannot authoritatively restrict which utterances can be made, only contribute on an equal footing with the other participants. Participants are free to express whatever utterances they desire, and the institution selects which of these to bring into its state machine, matching itself to the flow of events, in a manner which is responsive enough to be useful to the participants.

One consequence is that not all events need to be understood by the institution: the participants are free to engage in whatever discussion or discursions they desire, as long as the utterances which are relevant to the institution can be found, understood and parsed into the institutional structures required. Another benefit is that existing community infrastructure can be re-used, shifting the burden of identity, authentication and communication away from the designers of the shadow institutions. Existing forms socialisation are preserved and respected, leading to a more *convivial* setting for the institution.

Understanding the kinds of coupled human-computer systems under discussion presents a different challenge to the kind of formal verification normally carried out for multi-agent systems. While work has been carried out on the verification of of LCC protocols (e.g. (Osman et al., 2006)) this does not address the important qualities of a system like this, which are grounded in its ability to support and influence human interaction.

This leaves two crucial questions which we discus here: firstly, how do SIs come to be, with protocols which match desired behaviours and bindings which relate these to real-world behaviour; and secondly, how should the design of SIs be approached, when some of the formal methods and techniques surrounding EIs are not available.

8.1. Creating and understanding Shadow Institutions

>From the examples given in section 7, it may be clear that creating shadow institutions is a complex task. As well as the knowledge of formal methods needed to create the protocol, some other technologies need to brought to bear in order to match it to the interaction stream. Here we speculate on the manner in which protocols and shadow institutions can be designed and deployed, and the ways in which they can fail.

Design of protocols requires a simultaneous understanding of data flow and the interactions which give rise to it. Currently, this is a time consuming step, requiring expert knowledge. However, the potential for 'play in' (Harel, 2008) or interactions exists, as does adaptation of the protocol in response to the behaviour of the community using it (Robertson and Giunchiglia, 2013). Additionally, the increasing ability to extract some level of structure from online interaction (Ritter et al., 2010; Zhang et al., 2013), along with the increasing amount of data available, paves the way for automatic generation of protocols which match observed behaviour.

As protocols are designed, there is also a need to help explain their behaviour and intended effects. As such, our implementation includes some simple graphical tools which can help to understand protocol structure, in particular the message passing which can occur between agents, and to illustrate the actions taken during the course of a run. These can help semi-expert users to more quickly understand protocol structures, but they are of little use to non-technical users.

Alongside this is the question of binding the resulting protocols to freeform utterances. Here, we have used regular expressions, which are both brittle with respect to surface features of language, and awkward to construct or maintain. There is clearly scope for more advanced natural language processing techniques to be used here. However, part of the strength of the shadow institution setup is that the institutional context provides structure which makes the recognition simpler and more robust—only certain utterances are of interest at any given time, and others may be ignored. Given a set of observed data, it is entirely plausible to create tools which aid in matching utterances to formal elements, aiding less technical users in designing and debugging the bindings.

The possibility that bindings incorrectly match utterances is a danger, as the institution is then misrepresenting the actions and desires of the participants. However, this is the tradeoff for a more lightweight approach to interaction: when free action is interpreted, there is always the potential for error. Minimisation strategies depend on the context: there may be some action people can take to ensure clarity (e.g. tagging their utterances), or the system may be able to ask for clarification if it is not certain. If the interaction is summarised in some way these discrepancies can show up and be corrected, allowing users and designers alike to verify that their behaviour was correctly interpreted.

Conversely, the utterances produced by the shadow institution may fail to achieve desired results when seen by humans. They may be poorly constructed, preventing understanding, or the rate and manner of communication may be inappropriate to the community. In general, this is a link which must be designed *in situ*; not enough is currently known about rewards, incentives and computational interference therein to reliably specify effective pathways for institutional feedback.

However, not every user has to be able to design an SI, and there is space for the gradual design of bindings and protocols which are reused by a wider community. Currently, systems where experts setup protocols which users can invoke with a minimum of configuration are gaining traction. For example, If This, Then That (https://ifttt.com/) allows users to connect a curated, centrally developed corpus of scripts together by setting a few key variables and triggers. This situation is very close to selecting an appropriate protocol with bindings and adding a small amount of configuration for the interaction at hand. While the authors would tend towards a more open approach, it illustrates the power of a well designed set of interaction primitives when they are made ergonomic for general users.

8.2. **Appropriate design**

Ameliorating the difficulty of designing shadow institutions is the idea that they do not have to everything all at once. In a given situation, there are some parts of the interaction which it is appropriate for the humans to carry, and parts which can be practically mechanised. Modelling the institution alongside the stream of interaction rather than prior to it allows for a mixed mode of operation, so that islands of formality can grow organically, where appropriate.

Where in a traditional institution, any interaction must entirely match the protocol used, in a shadow institution there is the possibility of divergence. While this is likely to be anathematic to many institution designers, it allows a certain flexibility: the protocol need only cover the aspects of the interaction which are necessary for formalisation, leaving the rest unmodelled and open. Similarly, partially finished institutions can be deployed, and the violations, transgressions or extensions observed and used to refined the protocols—a *desire lines*(Alexander, 1975) approach to electronic infrastructure. This opens the possibility of learning protocols through observation, as participants are free to act in whatever way they choose, rather than within the constraints of the institution as deployed.

The driving question is then: *"Which parts of this interaction should be left informal, and which parts would benefit enough from computational support to justify formalisation"*. Shadow institutions allow a choice about which parts of the interaction should be carried by the humans, and which by the institution. Some lines of division are as follows:

Trust: if the interaction is based in a community which already has a level of trust, then there is little need to create an architecture of trust and reputation within the electronic institution. SIs are typically deployed into existing groups of people, where a significant proportion of the population who would like to be able to carry out the interaction in question, so social means can be used to sanction poor behaviour. For example, selling items on a departmental mailing list works because there is already a fabric of trust and reputation which means that a formal institution for commerce is an unnecessary burden.

Decision making: part of the reason to bring in an institution may be to integrate some form of computational decision making into the interaction; however, at other times, the institution can be used to process and present information, leaving the choices of how to act up to the community.

Contextual knowledge: creating a comprehensive context for agents to work within can be a huge task; if this knowledge exists in the human community, it may be that it does not require formal representation within the institution.

Discovery: most social networks have comprehensive features to share and discover content. This is then a task which can be passed off to a combination of the users and the supporting infrastructure.

Commitment: eBay is an onine marketplace, and as such a gigantic commitment machine, with clear guidelines and procedures in place for violation; however people still sell goods and services through handwritten cards on noticeboards. There are situations where the humans are in the best place to verify the discharge of commitments.

In summary, the principle is to create the lightest weight possible institutional framework which still brings the benefits of formalisation and computational intelligence to bear on the interaction at hand. By being both lightweight and contextually appropriate, this way of thinking about institutions expands the range of situations in which they can be applied.

9. CONCLUSIONS

Electronic Institutions offer powerful structuring mechanisms which help heterogeneous agents coordinate interaction around shared tasks. However, they have historically been limited in scope and have not found a home in the the fluid and frictionless world of online human interaction. In response, we have developed a technique for applying EIs in a 'softer' way, trading regulatory power for seamless integration and low barriers to entry: shadow institutions.

Shadow institutions are designed so as to be lightweight with regard to the users of the system, in order to support their activities without constraining them. SIs sit alongside interaction streams, annotating utterances with institutional structures. This means that the powers of EIs can be brought to bear, assisting humans in structuring their interaction around particular goals in situations which are typically somewhat chaotic.

The central contribution here is a re-contextualisation of the ways in which institutions function, to avoid the need for participants to submit their behaviour for institutional regulation. Instead, the *constitutive* power of institutions is used as the main animus of shadow institutions, as they observe interaction and map it into institutional structures using *counts as* relations. These institutional facts are then processes by computational agents on behalf of the humans, carrying the burden of understanding interaction protocols and entraining with the formal machinery of the institution. In the extreme case, the agent's mediation means that participants may be unaware that any computational support exists until the institution contributes something useful to the discussion.

This abandonment of regulation allows for a traditional institutional state machine to run alongside freeform, natural interaction, creating a formal trace of what has happened, unfolding protocols alongside discussion. The structures created then serve as a framework on which to hang computational support, whether in the form of intelligence, access to data, or simply recording and aggregating.

We have developed the theoretical aspects of Shadow Institutions alongside an implementation, and as such we present worked examples, using a process calculus to describe institutional protocols and simple matching functions to relate this to human interaction. This implementation is open source, and several example interactions are available to be modified and run, following conversations on Twitter and carrying out simple computational support tasks in the background.

Finally, we have described some design approaches, to understand how this novel approach to deploying electronic institutions can help to support online social interaction with minimal changes to the structure of existing communities. We discuss how computational intelligence can hence be applied in ways which are natural and convivial, supporting human endeavour rather than constraining it.

COMPETING INTERESTS

The authors declare that they have no competing interests.

ACKNOWLEDGEMENTS

This work is supported under SOCIAM: The Theory and Practice of Social Machines. The SOCIAM Project is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/J017728/2 and comprises the Universities of Oxford, Southampton, and Edinburgh.

10. REFERENCES

Ahmad, S, Battle, A, Malkani, Z, and Kamvar, S. (2011). The jabberwocky programming environment for structured social computing. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11* (2011), 53.

Aldewereld, H, Dignum, F, García-Camino, A, Noriega, P, Rodríguez-Aguilar, J. A, and Sierra, C. (2007). Operationalisation of norms for electronic institutions. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*. Springer, 163–176.

Alexander, C. (1975). *The Oregon Experiment*. Vol. 3. Oxford University Press.

Aranda, G, Trescak, T, Esteva, M, and Carrascosa, C. (2011). Building quests for online games with virtual institutions. In *Agents for games and simulations II*. Springer, 192–206.

Aranda, G, Trescak, T, Esteva, M, Rodriguez, I, and Carrascosa, C. (2012). Massively multiplayer online games developed with agents. In *Transactions on Edutainment VII*. Springer, 129–138.

Arcos, J. L, Esteva, M, Noriega, P, Rodríguez-Aguilar, J. a, and Sierra, C. (2005). Engineering open environments with electronic institutions. *Engineering Applications of Artificial Intelligence* 18, 2 (2005), 191–204.

Artikis, A and Pitt, J. (2009). Specifying open agent systems: A survey. *Engineering Societies in the Agents World IX* (2009), 29–45.

Baldoni, M, Baroglio, C, Bergenti, F, Marengo, E, Mascardi, V, Patti, V, Ricci, A, and Santi, A. (2011). An Interaction-Oriented Agent Framework for Open Environments. In *AI*IA*. 68–79.

Baldoni, M, Baroglio, C, Marengo, E, and Patti, V. (2013). Constitutive and regulative specifications of commitment protocols. *ACM Transactions on Intelligent Systems and Technology* 4, 2 (March 2013), 1–25.

Barkhuus, L and Polichar, V. E. (2010). Empowerment through seamfulness: smart phones in everyday life. *Personal and Ubiquitous Computing* 15, 6 (Dec. 2010), 629–639.

Bernstein, A, Klein, M, and Malone, T. W. (2012). Programming the global brain. *Commun. ACM* 55, 5 (May 2012), 41.

Bex, F, Lawrence, J, and Reed, C. (2014). Generalising argument dialogue with the dialogue game execution platform. *Computational Models of Argument: Proceedings of COMMA 2014* 266 (2014), 141.

Bogdanovych, A. (2007). *Virtual Institutions*. Ph.D. Dissertation. University of Technology, Sydney.

Bogdanovych, A, Berger, H, Simoff, S, and Sierra, C. (2005). Narrowing the gap between humans and agents in e-commerce: 3D electronic institutions. In *E-Commerce and Web Technologies*. Springer, 128–137.

Boyd, D, Golder, S, and Lotan, G. (2010). Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *International Conference on System Sciences*. Ieee, 1–10.

Brambilla, M, Fraternali, P, and Vaca, C. (2012). BPMN and Design Patterns for Engineering Social BPM Solutions. In *Business Process Management Workshops*, Florian Daniel, Kamel Barkaoui, and Schahram Dustdar (Eds.). Lecture Notes in Business Information Processing, Vol. 99. Springer Berlin Heidelberg, 219–230.

Bruns, A and Burgess, J. E. (2011). The use of Twitter hashtags in the formation of ad hoc publics. In *6th European Consortium for Political Research General Conference*. Reykjavik, Iceland.

Caire, P. (2009). Designing convivial digital cities: a social intelligence design approach. *AI & Society* 24, 1 (Feb. 2009), 97–114.

Campos, J, López-Sánchez, M, and Esteva, M. (2009). Coordination support in multi-agent systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 1301–1302.

Castelfranchi, C and Piunti, M. (2012). AmI Systems as Agent-Based Mirror Worlds: Bridging Humans and Agents through Stigmergy. In *Agents and Ambient Intelligence: Achievements and Challenges in the Intersection of Agent Technology and Ambient Intelligence*, Tibor Bosse (Ed.). Ios Press.

Cherry, C. (1973). Regulative Rules and Constitutive Rules. *Philosophical Quarterly* 23, 93 (1973), 301–315.

Chesñevar, C and Maguitman, A. (2013). E 2 participation: electronically empowering citizens for social innovation through agreement technologies. In *Proceedings of the 14th Annual International Conference on Digital Government Research*. ACM, 279—–280.

de Jonge, D, Rosell, B, and Sierra, C. (2013). Human interactions in electronic institutions. In *Agreement Technologies*. Springer, 75—–89.

De Roure, D, Goble, C, Aleksejevs, S, Bechhofer, S, Bhagat, J, Cruickshank, D, Fisher, P, Kollara, N, Michaelides, D, Missier, P, and others, . (2010). The evolution of myexperiment. In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*. IEEE, 153–160.

Dignum, F. (2002). Abstract Norms and Electronic Institutions. In *Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*. 93 —– 104.

D'Inverno, M, Luck, M, Noriega, P, Rodriguez-Aguilar, J. a, and Sierra, C. (2012). Communicating open systems. *Artificial Intelligence* 186 (July 2012), 38–94.

Esteva, M. (2003). *Electronic Institutions: from specification to development*. Ph.D. Dissertation. Technical University of Catalonia.

Esteva, M, De La Cruz, D, and Sierra, C. (2002). ISLANDER: an electronic institutions editor. In *Autonomous agents and multiagent systems*. ACM, 1045–1052.

Esteva, M, Rodriguez-Aguilar, J. A, Arcos, J. L, and Sierra, C. (2011). Socially-aware lightweight coordination infrastructures. *Agent-Oriented Software Engineering (AOSE 2011)* (2011), 117–128.

Esteva, M, Rodriguez-Aguilar, J.-A. J, Sierra, C, Garcia, P, and Arcos, J. L. (2001). On the formal specification of electronic institutions. In *Agent mediated electronic commerce*. Springer, 126–147.

Franklin, M. J, Kossmann, D, Kraska, T, Ramesh, S, and Xin, R. (2011). CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 61—–72.

García-Camino, A, Noriega, P, and Rodríguez-Aguilar, J. A. (2005). Implementing norms in electronic institutions. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM, 667–673.

Gelernter, D. H. (1991). *Mirror worlds, or, The day software puts the universe in a shoebox–: how it will happen and what it will mean*. Oxford University Press New York.

Harel, D. (2008). Can Programming Be Liberated, Period? *Computer* 41, 1 (jan 2008), 28–37.

Honeycutt, C and Herring, S. (2009). Beyond microblogging: Conversation and collaboration via Twitter. In *International Conference on System Sciences*. 1–10.

Illich, I. (1973). *Tools for Conviviality*. Harper & Row. ISBN: 0006336213.

Lamizet, B. (2004). Culture - commonness of the common? *Trans, Internet journal for cultural sciences* 1 (2004), 15.

Liu, J. (2013). Interactions: The Numbers Behind #ICanHazPDF. http://www.altmetric.com/blog/interactions-the-numbers-behind-icanhazpdf/. (2013). Accessed: 2015-11-03.

Luck, M, Mahmoud, S, Meneguzzi, F, Kollingbaum, M, Norman, T. J, Criado, N, and Fagundes, M. S. (2013). Normative agents. In *Agreement Technologies*. Springer, 209–220.

Malone, T. W, Laubacher, R, and Dellarocas, C. (2009). Harnessing Crowds : Mapping the Genome of Collective Intelligence. (2009).

Malone, T. W, Laubacher, R, and Dellarocas, C. (2010). The Collective Intelligence Genome. *MIT Sloan Management Review* 51, 3 (2010), 21–31.

Marwick, a. E and Boyd, D. (2010). I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media & Society* 13, 1 (July 2010), 114–133.

Murray-Rust, D and Robertson, D. (2014). LSCitter: building social machines by augmenting existing social networks with interaction models. In *Social Machines workshop at WWW2014*. Seoul.

Myhill, C. (2004). Commercial success by looking for desire lines. In *Computer Human Interaction*. Springer, 293–304.

North, D. C. (1990). *Institutions, institutional change and economic performance*. Cambridge University Press.

OASIS, . (2007). Web Services Business Process Execution Language, version 2.0, OASIS Standard. (2007).

Object Management Group, . (2011). Business Process Model and Notation (BPMN), version 2.0. (2011).

Omicini, A, Ricci, A, and Viroli, M. (2006). Agens Faber: Toward a Theory of Artefacts for MAS. *Electronic Notes in Theoretical Computer Science* 150, 3 (May 2006), 21–36.

Omicini, A, Ricci, A, and Viroli, M. (2008). Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems* (2008).

Osman, N, Robertson, D, and Walton, C. (2006). Run-time model checking of interaction and deontic models for multi-agent systems. In *Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*. 238–240.

Parikh, R. (2002). Social software. *Synthese* 132 (2002), 187–211.

Risman, P. (2009). Pear Analytics: Twitter Study. http://www.scribd.com/doc/18548460/Pear-Analytics-Twitter-Study-August-2009. (2009).

Ritter, A, Cherry, C, and Dolan, B. (2010). Unsupervised Modeling of Twitter Conversations. In *Human Language Technologies*. 172–180.

Robertson, D. (2004). Multi-agent coordination as distributed logic programming. In *ICLP 2004, LNCS 3132*, B. Demoen and V. Lifschitz (Eds.). Springer-Verlag, 416–430.

Robertson, D. (2012). Lightweight coordination calculus for agent systems: retrospective and prospective. In *Declarative Agent Languages and Technologies 2011, LNAI 7169*. Springer, 84–89.

Robertson, D, Barker, A, and Besana, P. (2008). Models of interaction as a grounding for peer to peer knowledge sharing. In *Advances in Web Semantics I, LNCS 4891*. 81–129.

Robertson, D and Giunchiglia, F. (2013). Programming the social computer. *Philosophical Transactions of the Royal Society, Series A: Physical Sciences and Engineering* 371, 1987 (2013).

Ross, J. W, Weill, P, and Robertson, D. C. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business Press.

Savarimuthu, B. T. R and Cranefield, S. (2009). A categorization of simulation works on norms. In *Normative Multi-Agent Systems (Dagstuhl Seminar Proceedings)*, Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen (Eds.). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany.

Schall, D, Satzger, B, and Psaier, H. (2012). Crowdsourcing tasks to social networks in BPEL4People. *World Wide Web* (Aug. 2012).

Searle, J. R. (2005). What is an institution? *Journal of Institutional Economics* 1, 1 (June 2005), 1–22.

Siebes, R, Dupplaw, D, Kotoulas, S, Pinninck, A. P. D, Harmelen, F. V, and Robertson, D. (2007). The OpenKnowledge System : An Interaction-Centered Approach to Knowledge Sharing. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*. 381–390.

Sierra, C, Rodriguez-Aguilar, J. A, Noriega, P, Esteva, M, and Arcos, J. L. (2004). Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional* 4, 4 (2004), 33–39.

Trescak, T, Rodriguez, I, Lopez Sanchez, M, and Almajano, P. (2013). Execution infrastructure for normative virtual environments. *Engineering applications of artificial intelligence* 26, 1 (2013), 51–62.

van Kleek, M, Smith, D, Shadbolt, N, and Schraefel, M. (2012). A decentralized architecture for consolidating personal information ecosystems: The WebBox. In *PIM 2012*.

Walther, J. B, Carr, C. T, Choi, S. S. W, DeAndrea, D. C, Kim, J, Tong, S. T, and Van Der Heide, B. (2010). Interaction of interpersonal, peer, and media influence sources online. *A networked self: Identity, community, and culture on social network sites* 17 (2010).

Williams, S. (1967). Business process modeling improves administrative control. *Automation* (December 1967), 44–50.

Wu, S, Hofman, J. M, Mason, W. A, and Watts, D. J. (2011). Who says what to whom on twitter. In *World Wide Web 2011*. ACM, 705–714.

Zhang, H, Monroy-Hernández, A, and Shaw, A. (2014). WeDo: End-To-End Computer Supported Collective Action. In *AAAI Conference on Weblogs and Social Media*. 639–642.

Zhang, R, Li, W, Gao, D, and Ouyang, Y. (2013). Automatic Twitter Topic Summarization With Speech Acts. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING* 21, 3 (2013), 649–658.

A. REWRITE RULES FOR AN LSC INTERPRETER

$S_1 \xrightarrow{A,\mathscr{P},M_i,M_o} S_2$ denotes a transition in the social computation from the state represented by definition S_1, taken from social artifact P, to the state represented by definition S_2 via the actions of actor A. M_i and M_o are sets of messages remaining to be processed at the start and end of the transition. Definition:

$$\begin{array}{lll}
T_1\, or\, T_2 \xrightarrow{A,\mathscr{P},M_i,M_o} E & \text{if} & T_1 \xrightarrow{A,\mathscr{P},M_i,M_o} E \\
T_1\, or\, T_2 \xrightarrow{A,\mathscr{P},M_i,M_o} E & \text{if} & T_2 \xrightarrow{A,\mathscr{P},M_i,M_o} E \\
T_1\, then\, T_2 \xrightarrow{A,\mathscr{P},M_i,M_o} E\, then\, T_2 & \text{if} & \neg\odot(T_1),\, T_1 \xrightarrow{A,\mathscr{P},M_i,M_o} E \\
T_1\, then\, T_2 \xrightarrow{A,\mathscr{P},M_i,M_o} T_1\, then\, E & \text{if} & \odot(T_1),\, T_2 \xrightarrow{A,\mathscr{P},M_i,M_o} E \\
a(R,I) \xrightarrow{A,\mathscr{P},M_i,M_o} D & \text{if} & \mathscr{P} \looparrowright a(R,I)::D \\
X \Leftarrow A_1 \xrightarrow{A,\mathscr{P},M_i,M_o} c(X \Leftarrow A_1) & \text{if} & M_o = M_i \setminus \{m(A,X \Leftarrow A_1)\} \\
M \Rightarrow A_1 \xrightarrow{A,\mathscr{P},M_i,M_o} c(M \Rightarrow A_1) & \text{if} & M_o = M_i \uplus \{m(A_1,X \Leftarrow A)\} \\
T \leftarrow C \xrightarrow{A,\mathscr{P},M_i,M_o} c(E \leftarrow C) & \text{if} & \mathbb{K}(A,C),\, T \xrightarrow{A,\mathscr{P},M_i,M_o} E \\
C \leftarrow T \xrightarrow{A,\mathscr{P},M_i,M_o} c(C \leftarrow E) & \text{if} & T \xrightarrow{A,\mathscr{P},M_i,M_o} E,\, \mathbb{K}(A,C)
\end{array}$$

We also assume the following transitivity rule:

$$T_1 \xrightarrow{A,\mathscr{P},M_1,M_3} T_3 \quad \text{if} \quad T_1 \xrightarrow{A,\mathscr{P},M_1,M_2} T_2,\, T_2 \xrightarrow{A,\mathscr{P},M_2,M_3} T_3$$

$\odot(T)$ denotes that an interaction term, T has been covered by the preceding interaction (we say that it is closed). Definition:

$$\begin{array}{lll}
\odot(c(X)) & & \\
\odot(A\, then\, B) & \text{if} & \odot(A) \wedge \odot(B)
\end{array}$$

$\mathscr{P} \looparrowright X$ is true if clause X appears in the interaction framework $\mathscr{P}$.
$\mathbb{K}(A, C, \psi)$ is true if C can be satisfied from the actor's current state of knowledge, given whatever computational system the actor deploys internally. Definition:

$$\begin{array}{lll}
\mathbb{K}(A,k(C)) & \text{if} & C \text{is known by} A \\
\mathbb{K}(A,e(C)) & \text{if} & C \text{is satisfied through interaction with} A \\
\mathbb{K}(A,C) & \text{if} & C \text{is a predefined function that the system can perform} \\
\mathbb{K}(A,C_1 \wedge C_2) & \text{if} & \mathbb{K}(A,C_1) \text{and} \mathbb{K}(A,C_2)
\end{array}$$

B. SHADOW INSTITUTIONS COMPARED TO TRADITIONAL ELECTRONIC INSTITUTIONS

To illustrate the difference in operation between traditional electronic institutions and their deployment as shadow institutions, Figure 9 sets out the relations between different components in each system, and Figure 2 compares the way in which they operate.

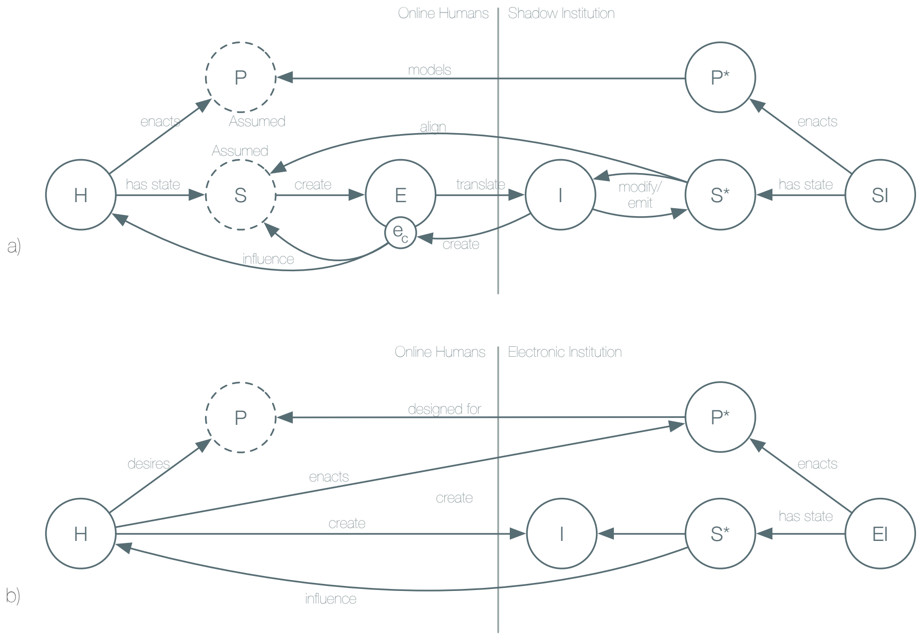


***Figure 9.* Model of the linkages**
for electronic institutions and shadow institutions interacting with an online human population.

Traditional Electronic Institution: In a traditional setting, where an electronic institution is created and then actors engage with it using its own infrastructure, an interaction happens as follows:

T.1 A person h_c has some interaction they would like to carry out, potentially with an informal idea of a protocol P they would like to follow.

T.2 They then discover or design a protocol P^* which matches their idea of how the interaction should run.

T.3 An electronic institution EI is then created running protocol P^*.

T.4 h_c now recruits people to participate in the interaction. This includes:

- discovering that there is a group of people who are doing something of interest;
- deciding to take part;
- creating some form of identity to use, and assessing the trust and reputation of others ;
- entraining with the institution in some way: downloading infrastructure, getting to know the protocols etc.

T.5 The group H can now carry out their interaction. They do this by creating institutional utterances i which are formal moves within the institution.

T.6 These utterances move the state of the institution $S*$ forward, changing the variables, scenes and other relationships within the institution.

T.7 The participants are influenced by the state of the institution in terms of what moves they can make, the value of certain variables (e.g. credit), any information sent out by the institution and any state which is made visible.

Shadow Institution: By contrast, in a shadow institution setting, where the interaction happens on top of existing infrastructure, it unfolds like this:

S.1 A person h_c has some interaction they would like to carry out on a social network which they are part of.

S.2 They find a protocol P^* which models the way that people in the network go about carrying out that interaction (P).

S.3 The protocol is run in a shadow institution *SI*, which can translate some utterances on the network into institutional moves $e \nrightarrow i iff e \in E_i$.

S.4 The existing mechanisms of the social network are used for discovery of the interaction, identity and trust/reputation between the participants.

S.5 The institution can create *shadow agents* which represent people within the formal system, by translating their utterances into institutional moves.

S.6 Some of the participants utterances are now translated into institutional moves i, which move the state machine S^* forwards. The institution attempts to maintain alignnment between S^* and the assumed state of the humans' interaction S.

S.7 The institution may emit institutional utterances i, which are translated to utterances e_c in and placed into the stream of events E.

S.8 By doing this, the institution can influence S, and also the behaviour of the humans in H. As before, institutional state may be presented to participants, and institutional variables may also have some relevance to them.