Social Coordination Systems with Ontology Heterogeneity

Paula Chocrón, Marco Schorlemmer

Artificial Intelligence Research Institute, IIIA-CSIC
Bellaterra (Barcelona), Catalonia, Spain

***Abstract*—We tackle the problem of ontology heterogeneity between participants in artificial social organising systems for multi-agent interactions. We take the case of electronic institutions, and propose an ontology aligning mechanism for systems organised in a centralised way that integrates smoothly with the framework. Our mechanism is based on the Interaction-Situated Semantic Alignment (I-SSA) technique, considering meaning to be dependent on context of the interaction. Agents match their own ontology against a central one, considered as the expected behaviour, while executing the interaction. We provide an alternative, or even a complement, to heterogeneous agents aligning their complete ontologies before the interaction starts. At the same time, we extend I-SSA with new features and set the basis for future work on this direction.**

I. Introduction

Multi-agent systems (MAS) are open environments in which heterogeneous and autonomous agents interact with each other to accomplish some goal. Its qualities of openness and heterogeneity are essential, as they allow agents with different characteristics to participate in the interaction, and this is more important now than ever, with the growth of large distributed systems. The openness of the systems also brings the need for organising the agents for the interactions to be successful. Many frameworks have been proposed to this aim [1], [2], [3]. In [4], the authors discuss the challenge of developing social coordination systems, and outline a general architecture for them. They define these systems as satisfying, among others, the assumptions of openness (agents can enter and leave the system at any time) and heterogeneity of agents (they "may have different decision models, different motivations and respond to different principals"). We argue that an organising system should also take into account another kind of heterogeneity: the one of ontologies. With agents coming from different backgrounds, it is probable that their ontologies will differ, and we believe it is a task of social coordination systems to provide the necessary features for their integration.

A well known example of a social coordination system is the one of electronic institutions [5]. This framework provides a computational analogue of human organisations for complex multi-agent interactions. Several scenarios with fixed rules can be defined, just as in a human organisation. Autonomous and self-interested agents play roles in these scenarios and interact to achieve some goal, with the institution being in charge of controlling their behaviour for the norms to be respected. In this way, electronic institutions that manage to integrate agents in a common and organised environment with very few requisites on them: they do not need to know the protocols to follow, as their behaviour is regulated by the institution itself.

Agents taking part in an electronic institution are, however, supposed to communicate with each other using one central ontology, this is, to use the same vocabulary with the same meanings. This is a very strong assumption, as heterogeneous agents will rarely use the exact ontology provided by the electronic institution they want to participate in. This leaves only one possibility: forcing an ontology matching process between the agents and the electronic institution before the interaction starts. There exist many well-known methods to align different ontologies [6], [7]. However, aligning the complete before the dialogue starts is not always the best choice. On the one side, ontologies can be very large, and it is probable that only a small part of them is used in the interaction, so aligning them completely means a lot of unused work. But we argue that aligning previously to the interaction does not only mean unnecessary work: it can also be not enough. In fact, meaning in a dialogue is not always previously fixed, but instead it evolves together with the interaction. Actually, meaning depends heavily on the interaction context itself. Things such as the moment in which a message is uttered, who the interlocutor is, and what has been previously said can drastically change its meaning. Of course, all these factors will not be taken into account if we align ontologies outside the interaction, losing many important clues to semantic matching. Moreover, even if the agents do align previously to the interaction, the ontologies can eventually evolve and change, so actualizations are needed.

The I-SSA (Interaction-Situated Semantic Alignment) technique [8] takes these ideas and provides an ontology alignment method based on the interaction context. I-SSA provides a meta-procedure to perform semantic alignment between two agents that speak in a propositional language. The alignment is done during the interaction, and received messages are aligned with the messages the agent is expecting in that moment, reflecting the idea of meaning being dependent on the interaction itself.

In this paper, we present a version of I-SSA extended to the complexity of electronic institutions. Our procedure provides an ontology alignment mechanism that is performed during the dialogue, for systems of any number of agents that interact in a centralised way, exchanging messages in a many-sorted first-order language with only predicates. In this way, we do not only provide an alignment mechanism that is suitable for frameworks such as electronic institutions, but also make some advances in the I-SSA technique that can later be extended in order to obtain a stronger procedure. We implemented a preliminary version of our method as a feature for electronic institutions which works as expected with simple examples. Extensive evaluation is planned as future work.

The rest of the paper is organised as follows. In Section II we give a brief overview of the state of the art. In Section III, we present the central concepts of electronic institutions briefly and explain how semantic heterogeneity can arise and how it is currently handled. We also provide an example to illustrate these ideas. In Section IV we present an ontology matching method that extends I-SSA to multi-agent, centralised, first-order interactions. Section V shows how this method is integrated in electronic institutions. Finally, in Section VI we draw conclusions on the work and provide ideas for future work. The Appendix gives more technical details on the implementation.

II. Related Work

Although the majority of research in ontology alignment has been focused on aligning entire, static ontologies, there exist some approaches to matching in the context of interaction for multi-agent dialogues.

In [9], the authors present a layered ontology negotiation protocol for multi-agent interactions. The goal is to build a common ontology on demand, during interaction. Whenever a message is not understood by the receptor, interlocutors move to an meta-interaction where the sender tries to explain the concept in using common terms.

In [10] a communication mechanism in which agents dynamically generate translators between their vocabularies is provided. These translators are partial homomorphisms between the agents' ontologies when needed by the communication.

In [11], a framework to perform alignment in peer-to-peer networks is presented. Semantic bridges between the different ontologies are built on demand. To decide which bridges should be considered, the agents use traditional semantic techniques (syntactic and semantic), as well as the information of previous matchings.

All of these methods succeed in performing the alignment only in the portions of the ontologies that are used in the interaction, avoiding extra work. However, contrary to the method we propose, the alignment relies in traditional matching methods and does not take into account the context of the interaction, which we consider as crucial for the alignment. Our method also differs from previous work in that it makes the social organising system an active participant in the alignment, taking advantage of its characteristics to help the process.

III. Semantic Heterogeneity in Electronic Institutions

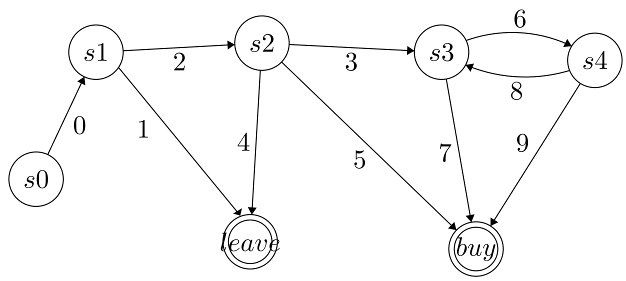
An electronic institution is a representation of a human one. Institutions are organised in scenarios in which participants with different roles interact to perform a task. In electronic institutions, these scenarios are called *scenes*. The agents that take part in an electronic institution move between these scenes following a predefined path. When they enter a scene, they take a role and begin to interact by sending messages to other agents present in the same scene instance. Each scene is defined with a protocol in the electronic institution, that can be expressed as a finite-state machine, and specifies the expected behaviour for each role. The electronic institution counts with different kinds of internal actors, called monitors, which mediate the communication between the participant agents and ensure they behave accordingly to the electronic institution. Whenever an agent tries to send a message that is not specified in the protocol of the scene, it is caught by this monitors and an error is raised.

The specification of an electronic institution consists of:

- A performative structure: it specifies all the scenes in the electronic institution and how agents can move from one to each other.
- A dialogical framework: it fixes the context of interaction and which messages the agents are allowed to say when communicating. It consists of three elements:
  - Roles: the names of the roles that agents can take when interacting in a scene.
  - Illocutionary particles: keywords that are sent with each message to indicate its force (for example *ASK* or *CONFIRM*).
  - Ontology: the vocabulary for the content of messages. It contains everything an agent can say to another one.
- Scenes: each one contains the specification of an interaction, a finite-state machine in which arcs are labelled with messages (expressed in the ontology) or timeouts.

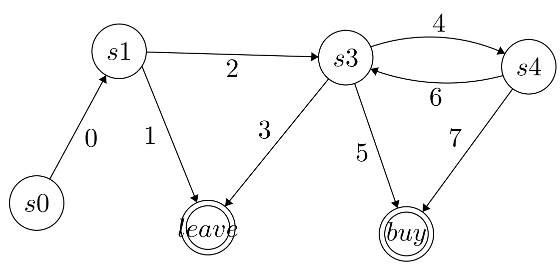
In electronic institutions, all elements of the dialogical framework, including the complete ontology, are considered as shared. Of course the agents, by being autonomous, *can* say anything they want, but they are not expected to do so. So whenever an agent utters a message that is not in the specification of the scene it is in, an error is raised. This means that they are not only expected to share the vocabulary with the electronic institution, but also to follow the exact protocol. Notice that, if we consider meaning as dependent on the interaction, this is equivalent to being semantically heterogeneous.

In this paper we consider the case in which agents can be ontologically heterogeneous with respect to the electronic institution. The specification language such as the illocutionary particles and the name of the roles are common, but agents can have variations with respect to the electronic institution. This heterogeneity can be of two types:



0 = SAY (a buyer) (b seller) (interest item)
1 = SAY (b seller) (a buyer) (decline)
2 = SAY (b seller) (a buyer) (offer price)
3 = SAY (a buyer) (b seller) (offer price)
4 = SAY (a buyer) (b seller) (decline)
5 = SAY (a buyer) (b seller) (accept)
6 = SAY (b seller) (a buyer) (offer price)
7 = SAY (b seller) (a buyer) (accept)
8 = SAY (a buyer) (b seller) (offer price)
9 = SAY (a buyer) (b seller) (accept)

Fig. 1: **The specification of a bargaining interaction in the electronic institution**



0 = SAY (a buyer) (b seller) (interest item)
1 = SAY (b seller) (a buyer) (decline)
2 = SAY (b seller) (a buyer) (offer price)
3 = SAY (a buyer) (b seller) (decline)
4 = SAY (a buyer) (b seller) (offer price)
5 = SAY (a buyer) (b seller) (accept)
6 = SAY (b seller) (a buyer) (offer price)
7 = SAY (b seller) (a buyer) (accept)

Fig. 2: **The specification of a bargaining interaction followed by the buyer agent**

- Different vocabularies. The agent uses a separate ontology for messages.
- Different protocol. The behaviour of the agent does not follow the protocol in the electronic institution exactly: their semantics differ.

Of course, these variations can be combined to form complex heterogeneities. It is worth saying that, while there is no formal limit to the heterogeneity, it will be very difficult for agents with extremely different ontologies to understand each other, so we will work with small variations. How to measure differences between these kinds of ontologies and where the threshold for the alignment is very interesting future work.

We provide an extension for electronic institutions that enables heterogeneous agents to align automatically and while interacting with the ontology and behaviour defined in the electronic institution. First, let us introduce an example to show a possible situation of semantic heterogeneity in electronic institutions.

A. An Example: Street Market

Consider an electronic institution for a street market. A very probable interaction in which agents can engage in this environment is the one of bargaining to fix the price of a product. To provide an scenario for this interaction, the street market will include the scene *bargain*, with two participating

roles: a *seller* and a *buyer*.
The interaction between these actors is as follows: The buyer approaches the seller and shows interest in a given article. If the seller wants to sell the item, she tells her its price; if not she finishes the interaction. If this is not the case, the buyer has three options: she can buy the item without complaining, she can decline the offer and leave, or she can offer a different price. If she chooses to offer, the bargaining begins, and both of them continue to propose prices until any of them agrees with an offer. Notice that the buyer cannot leave without buying after she started the negotiation.

The specification of this behaviour can be found in Figure 1. Let us make some remarks on the protocol it shows. The labels in the arcs are the messages the agents can utter. Messages have four parts: The first word, in capital letters, is the illocutionary particle. The second and third items are the identifiers for the sender and receiver agents respectively, with their name and role. What remains is the content of the messages. We can find two kinds of messages. One, as *accept*, is a fixed constant. The other, as for example *offer price* is a predicate with a variable. These constants and variables will be defined in the ontology of the protocol's owner.

Now suppose two agents find themselves involved in this interaction. The agent playing the seller knows perfectly well how the bargain works, but does not speak English, and performs the whole interaction in Spanish. She follows the same specification protocol as the one in the electronic institution (Figure 1), but with the following messages as labels:

0 = SAY (a buyer) (b seller) (interés objeto)
1 = SAY (b seller) (a buyer) (rechazar)
2 = SAY (b seller) (a buyer) (ofertar precio)
3 = SAY (a buyer) (b seller) (ofertar precio)
4 = SAY (a buyer) (b seller) (aceptar) $\vdots$

As we mentioned, he shares illocutionary particles and roles, but has its own vocabulary.

The buyer (Figure 2) speaks English, but is not very experienced in bargaining, and therefore thinks she can leave without buying at any time in the interaction, as can be seen in the arrow 5. Let us make an observation about this. Semantically, the difference between the buyer and the seller (or the central protocol) is the meaning they assign to the first *offer* the buyer utters. For the seller, this offer has a special meaning, different from the following ones, as it implies a commitment of buying the product in addition to the offer itself. We can find here a clear example of how the context in which a message is uttered can change its semantics. The buyer, on the other hand, does not make this distinction between the first offer and the following ones, causing the semantic mismatch.

Notice that we can find the two types of heterogeneity that we mentioned previously in these agents. The seller shares the structure with the electronic institution, but differs in the vocabulary she uses, while the buyer always sends messages that belong to the central ontology, but behaves in a slightly different way. In the following section we propose an alignment method that deals with these two cases.

IV. An Alignment Technique for Centralised Interactions

The I-SSA technique provides an ontology alignment procedure that agents perform during an interaction. The main idea behind it is that in a dialogue, meaning is given in an important part by the state the participants are in and the messages they are expecting in that moment. The alignment is performed in a meta-interaction: any time an agent wants to send a message, it will instead send a meta-message with the original one as content, and a key illocutionary particle indicating it is a message to align. The receiver will understand it and try to match the original content with the messages it is expecting at the moment, using the information he has from previous interactions. If the matching is successful, agents still exchange some more meta-messages to inform each other about if they arrived to a final state or not. If they agree, the interaction continues, if not, it fails. An interaction is considered successful if both agents arrive to a final state at the same time.

In our method, the agents apply a procedure very similar to I-SSA to align, not between each other directly, but with a central ontology. In this version, we model a MAS as a set of agents plus a special agent, which we will call the central mechanism. Each agent besides the central mechanism is identified with a name and can take a role from a predefined set. Agents communicate by sending messages. Whenever an agent utters a message, it is sent to the central mechanism, which can perform checks or modifications before forwarding it to the corresponding receiver. We now detail our alignment mechanism for these kinds of social coordination systems.

Messages and Interaction Models

Consider each interaction to have the following sets shared between all agents:

- S, of sorts
- R, of roles
- I, of illocutionary particles
- SP, of predicates called *state properties*, such that *initial* and *final* belong to SP

Each agent in the interaction has its own language. The language $\mathcal{L}$ for an agent a in the interaction is a many-sorted first-order language with sorts S,a set of predicates P (each one associated to sorts $\tau_1, \ldots, \tau_r \in S$), and a set of constants C (each one associated to a sort $\tau \in S$).

An agent a can send messages of the type $\langle i, s, r, c \rangle$, where:

- $i \in I$
- s and r are tuples $\langle n, r \rangle$, where n is a string identifying the agent and $r \in R$
- c is an atomic sentence in $\mathcal{L}$: either a 0-ary predicate in P or a predicate in P instantiated with constants in C

A *message pattern* has the same structure as a message, but s and r have variables instead of strings as the first component, and the content c is a non-ground atomic sentence in $\mathcal{L}$: a 0-ary predicate or $p(x_1, \ldots, x_r)$ if p is a r-ary predicate and $x_1, \ldots, x_r$ are variables. Message patterns are useful to specify interactions, as they define the structure of what can be said, while the names of the agents and the instantiation of the content are added while executing.

We model the specification of interactions by means of an interaction model IM, which consists of:

- a set of states
- a set of arcs between the states, each one labelled with a message pattern
- a truth value for each term $p(s)$ with $p \in SP$ and s in the set of states, such that $initial(s) = true$ for only one state s

In short, IMs are finite-state machines whose transitions are labelled with message patterns and whose states are assigned state properties.

Each agent follows its own interaction model, which specifies its behaviour. When executing an interaction, agents will ground the predicates in the content of the messages with constants depending on the situation. The central mechanism also owns an interaction model, that works as the specification of a "correct" interaction.

Notice that agents having their own IM means that, in addition to different vocabularies, agents can have variations in the structure of the interaction which, in the context of I-SSA, means to disagree in the semantics of terms. To summarise, we model as shared information the specification language of the protocols (illocutionary particles, names of the roles, names of state properties), while the content language and the protocols itself are local for each participant. This distinction can be better understood using an analogy with programming languages, where the syntax of the language is fixed, the names of constants, functions, and their semantics is local to each programme.

Let us make a final observation. We model agents as following an IM, but do not force its implementation. To take part in the interaction, we only ask agents to be able to answer, at any time, two questions:

- which messages they are expecting to receive, and
- the values for the properties in the state they are in.

A. The Alignment

Our alignment procedure has the central mechanism as an active participant, and agents align with it instead of between each other. In this way, the language of the central mechanism works as an *interlingua*, and its protocol as the expected behaviour. Aligning against a central ontology is specially useful when we consider interactions between multiple agents, as it keeps a linear number of alignments instead of growing exponentially with the number of agents. It also solves easily the broadcast problem that arises in I-SSA or other peer-to-peer alignment methods. In addition, it models adequately an environment such as the one in electronic institutions, in which there is a "correct" behaviour. In fact, with our procedure every message that is effectively said in the interaction level accords with the central protocol. However, this also has some drawbacks. Mainly, an external ontology is introduced for each interaction, so agents do not really learn about their interlocutor's ontologies, which could be useful later. Including a central mechanism also implies a computational expense, as each message is said and aligned twice for an utterance.

Agents that want to align their ontologies first have to identify themselves as aligners, so that the central mechanism knows how to handle their messages. The alignment is made in two ways. When an aligner agent sends a message, the central mechanism receives it aligns it with the central interaction model, sending the result to the receiver. When an aligner agent receives a message, it aligns it with its own expected messages. Each of this alignments is performed essentially as in standard I-SSA between two agents.

The interaction is considered successful if both the agent and the central mechanism reach a final state at the same time. It is considered unsuccessful when they disagree on the value of a state property for the state they are in. The information of the properties of the reached state is sent by an aligner agent to the central mechanism in a special meta-message with label *INFORM* right after an interaction took place, this is, when it sent or received a message. An unsuccessful alignment with one agent means the failure of the complete interaction.

We now detail the alignment procedure for a message interchanged between two aligner agents. Suppose agents a and b suspected they where heterogeneous with respect to the central ontology, and identified themselves as aligners before starting the interaction. Now consider a wants to send a message with content c and illocutionary particle i to b. The alignment mechanism works as follows:

1) a sends the message $\langle i, a, b, c \rangle$.
2) The central mechanism identifies a message coming from a, which is an aligner agent, so it should be aligned. It calls a matching algorithm to align the content c with the message patterns that it is expecting. If the central mechanism is in state s, this is the set built with the contents of the messages that:
   - are labels of an arc with source state s
   - have illocutionary particle i
   - have sender a and receiver b

   If this set is empty or the matching fails, the alignment is considered unsuccessful and this is informed to everyone. Suppose the matching algorithm matches c with a content c' of a message that labels an arc with target state s_1. Then the central mechanism sends the message $\langle i, a, b, c' \rangle$ to b and keeps the match between c and c' in its list of alignments with a.
3) b now sees a message that is addressed to him. As it is an aligner agent, it tries to align it with the ones it is expecting. It computes the same set of possible matches as in the last item and calls the matching mechanism. If this fails, he informs the central mechanism, which in turn tells all the participants to stop the interaction.
4) Both a and b performed an alignment action, so they send to the central mechanism information about the state they

reached. To this aim, they send a meta-message with the special label *INFORM*, no receiver, and content $\{p \mid p \in SP$ such that $p(s) = true$ in their IM $\}$.

5) The central mechanism compares this to the information it has of the state it reached. If it matches, the alignment continues, if not, it failed and this is communicated to the agents.

In this case, both agents perform as aligners. But notice that this is not mandatory: as the alignment is performed exclusively between the agent and the central mechanism, one of them could align while the other does not. If, for example, only the sender a aligns, step 3 would not happen, and only a would send a message in step 4.

The Matching: We mentioned a matching algorithm was called, both by the central mechanism and the receiver aligner agent. This algorithm matches the content c of a received message with a set D of possible matchings that are computed from the interaction model in the receiver. First notice the very important fact that c is a the content of a message in the execution of an interaction, and therefore it is a grounded term, while D has contents of message patterns, which are non-grounded atoms (or 0-ary predicates).

The first action that is performed is to identify the type signature (arity and sort) of the content c to be matched and to keep only the patterns in D with the same one. This means that if c is a 0-ary predicate, the algorithm will stay only with 0-ary predicates, and if c is an r-ary predicate instantiated with r constants of type $\tau_1, \ldots, \tau_r$, it keeps just the predicate symbols this sort.

Once this is done, the matching itself between the names of the predicates is done. Once we obtained the resulting matching, the last thing to do is to ground it. This is done with the constants that grounded c in the original content. In this way, if $c = p(c_1, \ldots, c_r)$ and p matched with p', the matching algorithm will return $p'(c_1, \ldots, c_r)$.

The names of the predicates are matched propositionally, as in standard I-SSA. The key idea here is to take into account previous successful interactions between the agents. If there is no information about how c was matched in the past, the algorithm chooses randomly an item from D. But if there have been interactions where it was matched and the interaction ended in success, it will consider this information to choose the match. This is done by weighting the items in D according to how many times they matched successfully with c, in order to choose with more probability those that were good options before. For more detail on this part of the algorithm, we refer to Section 5.1.2 (“The Matching Mechanism”) in [8].

Considering only elements of the same sort is of course only a first approach to the matching, and more interesting alternatives can be investigated, as we discuss in Section VI.

V. The Alignment Mechanism for Electronic Institutions

The alignment mechanism we presented in Section IV integrates smoothly with the electronic institutions framework. We provide a detailed description on the implementation in the Appendix. In this section, we will explain briefly the integration.

For simplicity, in this article we work with a slightly restricted version of electronic institutions. The main changes are in what messages the agents can send (only predicates and constants) and what the arcs between states can have (only messages). These restrictions are not difficult to overcome, and extending the technique to embrace the complete possibilities of electronic institutions is future work. We also added one new feature, and it is the possibility to define different properties for the states.

The relation between the method and the elements in electronic institutions is straightforward. The interactions during which the alignment is performed are the scenes, which are specified with protocols that work as the interaction models. The central mechanism is the electronic institution, which has different actors that control the behaviour of the agents. The electronic institution owns an ontology and a protocol for each scene.

The alignment framework in electronic institutions can be divided in three parts:

- The aligner agent: we implemented an special kind of agent that performs the alignment mechanism during an interaction. If an agent wants to align in a scene, it will use this special way of performing, and the electronic institution will know it is aligning. To be able to use this feature, an agent must be able to answer to which messages it is waiting at the moment, and the properties of the state it is in. Aligner agents behave as it is described in Section IV-A.
- The features in the electronic institution: we modified the behaviour of the electronic institution to make it able to recognise aligner agents and align the messages they send with the ones in the specification of that scene.
- The matcher: as both the agents and the electronic institution perform matching of contents, the Matcher is implemented as a separate class. This class performs the mechanism described in Section IV-A to match the content of a message to an element in a set of expected message patterns.

Our implementation is currently functional and some examples have been tried, including the Street Market presented in **??**. The examples work as expected, allowing semantically heterogeneous agents to align their ontologies. Experimentation with more complex examples as well as an evaluation of the success rates for different cases is future work.

A. The Example in Action

Let us go back to the street market institution, and particularly to its bargain scene.

Now that we have a complete picture of the alignment, let us complete the specification of the protocols in our version of electronic institutions. Consider the set of state properties SP to be $\{final, initial, success, failure\}$. The last two ones need some explanation: *success* is true in the final state in which the transaction is performed, and *failure* when it is not.

This captures the very different ways of finishing a bargaining interaction; in fact, the participants would notice it if one of them wanted to buy and the other did not.

Now let as follow the interaction for some messages. Recall that the seller and the agent have their interaction models and behave according to them, while the electronic institution regulates following its own. As both agents suspect they can differ with the central protocol, they will identify themselves as aligners when starting the interaction. The electronic institution will therefore be prepared to align any message that arrives from them.

1) The buyer utters *SAY (a buyer) (b seller) (interest carpet)* . The electronic institution matches *interest* with *interest* and sends to the seller ***SAY (a buyer) (b seller) (interest carpet)*** . The seller matches *interest* with *interés*. Now the buyer sends *INFORM (a buyer) initial* , and the seller the equivalent. The electronic institution has the same information about the reached state, so everything continues.
2) The seller utters *SAY (b seller) (a buyer) (tasar 40)*. The electronic institution matches *tasar* with *value* and sends to the buyer ***SAY (a buyer) (b seller) (value 40)*** . The buyer matches *value* with *value*. Both agents send the *INFORM* message with no content, as no property is satisfied. The electronic institution agrees.
3) The buyer decides to decline the offer and utters *SAY (a buyer) (b seller) decline* . The electronic institution is expecting three different messages, but only two are 0-ary predicates. The electronic institution performs the matching algorithm to decide if it should match *decline* with *decline* or with *accept*. Suppose she chooses the former one, and sends *SAY (a buyer) (b seller) decline* . Now b needs to align. Two things can happen:
   - b aligns *decline* with *rechazar*. Both b and a send *INFORM (a buyer) final, failure* , which matches with the information in the electronic institution. The interaction is considered to be successful.
   - b aligns *decline* with *aceptar*. a sends *INFORM (a buyer) final, failure* , but b instead sends *INFORM (a buyer) final, success* , which does not match. The interaction failed, and this is informed to the participants.

Notice that the matching algorithm is performed even when an agent receives a message syntactically identical to one it is waiting for. This is because we consider the context of interaction to be the only source of meaning, and we take into account the case in which the same term means different things for each agent. A direction of future work is to analyse how to combine this kind of alignment with traditional ontology matching techniques.

VI. Next steps

Our goal in this paper was to provide a context based ontology alignment feature for a particular social coordination system: the one of electronic institutions. We developed a method based in the I-SSA technique to perform alignment in centralised frameworks, and implemented it as an extension for electronic institutions. At the same time, we made advances in the I-SSA method such as a basis for including first-order messages and the possibility of using it in real multi-agent interactions.

Many directions for future work can be taken from this article. As we have already said, we plan to extend the aligning mechanism to consider all the possibilities of electronic institutions, with no restrictions. This means mainly to add support for propositional constraints in the arcs, and to consider other types of messages besides predicates and constants, such as data types or lists. Our final aim is to go beyond electronic institutions and provide a general aligning method for social coordination systems.

Our implementation needs to be extensively tested and evaluated. We plan to create examples that explore the different complexities of heterogeneity and to analyse how our method works on each of them.

On the mechanism itself, we can also find interesting research directions. The main one is to develop the first-order alignment, which is now in a propositional phase. We plan to analyse how to match predicates in a more complex way, being more flexible in the sorts and arity. In this sense, we also plan to consider messages with variables that have been already instantiated before, which is a very common case in interaction specifications (in the literature, $!x$ against $?x$). Finally, analysing how to combine this interaction based with traditional ontology matching techniques is a broad and interesting direction that we also want to explore.

VII. Acknowledgments

The research presented in this paper has been funded by the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 567652 ESSENCE: Evolution of Shared Semantics in Computational Environments (http://www.essence-network.com/). We thank the anonymous reviewers for their useful observations and remarks.

References

[1] H. Cardoso, J. Urbano, P. Brando, A. Rocha, and E. Oliveira, “Ante: Agreement negotiation in normative and trust-enabled environments,” in *Advances on Practical Applications of Agents and Multi-Agent Systems*, ser. Advances in Intelligent and Soft Computing, Y. Demazeau, J. P. Mller, J. M. C. Rodrguez, and J. B. Prez, Eds. Springer Berlin Heidelberg, 2012, vol. 155, pp. 261–264.

[2] R. Kitio, O. Boissier, J. Hbner, and A. Ricci, “Organisational artifacts and agents for open multi-agent organisations: giving the power back to the agents,” in *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, ser. Lecture Notes in Computer Science, J. Sichman, J. Padget, S. Ossowski, and P. Noriega, Eds. Springer Berlin Heidelberg, 2008, vol. 4870, pp. 171–186.

[3] N. Fornara, F. Vigan, M. Verdicchio, and M. Colombetti, “Artificial institutions: a model of institutional reality for open multiagent systems,” *Artificial Intelligence and Law*, vol. 16, no. 1, pp. 89–105, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10506-007-9055-z

[4] P. Noriega, J. Padget, H. Verhagen, and M. dInverno, “The challenge of artificial socio-cognitive systems,” in *17th International Workshop on Coordination, Organisations, Institutions and Norms (COIN 2014)*, 2014.

[5] M. d'Inverno, M. Luck, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra, "Communicating open systems," *Artificial Intelligence*, vol. 186, pp. 38–64, 03/2012 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370212000252?v=s5
[6] Y. Kalfoglou and W. M. Schorlemmer, "Ontology mapping: the state of the art," *Knowledge Eng. Review*, vol. 18, no. 1, pp. 1–31, 2003. [Online]. Available: http://dx.doi.org/10.1017/S0269888903000651
[7] J. Euzenat and P. Shvaiko, *Ontology matching*. Springer, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-49612-0
[8] M. Atencia and M. Schorlemmer, "An interaction-based approach to semantic alignment," *Journal of Web Semantics*, vol. 12-13, pp. 131–147, 2012.
[9] J. van Diggelen, R.-J. Beun, F. Dignum, R. M. van Eijk, and J.-J. Meyer, "Ontology negotiation in heterogeneous multi-agent systems: The anemone system," *Appl. Ontol.*, vol. 2, no. 3-4, pp. 267–303, Aug. 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1412408.1412409
[10] R. M. van Eijk, F. S. de Boer, W. van der Hoek, and J. C. Meyer, "On dynamically generated ontology translators in agent communication," *Int. J. Intell. Syst.*, vol. 16, no. 5, pp. 587–607, 2001. [Online]. Available: http://dx.doi.org/10.1002/int.1025
[11] P. Besana, D. Robertson, and M. Rovatsos, "Exploiting interaction contexts in P2P ontology mapping," *CEUR Workshop Proceedings*, vol. 139, 2005.

APPENDIX
THE IMPLEMENTATION IN ELECTRONIC INSTITUTIONS

In this Appendix we explain with more detail the implementation in the electronic institutions framework. The main changes are the addition of a class that implements the behaviour of an aligner agent, the addition of features in the electronic institution to recognise these new agents and act accordingly, and the inclusion of a Matcher class, in charge of the matching algorithm. In what follows we explain these three elements. We also modified the states in the specification of scenes, to add the possibility of having different boolean values, which are what we called state properties.

A. The Aligner Agent

In electronic institutions, an agent's behaviour for an specific scene is specified in a Performance. As the decision of whether to align or not should be done for each scene, we implemented a performance (the abstract class AlignerPerf) that is in charge of the alignment. If an agent wants to align in a given scene, its performance for that scene will inherit from this instead of from the usual ScenePerf.

The ScenePerf can be summarised as follows:

- It overrides the method receive(message), which is called when the agent receives a message. This method now implements the functionality of calling the Matcher and returning the expected message it chose to match with.
- It implements the method informStateProperties(), which is called after an utterance, and informs the electronic institution of the attributes of the state it will move to.
- declares the method getExpectedMessages(). It must return the messages an agent is expecting in the moment, for a given illocutionary particle. Must be implemented by subclasses.
- declares the method isState(attribute), which returns if the attribute is true in the current state. Must be implemented by subclasses.

An agent that wants to align during an interaction will therefore have a performance that inherits from AlignerPerformance, implementing the getExpectedMessages() and isState(stateProperty) methods. Agents can do this in any way, but we considered the practical case in which they own an explicit Scene protocol specifying their behaviour. To this aim, we created a special kind of aligner Agent, which receives the protocols for each scene building the electronic institution's experiment.. This agents use the WithProtocolPerf, which implements the behaviour of the agent that is following the protocol in a scene. This class inherits from AlignerPerformance and implements very easily its abstract methods: getExpectedMessages() by returning all the contents of the arcs that enter the current state, and isState(stateProperty) by checking the state properties in the protocol.

B. Support for Alignment in the Electronic Institution

We also need to provide the features to perform the alignment in the electronic institution. Basically, it needs to identify when an agent is an AlignerAgent, and catch their messages to perform the alignment before forwarding them to the receiver, or to check the state properties. This is done in the SceneManager class, which is the actor in charge of the interaction between agents and a scene in the electronic institution. This class has access to the electronic institution's specification of the scene (via the class Scene). When the SceneManager receives a message, it first analyses if it comes from an AlignerAgent. If it does, there are two possibilities. It can be a normal interaction message, in which case the manager computes the expected messages for the current state in that scene, calls the Matcher, and forwards the result if the matching was successful. The received message can also be a meta-message labelled with the illocutionary particle *INFORM*, in which case it computes the properties of the target state in the electronic institution's specification. If they do not match, or if there was no possible matching, it raises an Alignment Error.

C. The Matcher

The matcher is a separate class, as it is useful for both the electronic institution and the agents. It implements the method match, that receives a set of TypePattern elements as the possible matches and an Object as the content. As we said, we only consider the cases in which the content is of type FunctionInstance, or an instance. The matching is performed exactly as explained in Section IV-A.