

# A Framework for building EI-enabled Intelligent Organizations using MAS technology.

Armando Robles P.      Pablo Noriega B.V.

*Institut d'Investigació en Intel·ligència Artificial  
IIIA-CSIC. Bellaterra, Barcelona, Spain*

## Abstract

In this paper we propose an agent oriented framework for producing complex information systems using the electronic institutions paradigm. We define an intelligent organization environment that enables the use of available information technologies resources with MAS technology. Using electronic institutions we control the interactions of the intervening agents and the commitments generated by those interactions according to institutional specification. From an information systems developer point of view, our framework allows to separate form design, work-flow specification, and the programming of business rules and agent behaviour in the development of large information systems.

## 1 Introduction

We believe that many of the promises and proven results of agent technologies will have a positive impact in the development of traditional information systems (IS), provided a sound methodology and appropriate tools are put in place.<sup>1</sup> This paper is a proposal in that direction. It outlines a framework for using agent-technologies in the information systems of the so-called intelligent organizations (IO).

Our proposal is based on an "institutional perspective" where an organization is thought of as a group of people that use all types of resources –including IT– in order to better achieve some shared objectives, but they do it in an "institutional" way when they follow some conventions that are intended to facilitate the articulation of their activity. Thus, we assume that if a corporate information system is to support the operation of an intelligent organization it should incorporate the conventions that govern the organization.

We use the notion of an electronic institution (*EI*) to implement that institutional perspective because it provides a convenient way of establishing a link between the conventions that say how an organization should work and the corporate system that support the actual operation. It also provides a unifying metaphor that can be used from the design and specification stages all the way to the testing and updating of a deployed system. Furthermore the *EI* approach favors a clear separation of standard procedures and discretionary behavior. Last, it provides a way of bringing agents effectively into the top level specification layer of the information system as well as in the bottom operation layer of the system's components.

In fact, we use electronic institutions –and software agents– to enable a concrete form of "intelligent organizations" where corporate knowledge is captured through the procedures that establish the operation standards of the organization, on one hand and, on the other, role-specific knowledge is captured in software agents that complement or implement human tasks. While the former takes

---

<sup>1</sup>There are reports of successful applications in different business domains like logistics, manufacturing and e-commerce [10], and also as enabling technology for some IT tasks like simulation, communication, web foraging. Furthermore, although the AgentLink road map draft, [6], reports the likely application drivers for agent technology and the challenges for agent-based computing without explicit reference to corporate systems, it does mention business domains where large CIS are frequent, such as telecommunications, transportation, manufacturing and health care.

care of more stable corporate practices, the second serves to implement more fluid policies and infrequent or exceptional situations. We also intend our framework to implement flexible information systems for organizations that need to adapt to the dynamics of their business domains.

This paper is a discussion of our current work on a framework that involves conceptual distinctions, formal constructs, tools and methodological guidelines. The proposal is based on our experience with large corporate systems and with electronic institutions. We intend it to be applicable to the development of new corporate information systems that are agent enabled but we understand that it should also be pertinent for agentifying legacy corporate systems.

## 2 Background

### 2.1 Intelligent organizations and Corporate information systems

One may think organizations as a group of individuals that act according to a set of shared conventions in order to achieve goals in the best possible way. The shared conventions establish stable procedures that reduce uncertainty about the interactions and facilitate decision making and coordination, [7, 8, 3].

An intelligent organization is understood as a "knowledge-based organization whose business operations and internal processes are founded on knowledge competencies and the value of its products and services is given by the know-how, the intellectual capital and the technological advantage of the organization" [5].

Information systems for organizations, capture the way those organizations work. They are complex systems of programs, data repositories, best practices, operation flows, and canonical documents. We will talk about Corporate Information Systems (CIS) to mean complex information systems that instrument the operation of a corporation or large organization. CIS involve internal users (staff) who take different roles and external users (clients and services) that interact with the organization.

Current practices address three main components in the design and development of a CIS: business rules programming, form design and work-flow modelling. In current CIS development, there are two options to specify the interlacing of this three elements:

- Form centered programming. The flow of activities is governed by forms; that is, the intervening business rules are invoked by fields of the form where they read or write data, in a sequence that is determined by the form design. This approach has the advantage of easy programming, but provides no facilities to implement work-flow control on the intervening processes, and clearly there is no room for normative rules to govern the interaction between the intervening components.
- Work-flow centered programming. The sequencing of the required business rules is specified from a work-flow perspective; that is, the intervening business rules are invoked by states of the the specified work-flow. This approach has the advantage of having all processes well sequenced and with the proper follow-up. However all links between the states and, both, business rules and forms, have to be specified and programmed at design time, resulting in poor flexibility for the "behaviour" of the intervening agents.

We propose a new methodology for building those CIS. It separates forms, business rules and work-flows, and use agent technologies with regular IT resources. The core of the proposed framework is to use the *EI* paradigm and build tools and methodology around it.

### 2.2 A quick look at electronic institutions

The easiest way to describe electronic institutions is as the computational counterpart of traditional institutions. A traditional institution is a means to organize, articulate, or in some other way structure agent interactions. For instance, economist D. North [8] characterizes institutions as "the rules of the game" or as "humanly devised constraints that shape human interactions". Thus, institutions are conventions that a group of agents follow in order to accomplish some socially agreed upon objectives. Analogously, then, an electronic institution is an implementation of conventions that apply to the interactions of agents that may be human or software agents.

In this paper we will use the notion of electronic institution (*EI*) as defined by the IIIA team for the EIDE environment (vid.e.g. [1]). The main elements of that definition are:

1. Participants are commitment-making agents (human or software).
2. All institutional interactions are speech acts.
3. All admissible institutional interactions have the intended institutional effects.
4. An electronic Institution is specified through the following components:
  - A Dialogical Framework that defines the ontology of that *EI* and the language conventions that will be admissible in that *EI*.
  - A Deontological component that establishes the pragmatics of the admissible illocutory actions. This component may be understood as a set of norms that constrain possible conversations and manages the obligations established within the institution. It is currently formed by two constructs:
    - A performative structure that includes a network of scenes linked by transitions between scenes. Scenes are role-based interaction protocols represented as finite state machines, arcs labelled by illocutions and nodes corresponding to an institutional state. Transitions describe the agent flow policies between scenes.
    - Rules of behavior that establish role-based conventions that regulate commitments established within the *EI*. These are expressed as pre and post-conditions of the illocutions that are admissible in the performative structure of the *EI*.
5. There is an *EI* set of tools that includes a specification language that generates an executable version of an *EI* and there is middleware that enables that version to be enacted by actual agents.

For the proposal outlined in this paper we extend that definition in two directions. First, we will now refer explicitly to the conventions that map the *EI* into the real world and, second, we define the composition of several *EI*s into one Federation of *EI*s.

### 3 A framework for *EI*-enabled intelligent organizations

In this section, we introduce a framework for building *EI*-enabled intelligent organizations. This framework is used to build CIS using MAS technology in an environment where agent interactions are regulated by electronic institutions.

Our proposed framework contributes to current CIS development with:

- An agent oriented methodology for producing complex information systems as intelligent organizations.
- The use of the electronic institutions paradigm in an organizational context to separate form design, work-flow specification, and the programming of business rules and agent behaviour.
- An intelligent organization environment that enables available information technologies resources (such as data bases, business rule repositories, data mining tools and automated decision making devices) with MAS technology.
- A prescriptive specification that controls the interactions of the intervening agents and the commitments generated by those interactions.
- A grounding language that allows the interaction between the institutional agents and the business domain agents, resources and repositories.

Our proposed framework allows the building of CIS that encapsulate corporate knowledge effectively and are flexible enough to adapt to a changing business environment. Figure 1 shows how we propose to associate a definition of the way we expect an organization should function to the corporate information system that supports the day to day operations of the actual organization. The core is an electronic institution that constitutes an implementation of a corporate conventions layer at the very top and is mapped to the actual CIS components at the bottom through a CIS middleware layer. The following paragraphs explore the elements of the diagram.

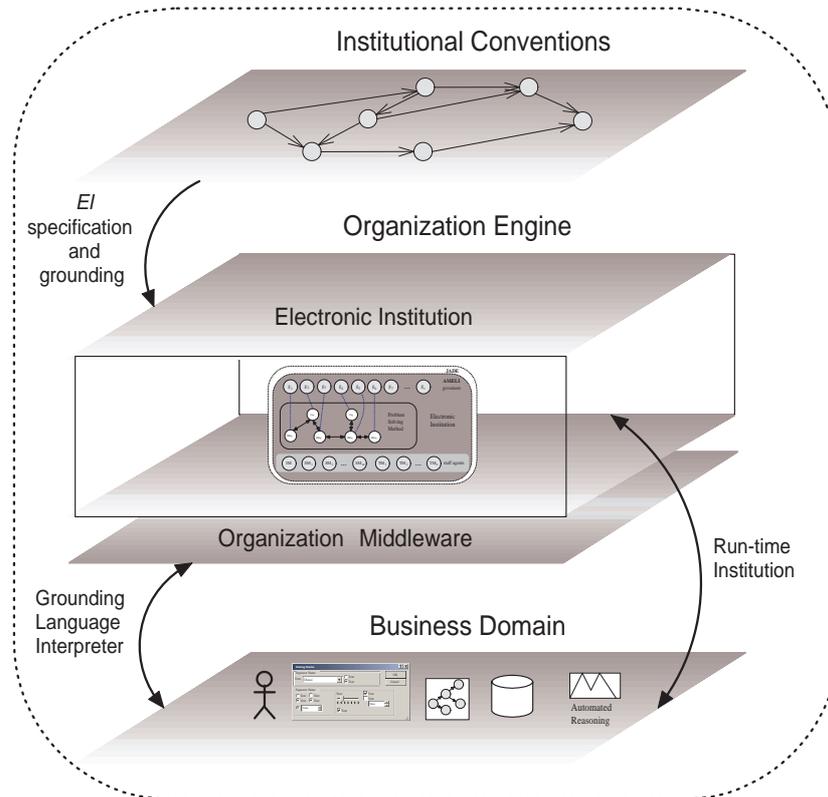


Figure 1: *EI*-enabled Intelligent Organization

### 3.1 Institutional conventions.

The function of this layer is to have a prescriptive representation of the organization, that is, the conventions that state how the organization should work. The inputs for this layer are, therefore, the institutional rules that capture corporate know-how such as organizational guidelines, and business process descriptions that establish *how* agent interactions should take place. In this layer we model norms, agent interactions, and business rules sequencing.

The institutional rules are taken by the *EI specification language* to produce an executable *EI* specification whose interactions with the business domain components are defined and handled through the *grounding language*.<sup>2</sup>

### 3.2 The organization engine layers

The purpose of the organization engine is to drive the execution of the whole organization. It is composed by two layers: the *EI* layer and the organization middleware layer. These two layers contain all the elements needed to interpret the institutional conventions and to enact them as an intelligent organization.

<sup>2</sup>In this proposal we use Islander [1] as the *EI* specification language, it provides a graphic environment for the institution specification and produces an *EI* specification in a *xml* file format.

### 3.2.1 Electronic institution.

This layer implements the normative specification using as input the *problem solving plans* —*EI* performative structures— produced by the *EI* specification language and instantiates these methods with the business domain entities specified through the grounding language.

The run-time institution is the program that enacts the *EI*.<sup>3</sup> It runs in close interaction with the organization middleware and it guarantees that all the institutional conventions are satisfied.

### 3.2.2 Intelligent organization middleware layer.

This layer glues the *EI* with *business domain* elements through a *grounding language interpreter* it contains.

The basic functions of this layer are:

- log users into the organization, controlling user roles, agent resources and security issues.
- monitor user interaction,
- execute the *grounding language* interpreter,
- implement interaction devices, and
- control the actual mappings between the *grounding language* interpreter and domain entities.

The Grounding language is used to specify agent behaviour and how it interacts with *business domain* elements. In order for domain agents to interact with other domain elements, for every object of the domain we create an abstraction that we call its *interaction device*. Hence, the *grounding language* contains commands that allow agents to use *interaction devices* and therefore communicate with humans, access *business domain* repositories or change the normal flow of interaction.

## 3.3 Business domain

The business domain contains –but it is not limited to– the following components:

- *Agents*.. There are three types of agents: organization staff agents that implement institutional behaviour of the organization staff, user agents that represents external users of the CIS, and server agents that integrates all the repositories of the business domain with the other agents.
- *Interaction devices*. This devices implement interfacing capabilities between agents and other domain elements, e.g. form handling, data base calls, business rules triggering.
- *Problem solving plans*. *EI* executable business context components that constitute procedural conventions for the organization. (details in [11]).<sup>4</sup>
- *Repositories*. Business rules, data base and AI techniques accessible to agents.

## 4 Linking framework and CIS

In order to show how the framework is intended to work, we use the example of a hotel corporate information system as a typical business domain.

From a functional point of view a hotel can be seen as a network of interrelated business activities that we call *business contexts*. Each business context accomplishes some specific corporate objectives and involves agents who are subject to corporate procedural conventions specific to that context.

We may define a business context as a regular electronic institution that defines the conventions participating agents should comply with. That is, an *EI* with roles, illocutions, repetitive scenes

---

<sup>3</sup>We use Ameli [1] as the run-time engine for this layer.

<sup>4</sup>In [11] problem solving plans were referred to as problem solving methods

and transitions where obligations are committed and fulfilled. For the moment we take business contexts as independent, although they may share corporate conventions and they may also have some coordination points where agents that belong to different contexts may interact (human assistants, human tasks that can be automated, latent tasks, system enabling tasks, hotel staff and CIS staff agents).

Figure 2 shows how a "Check-in" business context is related to other hotel business contexts and how it is transformed from a regular workflow description into an *EI* that is EIDE implementable.

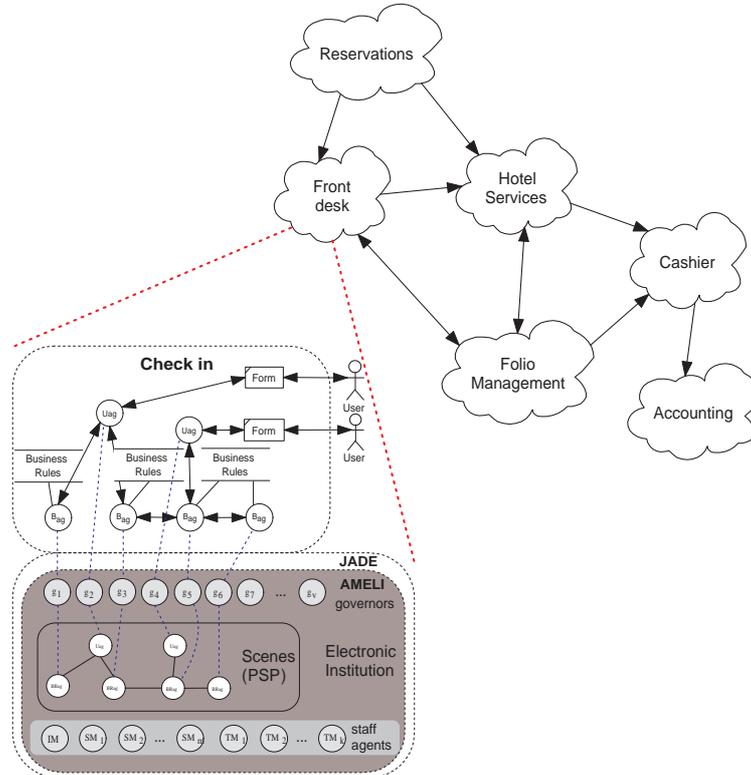


Figure 2: The hotel "Check-in" business context as part of the "Front-desk" sub-system and its corresponding *EI* description. Clouds represent sub-systems than integrate the CIS

An important feature of this association between a business context and an *EI* is that it permits to use various levels of detail in describing a single business context, and also to express alternative definitions of the same and similar business contexts. The devise is to have interchangeable blocks (scenes or performative substructures) that can be *spliced* into the *EI* definition.

The outcome we intend to achieve is to have a systematic way of encapsulating corporate knowledge. A balancing act of due-process and autonomy. We encapsulate corporate know how, on one hand, in the institutional conventions that govern agent interactions and, on the other, by providing rules and corporate knowledge that individual agents have in their internal decision models. The first type defines the corporate procedures all agents need to fulfill, the second allows us to capture policy and directives that some agents should take into account when making an institutional decision. The first hard-wires due process and relieves agents from improvisation, while the second gives grounds for discretional autonomous behavior that can react to uncertain situations or adapt to the changing business context.

Actual implementation, in EIDE, of business context *EIs*, like the "Check-in" example we have been discussing, is straightforward. Joining several context in a single *EI* needs extending the current *EI* definition and its EIDE implementation tools.

## 4.1 Federating business contexts

In order to join the business contexts into a cohesive single *EI* we need to deal with those actions that involve two (or more) business contexts as shown in Figure 3. The typical case is that of the receptionist (in the *Group Check-in* context) who needs to get clearance from the acting maintenance manager (from the *Hotel Services* context) when assigning rooms to an incoming group of guests as recently vacated rooms become available. In this case, each business context should have a scene where that clearance is dealt with.

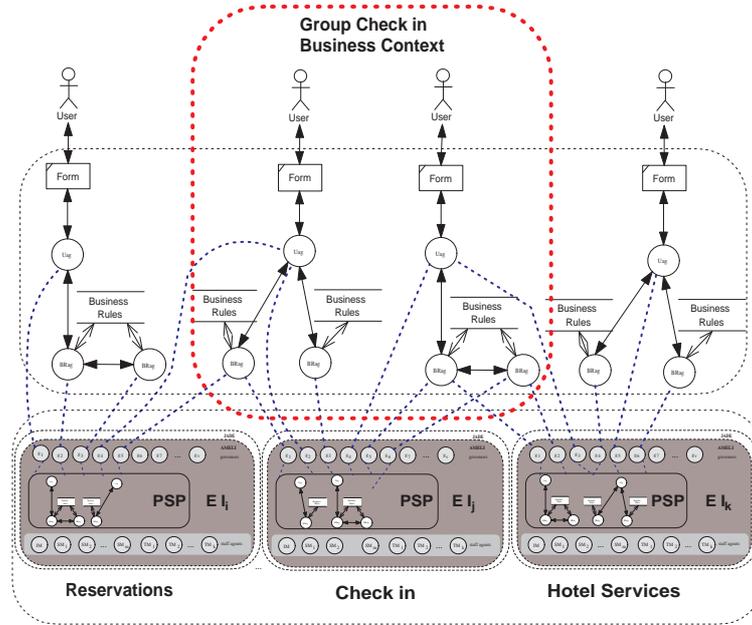


Figure 3: A more complex interaction corresponding to the "Group Check-in" business context in the "Front-desk" sub-system

Although details are beyond the scope of this paper, the actual formal problem involves ontology and illocution alignment which is dealt with through a mapping between institutions. Such mappings involve the alignment of the respective ontologies and transitions (of a new type) that connect scenes where existing illocutions involved a shared obligation. In essence we need to make sure of the singular instantiations of roles and object language terms –

$\langle \text{individual in role } r \text{ in scene } a \rangle \text{ is the same as } \langle \text{individual in role } s \text{ in scene } b \rangle$ ;  
 $\langle \text{entity } j \text{ in business context } x \rangle \text{ is the same as } \langle \text{entity } k \text{ in business context } y \rangle$

– and the states that enable and result from the utterance of the common illocution, are compatible with the states (in the original scenes) where the corresponding (simple) illocutions are being uttered, so that the associated obligations of both contexts are consistent with the obligations of the merged PS.

We refer to that type of cohesive merging of business context as a federation of institutions.<sup>5</sup> As Figure 4 illustrates such a federation may be characterized by the common aligned ontology, the federated institutions and their inter-institutional mappings.

## 4.2 EI grounding

While a connection between an *EI* and the real world had been assumed to exist in the dialogical framework of the current IIIA *EI* model, we now need to make explicit those conventions that guarantee a proper connection between the normative description of the organization and the corresponding operating CIS.

<sup>5</sup>The expression "federated electronic institutions" was used in a not altogether different way to describe a group of fish market sites that allow concurrent remote on-line bidding within the MASFIT project of the IIIA [2].

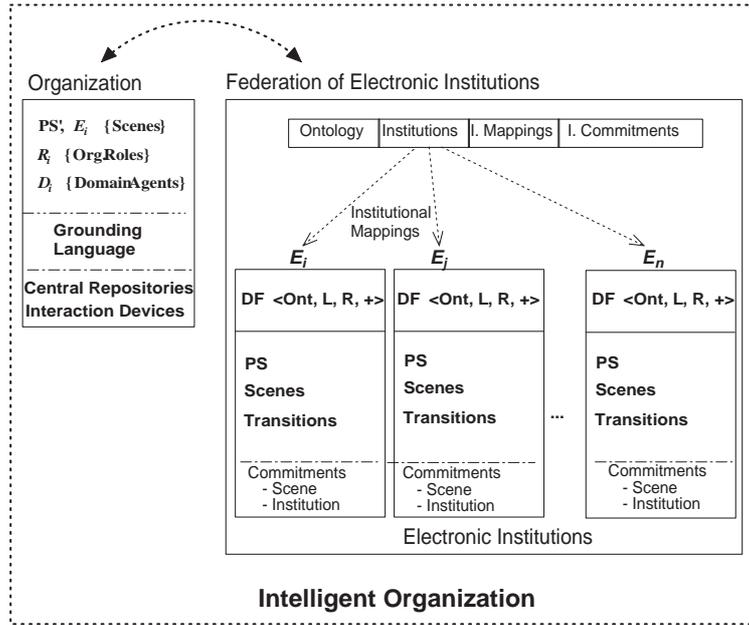


Figure 4: Hotel corporate activity as a federation of electronic institutions each corresponding to a corporate business context

The formal and practical details are beyond the scope of this paper but in general terms we can say that once the different business contexts have been federated into a single *EI* we want that every institutional action corresponds to an action in the business domain. That is, we need an isomorphism from the *EI* onto the CIS such that (i) there is a correspondence between real world entities and entities in the electronic institution, and (ii) every illocution in the *EI* corresponds to an action (transaction, message, display,...) in the CIS. In order to construct this isomorphism we establish a mapping of ontologies (using the grounding language) and build a middleware layer that mediates the translation between institutional terms and illocutionary expressions, on one side and, CIS-server-agent commands of the agentified CIS on the other. Agentification of the CIS amounts to coupling each CIS component (database, problem solving plans repository, display device) with appropriate server agents.

## 5 Current and future work

Our work follows two complementary lines.

**Implementation of the grounding process.** We are taking as much advantage as possible from the ideas and tools that have been used in *EIS*, and *EIDE* in particular, so far. For the agentification of traditional CIS components we have been developing *server agents* that perform institutional tasks. Hence, we have several types of agents each dealing with typical actions on a standard CIS component. For example, a business rule server agent that is in charge of finding and applying business rules that are stored in a business rule repository or a data-base server agent that reads and writes in a relational data base. Likewise for other interface servers.

An important type of server agents we are developing are those that deal with problem solving plans. We want to be able to compose or update business processes (scenes and simple performative structures) in response to various circumstances. For example the implementation of an *EI* from goal-directed descriptions, the on-line update of an *EI* according to policy directives or, likewise, as a reaction to some organizational change.

As a proof-of-concept exercise we are developing a grounding language, server agents and other middleware to MAS-ify the class of straightforward data base CISs.

**Formalizing and implementing *EI* extensions.** Part of our work along this line consists in refining the specification elements that are available in the current *EI* definition and EIDE. In particular, there is intensive and promising work in the development of an EIDE-compatible language for representation of norms that should allow a business domain expert to specify an *EI* in the high level description languages that are common in normative practices.

We are also pursuing theoretical extensions to the current *EI* model. In addition to the formalization of the grounding conventions that we have been mentioning in this paper, we are also involved in developing a framework to operate on the deontological components of *EIs*. In particular we are formalizing operations that permit the "splicing" and "composition" of performative structures and *EIs*. This work involves the definition of federations of *EIs* as we described them here, although the operations are more general. A crude version of these extensions is already implemented in the coming release of EIDE as the notion of a "sub performative structure".

## 6 Related work

For the definition of problem solving plans, in addition to direct ISLANDER specifications, we have also used (*cf.* [11]) *CommonKads* task templates ([12]). Our approach extends well-known agent oriented methodologies, such as GAIA [14] and SODA [9] for the definition of agent properties and interactions.

The OMNI framework proposed by the Utrecht group took a different approach for the definition of organizations [13]. OMNI is a multi-level normative framework that, like ours, uses the notion of electronic institution to specify conventions that guide, control and regulate the behaviour of heterogeneous agents that participate in a closed environment. In the OMNI approach, the authors use abstract norms to handle the organizational conventions. These abstract norms are then translated into concrete norms that are specified in the same type of expressions that they use for the specification of institutional structures and institutional procedures. Those concrete norms need in turn to be translated into electronic institution executable expressions that are eventually applied to agent interactions in order to enforce the abstract conventions in run-time. If when the OMNI proponents are able to implement properly the far from trivial two-stage translations, their approach would produce high-level declarative prescriptions that should achieve a control over agent interactions comparable to that of our framework.

Our approach to organizations, as described in section 2.1, allows us to deal with norms in a more straightforward way and still achieve the type of results they wish to achieve. We are able to express prohibitions, permissions and obligations in a high level deontic language that appears to be general enough to handle typical institutional conventions, but since we are concerned exclusively with interactions within the context of instantiated problem solving plans, we can rely on our grounding language for the interpretation of norms that are defined a priori and are then instantiated through agent interactions that take place in one of those plans. In this way the (EIDE-compatible) electronic institution turns out to be the only entity needed to enforce the norms in our framework. Work in progress along these lines has been described in [4].

## 7 Closing remarks

We intend to build MAS-enabled CIS that are better than traditional CIS and support actual Intelligent organizations. The core of our proposal is the notion of an electronic institution that allows us to establish an isomorphism between a prescriptive model of the organization and the operation of the MAS-ified information components of the organization.

In this paper we give the blueprint of a conceptual architecture. The intended isomorphism is achieved by taking advantage of agent-based systems and extending the current definitions of electronic institutions in two directions. First by composing regular *EIs* into a complex federation. Second, by dealing explicitly with the conventions that connect the *EI* with the real world.

From a design point of view, our proposal allows to escape the hardwiring of forms and work-flows. It provides a new type of flexibility because it allows one to capture organizational knowledge through the procedural conventions on one hand, and within role expertise on the other:

- Procedural conventions are represented in the performative structure and rules of behavior of the *EI*, and are as flexible as the specification and grounding tools may be.
- Role expertise is deposited in agents. One form of encapsulating role expertise is through the choice of decision points of the operative flow that are assigned to internal staff agents that are either human or software agents. Another is by encapsulating knowledge in the internal decision-making mechanisms of intervening software agents. In this way we can get two types of benefits: facilitating specialization and modularity of knowledge, and taking advantage of the autonomy of (in-house) agents to allow deliberative behavior.

Our proposal is intended to be used for the development of new CIS in a (closed) corporate environment. Our experience has told us that it is a costly and risky task to pass directly from a traditional CIS that has been form or work-flow centered programmed to a MAS-enabled CIS. We expect that our framework would eventually facilitate a more acceptable transition, provided CIS components may be coupled with appropriate server agents while facilitates the deployment of new MAS-enabled CIS.

In our proposal we attempt to provide a conceptual framework, tools and a methodology that allows for successive levels of abstraction so that tasks that pertain to different specialties –user, business expert, domain designer, software engineer– may concur smoothly in the design and deployment of complex MAS-enabled CISs. We also intend our framework to support CIS design that achieves modularity of components and enhance their reusability and their portability. By so doing, we would be having MAS-ified CIS that may be modified more easily –ideally even learn and adapt automatically– and therefore be better fit to the dynamic conditions of the business environments.

## Acknowledgment

This research is partially funded by the Spanish Ministry of Education and Science (MEC) through the Web-i-2 project (TIC-2003-08763-C02-00).

## References

- [1] Josep Arcos, Marc Esteva, Pablo Noriega, Juan Rodriguez-Aguilar, and Carles Sierra. Environment engineering for multiagent systems. *Engineering Applications of Artificial Intelligence*, (18):191–204, Elsevier Ltd. January 2005.
- [2] Guifré Cuní, Marc Esteva, Pere Garcia, Eloi Puertas, Carles Sierra, and Teresa Solchaga. Mas-fit: Multi-agent systems for fish trading. In *16th European Conference on Artificial Intelligence (ECAI 2004)*, Valencia, Spain, August 2004.
- [3] Richard M. Cyert and James G. March. *A behavioral theory of the firm*. Prentice-Hall, Englewood Cliffs, N.J. USA, 1963.
- [4] Andres Garcia-Camino, Pablo Noriega, and Juan Antonio Rodriguez-Aguilar. Implementing norms in electronic institutions. In *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2005.
- [5] Jay Liebowitz and Tom Beckman. *Knowledge Organizations*. Saint Lucie Press, Washington, DC, 1998.
- [6] Michael Luck, Peter Mc Burney, Onn Shehory, and Steve Willmott. Agent based computing . Agent Technoloty Roadmap draft, 2005.
- [7] James G. March and Herbert A. Simon. *Organizations*. John Wiley and sons, New York, USA., 1958.
- [8] Douglass C. North. *Institutions, Institutional change and economic performance*. Cambridge Universisy press, 40 west 20th Street, New York, NY 10011-4211, USA, 1990.

- [9] Andrea Omicini. SODA: Societies and Infrastructures in the Analysis and Design of Agent-Based Systems . In P. Ciancarini and M Wooldridge, editors, *Agent Oriented Software Engineering*, volume 1957, pages 185–193, New York, 2001. LNCS, Springer.
- [10] Michael Pechoucek, Donald Steiner, and Simon Thompson. *Industry track of the Fourth International Conference on Autonomous Agents and Multiagent Systems*. ACM press, New York, USA, 2005.
- [11] Armando Robles, Pablo Noriega, Francisco Cantú, and Rubén Morales. Enabling intelligent organizations: An electronic institutions approach for controlling and executing problem solving methods. In *Fourth Mexican International Conference on Artificial Intelligence*, Monterrey, NL, MEX, November 2005.
- [12] Guss Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van der Velde, and Bob Wielinga. *Knowledge Engineering and Management, The CommonKADS Methodology*. MIT Press, Boston, Massachusetts, 2000.
- [13] Javier Vázquez-Salceda, Virginia Dignum, and Frank Dignum. Organizing multiagent systems. 2004. Technical report UU-CS-2004-015.
- [14] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The GAIA Methodology for Agent-Oriented Analysis and Design. In *Autonomous Agents and Multi-Agent Systems*, volume 3, pages 285–312. Kluwer Academic Publishers, 2000.