An Agent Oriented Hotel Information System

Armando Robles
Grupo TCA
Monterrey, NL, México
arobles@grupotca.com

Pablo Noriega
IIIA, CSIC
Barcelona, Spain
pablo@iiia.csic.es

Francisco Cantú
Tecnologico de Monterrey
Monterrey, NL, México
fcantu@itesm.mx

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.11 [**Artificial Intelligence**]: Multiagent systems

General Terms

Industrial Software

Keywords

Agent Oriented Information Systems, Electronic Institutions Applications, Service Oriented Architectures

1. BACKGROUND

The vertical industry software market (for hotels, hospitals, convenience store chains, ...) is highly competitive. To better compete in this environment, Tecnología Computacional Aplicada (TCA) —a mid size vertical market IS development company— decided four years ago to explore agent technologies as a key component for a new business strategy. As a result of this exercise TCA converted its legacy *INNSIST* hotel information system —with 1.2 million lines of code developed over 22 years and a solid customer base— into the fully deployed agent-intensive system whose main features we present here.[1] A previous stage of this project was presented in [2].

1.1 Proposed Framework

We view an organization as a Multiagent System (MAS) with agents playing roles and interacting to achieve individual and collective goals and assume an *institutional perspective* where agents follow some explicit conventions. We build upon the notion of Electronic Institutions (EI) [1] to implement an (extended) electronic institution that states the conventions of the ideal behaviour of the organization and enforces these conventions on the operation of the actual information system (Fig. 1).

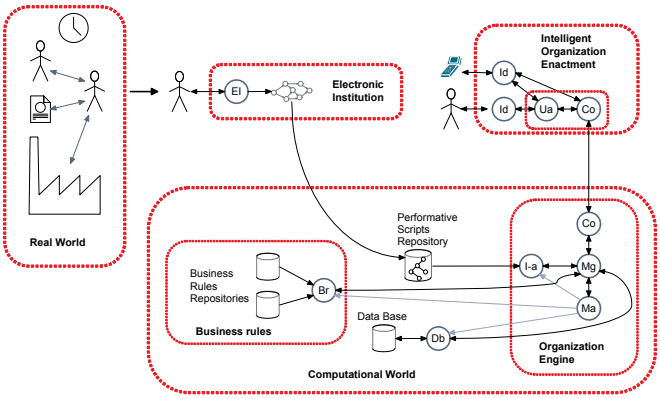

Figure 1: Framework Overview

Conceptual Model.

We keep to the EI model assumptions that all institutional actions are speech acts and the distinction between institutional or internal agents (that belong to the institution) and external agents. However, we extend the current conceptual model of EI with four innovations. (i) A distinction of internal agents that are either (a) *staff agents* that perform decisional tasks that could be performed by employees or (b) *institutional agents* of two main types: *user agents* that handle interface with human users, and *server agents* that handle information system components.[2] (ii) A richer performative structure —new *performative scripts* correspond to scenes that are assembled into atomic EIs and these may be aggregated into more complex EIs— to capture the molecular nature of business processes (iii) A *normative structure* to represent and use declarative conventions for scripts and staff agents, and finally (iv) A structure to handle the notion of *proficiency* that is used for dynamic improvement of conventions.

[1]To use the system access: http://www.innsistondemand.com/en, and follow the options shown in the web-page. Demos are available in that URL and also in http://www.youtube.com, in the "search" text box, enter: innsist, then select the video you want to see: General Overview, Reservations, Front-Desk or Housekeeping. For on-line training, access: http://www.innsistondemand.com/elearning/courses/IE1.1/Home.html, there are 109 practical cases explained in English, Spanish and Portuguese.

[2]There are several kinds of specialized server agents to handle the basic IS resources: data-base, business-rules-repository, etc. User agents control *interaction devices*, used to handle presentation platforms like forms or panels.

Cite as: An Agent Oriented Hotel Information System, Armando Robles, Pablo Noriega, Francisco Cantú, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra andCastelfranchi(eds.),May,10–15,2009,Budapest,Hungary,pp.1415–1416

Conventional IS components like workflows, business rules and forms are translated into performative scripts, institutional agents and rules stored in knowledge bases.

Computational Model.

We propose (i) a *specification language*, that extends ISLANDER, to capture the institutional conventions that are interpreted by (ii) an *organization engine* middleware that, following the workflows specified in the performative scripts, activates user and server agents that use (iii) a *grounding language* that maps illocutory entities into IS entities

In order to make the system agile, we move away from the centralized institutional layer of the current EI model. We do not need *governors* —because all interactions with external agents are mediated through our new *user agents* and consequently all interactions are among internal agents whose own architecture and contents is established by the institution. We also get away from a centrally updated state of the institution, because each performative script spawns and kills all the *server agents* it needs and all business rule variables instantiated in a script enactment are local to the script, hence it is able to control its own *state of interactions*, which is shared with another active script only when the same user agent is active in both scripts.

2. DISCUSSION

Implementation.

The legacy *INNSIST* software was converted into the aforementioned framework in four years by ten software development engineers with different commitment loads. This agentified version of *INNSIST* is currently in use in over one hundred hotels having a daily operation that involves more than one thousand employees.

Actual migration has followed industry standard practices with three particular aspects worth mentioning. First, the new system was re- conceptualized by defining new interaction panels for each sub-system —e.g.,the *front-desk rack* shown in the front desk demo video. Each panel behaves as an *interaction device* linked to an *user agent* and constitutes the only interface with the end-user for a given sub-system. Second, the original source code was grouped by sub-system functionality, building business rules modules sharing contextual information –variables local to all business rules and made available to the institution through its standard data models. Finally, all procedural decisions originally embedded in source code were removed, these procedural decisions are now represented in *performative scripts*, each with an appropriate *business rules module* and the corresponding *institutional agents* whose interactions are thus articulated.

Advantages.

- A new abstraction level for design: capturing all end-user interactions with interaction devices controlled by user agents, separates the concerns and efforts of graphic designers (interaction panels), from those of business experts (business logic) and those of software engineers (procedures and performance).

- Agent role specialization allows the differentiated treatment of different IT components. Hence allowing in-depth treatment of system- performance aspects like form versions, display media, end-user language —handled by *interaction devices* and *user agents*–, different data-base engines –controlled by different engine-specific *data-base server agents*— and non-procedural business logic —codified as *staff agents* and business rules handled with *business-rule server agents*.

- The use of illocutions as the only possible form of interaction facilitates modularization: to assemble the business rule module for a new performative script, one needs only identify the valid illocutions of the script and, then use a *business-rule-repository server agent* to extract from available business rule repositories the business rules involved in those illocutions. Later on, during the script enactment, it spawns only the *business-rule server agents* needed by those rules.

- One major advantage is the possibility of using *staff agents* to perform specialized tasks, including those that might be uneconomical with humans. For example to track server agents response time (to balance loads) or connection quality in user agents (to reactivate unwanted breakdowns).

- Thanks to these developments, the *INNSIST* site presented in this paper is already hosted in servers in Florida, California, London and Mexico. These sites are ready to support regional demand. Currently the system is available in English, Portuguese and Spanish and tailoring to particular regional requirements is only a matter of bringing further specialization to a few of the already available server agents and performative scripts.

Finally.

In addition the agentified *INNSIST* hotel information system, this experience has resulted in a new business model as well as the incorporation of components, methodologies and tools in all TCA products and services. These developments have given TCA a competitive edge and we are convinced that the agent-based framework we propose is adequate for the conversion as well as the design and deployment of large scale, human interaction intensive, web-based industrial information systems.

3. ACKNOWLEDGMENTS

Research was partially funded by projects: IEA (*TIN2006-15662-C02-01*), AT (*CONSOLIDER CSD2007-0022, INGENIO 2010*) and TCA Research Group's private funds.

4. ADDITIONAL AUTHORS

Additional authors from TCA:
Marco Julio Robles, Aida Carolina Robles
{mjrobles,crobles}@tcasoftwaresolutions.com,
Arnoldo Rodriguez, Hector Hernandez, Edgar Gutierrez
{arodriguez,hhernandez,egutierrez}@grupotca.com.

5. REFERENCES

[1] J. L. Arcos, M. Esteva, P. Noriega, J. A. Rodríguez, and C. Sierra. Engineering open environments with electronic institutions. *Journal on Engineering Applications of Artificial Intelligence*, 18(2):191–204, 2005.

[2] A. Robles, P. Noriega, M. Luck, and F. Cantú. Using mas technologies for intelligent organizations: a report of bottom-up results. volume 4293, pages 1116–1127. Springer, Springer, 2006.