# DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

*GOVERNMENTAL VIRTUAL INSTITUTIONS*

Claudia J. Abro de Arajo and
Flávio S. Corrêa da Silva

Março de 2009

# Governmental Virtual Institutions

Claudia J. Abro de Arajo

*Laboratory of Interactivity and Digital Entertainment Technology*
*University of Sao Paulo*
*Sao Paulo, BRAZIL*
claudiaj@ime.usp.br

Flavio S. Correa da Silva

*Laboratory of Interactivity and Digital Entertainment Technology*
*University of Sao Paulo*
*Sao Paulo, BRAZIL*
fcs@ime.usp.br

June 2, 2009

## Abstract

Virtual Worlds are very popular media for social interaction and we believe that the adoption of this media is suitable for Electronic Government applications. It can increase the capillarity of public services, facilitate the access to (and execution of) government services and provide citizens with a natural and immersive experience. In the present paper we introduce a Government Virtual Institution Model that satisfies relevant issues such as: user friendliness to citizens with diverse education levels; facilitated connection between heterogeneous government systems; satisfaction of government services requirements related to security, privacy, reliability, scalability and interoperability. The government services and the information flow across the Government Virtual Institution are formally described using LCC (the Lightweight Coordination Calculus) language, and the model specifies the use of the JamSession decentralized architecture for virtual worlds.

*Keywords: Electronic Government, Virtual Worlds, Artificial Intelligence, Intelligent User Interfaces, Interoperability, Information Representation*

# 1 Introduction

Virtual worlds constitute a highly popular media for general purpose social interaction, as well as for purpose oriented task execution. The tasks that can

be effected through virtual worlds can be related to activities such as entertainment, electronic commerce, education, culture, and the provision of services e.g. related to electronic government.

Relevant technologies to support the interaction through virtual worlds include multi modal interaction, and the related field of semantics of data that can be provisioned by heterogeneous sensors and actuators, to allow a natural – and therefore more immersive – interaction experience for users through virtual worlds.

Using a rather technical terminology, this can improve the *affordance* of the virtual worlds [1], i.e. the extent to which the experience of being in the virtual worlds is natural and intuitive, together with the extent to which the actions of users in these virtual worlds is guided to belong to a controlled set of actions, in such way that the users do not feel constrained in their attitudes.

We propose to employ virtual worlds as a natural and immersive interface for the electronic government. This paper presents a Government Virtual Institution Model for the provision of public services, that satisfies relevant issues such as:

- citizens education level;

- heterogeneous government systems;

- government services requirement for security, privacy, reliability and scalability

- government interoperability requirements.

We believe that independently of the level of citizens education, virtual worlds provides an easy interface for interaction and facilitates the learning process to use a public service system.

On the govern side, the system requirements are an important issue to address, before effectivelly provide a public service. In order to satisfy these requirements, we propose to adopt the JamSession decentralized architecture for virtual worlds. This architecture provides support to build special purpose virtual worlds, ensure reliability, robustness and security, also managing the required computational resources effectively.

The heterogeneity of the government systems is respected in this architecture both, on the back-end and on the front-end, where diferent kinds of system interface may be delivered.

To provide the correct execution of the system, the government services and the information flow are formally described using a LCC (Lightweight coordenation calculus) language. The government interoperability requirements are also specified in LCC.

Finally, we belive this model will contribute to increase the capillarity of public services, providing users with an immersive experience in a rather familiar environment, avoiding the need of them to physically travel to a governmental office, also avoiding the need of maintaining too many governmental offices.

2

In the present article we describe in detail the GVI model. In section 2, we present an overview of Eletronic Government. In section 3 we briefly review the Lightweight Coordination Calculus, in section 4 we introduce the JamSession Architecture and in section 5 we employ these concepts to present the GVI model. Finally, in section 6 we present some conclusions about this work.

## 2 Electronic Government

The Internet has proven to be an important tool to increase the capillarity of public services. An important issue to be taken into account, however, is that such services are directed to citizens whose levels of education can vary significantly. Hence, some recipients of these services may not be acquainted with digital technologies and with the software interfaces that are most commonly employed to access these services.

We propose to employ virtual worlds as a natural and immersive interface for the electronic government, simulating situations for users that they would live through if they were physically present at an office and attended by a human clerk. A user can navigate across a virtual world, interact with a synthetic character that simulates a human clerk and/or with avatars of governmental officers, get his/her tasks accomplished and reach his/her goals, experiencing situations that are familiar to him/her and yet avoiding the need to physically travel to a governmental office – and, in many cases, the need to count on an officer for personal assistance at the other end of the line. Figure 1 ilustrates such a situation, considering the example of issuing a document.



Figure 1: Simulation of a real situation

Many services offered through electronic government systems involve the exchange of sensitive information (e.g. bank transfer records, document numbers, personal identification codes, personal address and telephone number, etc.).

To deal with that, we define a model of Govenmental Virtual Institution (GVI) that presents a formal model of information representation, service modeling and governmental interoperability rules. We use the Lightweight Coordi-

3

nation Calculus (LCC) [2] to formally model, simulate and control the services and interoperability rules relevant that are relevant to electronic government.

The GVI model defines interactive systems to be built on the JamSession architecture, satisfying all necessary requirements and ensuring security and reliability in transactions.

# 3 Lightweight Coordination Calculus (LCC)

The Lightweight Coordination Calculus (LCC) is a process calculus for specifying social norms and was designed for expressing P2P interactions within multiagent systems [3]. LCC also permits simple but powerful mechanisms for analysis, simulation and deployment of interactive systems.

LCC can be used to specify, verify and execute interaction protocols. The specification of interaction protocols makes use of the formal syntax of LCC to build interaction models, which can be intuitively seen as templates that, once fulfilled by appropriate agents, processes and parameter values, can be performed as scripts of actions. Interaction models can also be formally verified, in order to ensure the satisfaction of desired properties.

LCC is employed in GVI to encode interaction protocols in two different senses:

1. Interaction protocols are defined to *guard* the behavior of objects in connected and synchronized virtual representations of agencies, so that undesired or unexpected behavior of peers can be identified and treated accordingly in real time.

2. Interaction protocols are also defined to encode the interactions between characters, services and simulations of physical objects in virtual agencies, to ensure the usability and the affordance in these representations of real agencies.

In a Governmental Virtual Institution, LCC is employed to encode govern interoperability protocols.

The syntax of LCC is presented in Figure 2:

- *null* denotes an event which does not involve message passing.

- *Term* is a structured term in Prolog syntax.

- *Id* is either a variable or a unique identifier for the agent.

- Operators ←, ∧ e ∨ are the normal logical connectives for implication, conjunction and disjunction.

- ⇒ represents a message being sent.

- ⇐ represents a message being received.

4

$$Framework ::= \langle Clause, ... \rangle$$
$$Clause ::= Agent :: Def$$
$$Agent ::= a(Type, Id)$$
$$Def ::= Agent \mid Message \mid Def \ then \ Def \mid Def \ or \ Def \mid$$
$$Def \ par \ Def \mid null \leftarrow C$$
$$Message ::= M \Rightarrow Agent \mid M \Rightarrow Agent \leftarrow C \mid$$
$$M \Leftarrow Agent \mid M \Leftarrow Agent \leftarrow C$$
$$C ::= Term \mid C \wedge C \mid C \vee C$$
$$Type ::= Term$$
$$M ::= Term$$

Figure 2: LCC Syntax

An agent is represented as $a(Type, Id)$, identifying the agent ($Id$) and its role ($Type$).

*Def* is the definition of interactions between agents, and it can be composed as sequences, choices or in parallel. It also may be have a precondition ($C$).

The language provides two primitive operations to access variables: *get* and *put*. Finally, the *timeout* introduces some time management into the language.

To give a flavor of how LCC works, we show in Figure 3 a very simple interaction protocol. This small example shall suffice, nevertheless, to provide an intuitive overview of LCC and suggest how it is used in GVI.

1.   $a(r_1, N_1) ::$
2.       $(req(X) \Rightarrow a(r_2, N_2)) \leftarrow need(X) \ then$
3.       $prov(X) \Leftarrow a(r_2, N_2)$

4.   $a(r_2, N_2) ::$
5.       $req(X) \Leftarrow a(r_1, N_1) \ then$
6.       $prov(X) \Rightarrow a(r_1, N_1)$

Figure 3: An Interaction Protocol in LCC

We employ the usual PROLOG convention for names of variables and terms,

namely capital letters indicate variables (i.e. placeholders for values) and small letters indicate constant terms. The terms $r_1$ and $r_2$ are the names of the roles that are being specified in this interaction protocol. This protocol requires the availability of two different agents, to assume each one of these two roles, in order to be fulfilled. The first agent shall have his name unified with $N_1$, and the second agent shall have her name unified with $N_2$.

Agent $N_1$ is prepared to assume role $r_1$ if he needs an object that can be unified with $X$, in which case a request is sent to agent $N_2$. If this clause can be satisfied, then agent $N_1$ will be satisfied if $N_2$ can provide $X$ to it.

Agent $N_2$, in turn, is prepared to assume role $r_2$ if he receives a request of $X$ from agent $N_1$. If this happens, then agent $N_2$ provides $X$ to agent $N_1$.

Lines 1 and 4 specify templates that can be fulfilled by specific concrete agents, provided that they are prepared to assume their corresponding roles. Line 2 is a clause, whose condition is $need(X)$, stating that agent $N_1$ needs the object $X$, and whose consequence is a request addressed to agent $N_2$. Double arrows denote message passing, and single arrows denote logical consequence. Line 3 is also a clause, whose condition is the provision of $X$ by $N_2$, and whose consequence is a state of satisfaction for $N_1$. The word "then" at the end of line 2 indicates that the clauses must be satisfied sequentially.

Similarly, line 5 denotes that agent $N_2$ receives a request from $N_1$, and then provides the requested object to $N_1$.

In the case of JamSession, if this interaction protocol is fulfilled, then a sequence of events can be added to the island(s) where $N_1$ and $N_2$ are located, which are then added to their queues of events accordingly.

When two characters stand up as candidates to fulfill this interaction model, they submit a proposal to assume their corresponding roles. For example, *john* and *mary* can be candidates, respectively, to assume roles $r_1$ and $r_2$. The constraints characterized by the predicates *need*, *req* and *prov* are then tested, upon proper instantiation of the variable $X$, and, all things going well, the model is fulfilled and scripts of events can be generated and addressed to the queues of events that govern the behavior of the corresponding characters in the virtual worlds where they are rendered.

# 4   The JamSession Architecture

Virtual worlds constitute a highly popular media for general purpose social interaction, as well as for purpose oriented cooperative task execution, in which a group of individuals – who may be geographically dispersed – share the digital simulation of a physical environment to cooperate in the achievement of a goal.

The tasks that can be effected through virtual worlds can be related to activities such as entertainment, electronic commerce, education, culture, and the provision of services e.g. related to electronic government. Relevant technologies to support the interaction through virtual worlds include multi modal interaction, and the related field of semantics of data that can be provisioned by heterogeneous sensors and actuators, to allow a natural – and therefore more

immersive – interaction experience for users through virtual worlds.

The wide adoption of this type of media has brought to attention the need for better support to:

- build special purpose virtual worlds more easily;

- ensure the reliability, robustness and security of these worlds; and

- manage the required computational resources to build, deploy and access virtual worlds effectively.

These features are required so that the adoption of virtual worlds for specialized tasks can become more widespread, including tasks that can be sensitive to features such as privacy, security and reliability.

In this section we introduce JamSession, a decentralized architecture for guarded virtual worlds that adds on desirable features related to all issues referred to above.

JamSession is an architecture for the exchange, communication and sharing of software applications based on virtual worlds. It is decentralized and based on the notion of islands. An island is hosted by a computational device connected to the Internet. When two or more islands are connected, they accept the mediation of a *router*, which is typically located together with one of the islands and coordinates the synchronization of the connected islands. Each island contains a queue of events that must occur within itself, and synchronization is based on the exchange of a minimal set of messages that ensures consistency among all queues of events. Hence, events do not necessarily occur simultaneously in all islands, but the sequence of events occurring in each of them is equivalent. Once a set of islands is connected and synchronized, the islands can virtually *see* each other through gateways. Physically, this is done by replication of all islands that are connected and synchronized in each computational device that hosts each one of the synchronized islands.

Upon mutual agreement, two islands in JamSession can exchange objects, i.e. the simulation of physical objects, characters or processes that embody services can migrate through a gateway across islands, thus enabling the interaction of objects whose origin is based on different islands, and creating the capabilities to build the purpose oriented cooperations for task execution. Physically, this is done by the dynamic update of the minimal set of messages that is exchanged between synchronized virtual worlds, as well as by the actual migration of mobile objects across the Internet.

JamSession is fully decentralized, and therefore does not require a full fledged network server to host the virtual worlds. Alternatively, computational devices with different capabilities – including desktops, smartphones, mobile and low cost equipment – can host *islands*, i.e. simulations of bounded spaces in which specific characters, services, objects and events can be found, which can be accessed through the Internet by other computational devices which can also have heterogeneous capabilities, and which can in turn host their own islands.

JamSession advances on the features presented above, by adding to it a higher level of abstraction that enables the guarded construction of islands, which are ensured to be reliable, robust, secure and efficient. Moreover, the guarded construction of islands is based on a conceptual language, whose basic terminology varies depending on the specific tasks at hand, in such way that it becomes more natural and user friendly.

The conceptual language for the guarded construction of islands in JamSession is based on LCC [2]. Using LCC, we can build *interaction models*, which are formal expressions that characterize templates of interactions among groups of agents. Once we find agents and conditions that fulfill an interaction model, the sequence of events that comprise that interaction can be effected.

Interaction models are used in two different ways in JamSession: at an object level, they characterize the concrete sequences of events that comprise the actions that must be coordinated in order to perform a task. At a system level, interaction models are also used to effectively *guard* the interactions between synchronized islands, ensuring that ill behaved objects do not migrate across different computational devices.

The motivation to build JamSession has been to provide users with a friendly, robust and secure environment to build and deploy services through the Internet. JamSession has been carefully engineered to run on a variety of computational devices and platforms. Using JamSession, users will be able to publish their ideas and systems easily and using low cost resources and equipment.

JamSession is still a work in progress. The JamSession architecture has been designed in detail, and some programming experiments have been developed to ensure that this architecture can be implemented effectively, see section 4.2 below for further details. In the very near future the first complete toolset will be deployed for users to build their own islands and publish them on the Internet. The JamSession project is committed to the release of open source software, and the JamSession architecture shall be freely available on the Internet.

## 4.1   The JamSession Architecture Components

JamSession is made of independent islands that are not replicated upon connection. In order for two islands to be connected, they must be *compatible* with each other. Compatible islands refer to the same virtual space, and their contents are ensured to be consistent with each other. This way, we avoid the traffic of multimedia contents across the Internet, to improve the efficiency and the security of JamSession.

Connected islands in JamSessoion are mediated by a router that maintains the consistency between their queues of events. This also guaranties synchronization of JamSession islands.

A JamSession island is comprised by a set of components, as depicted in Figure 4. The components are as follows:

- **VW**: virtual world (island). Virtual worlds in JamSession exchange messages with a **Router**, in order to keep synchronized with each other.
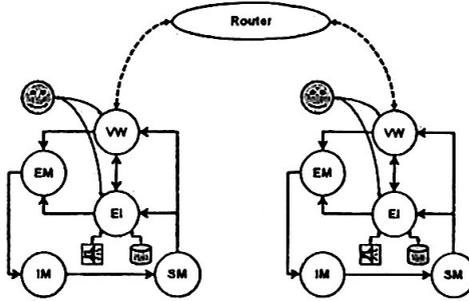
8

Figure 4: The Components of a JamSession Island

- **EI**: external interface, that interacts with a variety of computational devices, such as I/O devices and file storage devices. Both external interfaces and virtual worlds are visible to users. They comprise the presentation interface of JamSession. External interfaces and virtual worlds also identify relevant events that occur within themselves, and inform the occurrence of these events to the event manager EM.

- **EM**: event manager, that keeps track of relevant events occurring in virtual worlds and registered by external interfaces, contains a table of entry points for the existing interaction models and, when a set of concurrent events allows the fulfillment of an interaction model, matches the events with the corresponding interaction model and triggers the interaction.

- **IM**: database of interaction models, that is triggered by the event manager whenever an interaction model can be fulfilled. When this happens, the corresponding interaction model is run and a script of events is built and dispatched to the script manager SM. Interaction models are written in LCC.

- **SM**: script manager, that receives scripts from completed interaction models, turns them into series of events, and updates the local queues of events of **VW** and EI accordingly. Once these queues of events are updated, the Router synchronizes the queues that are connected, in order to preserve consistency across islands.

## 4.2   Some Virtual Worlds That Can Be Based On JamSession

The JamSession architecture has been through conceptual testing on a variety of domains and applications, in order to assess its flexibility and genericity, as well as whether it is truly becoming user friendly, robust and secure.

Each domain demands special attention to a specific subset of the components of the JamSession architecture, as discussed below.

Besides Electronic Government, the domains and applications that have been analyzed in the JamSession project to date are:

- Ambient Intelligence: ambient intelligence (*aka* smart environments) can relate to virtual worlds in at least two senses:

  1. a virtual world – which can be visible or invisible to end users, who inhabit a smart environment in this case – can mediate and manage the services provided by sensors and actuators in the environment. The virtual world, in this case, can simulate the physical environment, and provides a set of support services to enable the expected behavior of the physical environment.

  2. a virtual world can be itself the location for interactions among agents and computational services. The provision of services and the availability of interactions can be, depending on the application, influenced by the location, actions and gestures of the users. In this case, the roles of the physical and the virtual spaces contrast with the previous case, since it is the physical environment that provides support services to enable the expected interactions within the virtual environment.

  Of course, both possibilities can be combined in sophisticated applications, e.g. based on augmented reality, virtual reality, ambient intelligence *per se* and mixed reality.

  Any of these possibilities requires that the external interface interacts with sensors and actuators, in order to interpret information gathered from sensors and address appropriate messages to actuators. Hence, also in this type of applications we need to build sophisticated external interfaces to enable the implementation of appropriate JamSession islands. Contrasting with the previous sort of applications, however, if we have an asymmetry of roles between peers, in this case it is the external interface that sits on the client side of the connection that requires special attention, since this is the external interface that connects directly with the devices in the physical environment, considering that the client side of the connection comprises the computational devices that are physically placed together with the users who inhabit the smart environment.

  Digital Entertainment and Interactive Arts: in the JamSession project, we have given great attention to the design and implementation of synthetic characters, i.e. autonomous agents who share the virtual space with avatars controlled by human users. Such characters are of particular interest for applications related to digital entertainment, in which it is desired that automated characters, whose control is fully determined by computer programs, become similar and equivalent in all possible senses to human controlled characters.

10

We have worked on experiments to build synthetic characters presenting a variety of distinctive features, such as:

- synthetic characters whose behavior rules evolve with interactions;
- synthetic characters whose behavior is controlled by features of personality and emotion; and
- synthetic characters who are capable of generating humorous utterances autonomously.

The implementation of these distinctive features could be encoded in a variety of ways. In order to preserve the structure proposed in the Jam-Session architecture, we have disciplined ourselves to encode these features as interaction models. The implementation of the behavior of characters presenting the features above, therefore, has been devised as a collection of interaction models that are available to specific characters, who then satisfy the corresponding models whenever necessary and receive the corresponding scripts of actions to enact the corresponding interactions.

The construction of synthetic characters for digital entertainment – including interactive arts, such as interactive narratives and some sorts of performative arts – requires, therefore, that relatively sophisticated interaction models are built, with their corresponding event managers and script managers featuring compatible levels of sophistication, since they shall respectively provide the interaction models with sensed events that can trigger specific interactions and receive scripts that shall be translated into appropriate entries into queues of events in the synchronized virtual worlds.

# 5   The Governmental Virtual Institution

We use the concept of Virtual Institution as defined in [4]: *Virtual Institutions are 3D Virtual Worlds with normative regulation of interactions*. They propose to separate the development of Virtual Worlds based on the concept of Virtual Institutions into two independent phases: specification of the interaction rules and design of the 3D Interaction environment.

In GVIs this separation also exists, and has the advantage of making the specification of the interaction rules independent of particular Virtual Worlds technology used for the visualization of the system. It alows a quick and easy portability to new visualization platforms and diferent kinds of renderization. The specification of the interation rules and specifically the governmental interoperability protocol are written in LCC.

GVI assumes a distributed architecture, consisting on an interconnected network of virtual worlds where each node is a virtual governmental agency representative or an agency's department (see Figure 5). Users interact with agencies through their avatars, and agencies exchange information based on interaction protocols implemented using LCC and knowledge representation techniques,

11

such as ontologies and reasoning mechanisms based on formal logics. Different groups of services must be required by interaction protocols with different features and specificities.

GVIs do not have to worry about fundamental issues such as security, privacy, reliability, scalability and interoperability as they are guaranteed by the JamSession architecture.
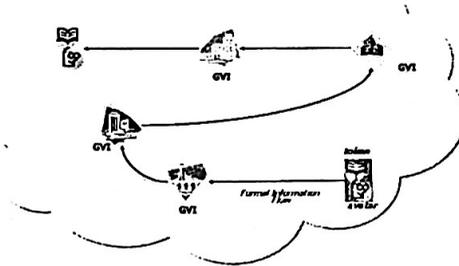


Figure 5: Virtual Govenment Institution

## 5.1   Simple Example: Document Issuing

To illustrate GVI, LCC and JamSession in action, we show in Figure 6 a simplified scenario related to document issuing in the context of electronic government. In this scenario, the external interface interacts with the database *system*, through messages coined *exists* and *register*, respectively to check whether a *birth* date is a valid entry in the database *system* and to update that database.

---

1.   $a(citizen, ID1) ::$
2.      $(need(DOC) \Rightarrow a(govagent, ID2))$
          $\leftarrow has(DOC, birth)$ then
3.      $issue(DOC) \Leftarrow a(govagent, ID2)$

4.   $a(govagent, ID2) ::$
5.      $(need(DOC) \Leftarrow a(citizen, ID1)) \leftarrow$
          $exists(birth, system)$ then
6.      $(issue(DOC) \Rightarrow a(citizen, ID1)) \leftarrow$
          $register(DOC, birth, system)$

---

Figure 6: Document Issuing in LCC

The components in JamSession for this scenario are devised as in Figure 7.

12

When a character assumes the role of *citizen*, it must check its birth date and then present the corresponding request for a document, *DOC*. This must be informed to a second character who must have assumed the role of *govagent*. The character who has assumed the role of *govagent* can, in turn, receive the request for the document. It verifies in the external database *system* whether the corresponding birth date exists. If it does, then it can issue the requested document and register the transaction in the external database.

Figures 6 and 7 shows how intuitive is the processing of the LCC protocol definition in the JamSession architecture components.
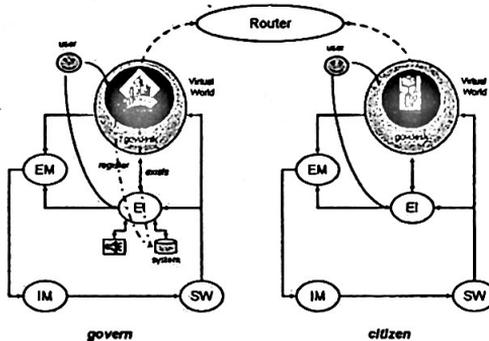


Figure 7: Issuing a Document in JamSession Architecture

Note that *govern* and *citizen* are two separated JamSession islands and *Router* is the mediator. They may be hosted anywhere that has access to the Internet. The two islands have all the JamSession components described in Section 4.1, but only the *govern* has a database (*system*) connected to the External Interface component (EI). The *citizen* island has no access to such a database. As the objective of the GVI is to provide an immersive experience to the users, our proposal of user interface is a 3D virtual world. However, to guarantee the capilarity of service and respect govern system heterogeneity, users may also see a 2D interface, without any loss of functionality for the system.

Security and reliability in transactions, whose steps must be carefully recorded in the databases that are connected to JamSession as well as to governmental information systems, is a prominent issue for this sort of applications. In the JamSession architecture, these features are dealt with by the *external interface*, which must be particularly well engineered in this case. Considering that there can be an assymmetry between peers in JamSession, in which a computational device functions as the provider of a service and another connected device functions as a client (for example, the provider can contain a local database of transactions, and connection to secured external sources of information about the whole population, whereas a client may contain solely connection to a personal database of the history of transactions of a single user), the external

13

interface that must be particularly well crafted in order to ensure the required quality of services for this sort of applications is the one hosted by the service provider.

# 6 Conclusions

In JamSession project, the slogan is: *complexity at the service of simplicity*, since the idea is to employ state-of-the-art technology to make the construction and the utilization of virtual worlds as simple and friendly as possible.

The motivation for that comes from the desire to provide end users with a set of tools that can be used to build and deploy sophisticated information services through the Internet, as easily as possible, freely and independently, with the potential to turn these services into profitable business.

In GVI project, we have the same motivation of JamSession, and the same ambition of being able to collaborate with initiatives for economic inclusion that can enrich the world economy as well as significantly improve the quality of life of many individuals, as for example the initiatives of Mohammad Yunus [5].

Also, we aim to increase the capillarity of public services, facilitate the access (and execution) to government services and provide citizens with a natural and immersive experience.

Our work has focused on the definition of a model of Governmental Virtual Institution that will facilitates the construction of a variety of government applications, whose central metaphor for interactions is based on participation in virtual worlds, which are physically implemented based on a decentralized model.

To date, we have a relatively mature conceptual design of what GVI shall be. We also have the results of some experiments that confirm that it can be implemented as planned.

Future work includes, evidently, the full fledged implementation of a governmental application using GVI, with a set of services modeled as described in the previous sections.

# 7 Acknowledgments

# References

[1] D. Norman. *The Design of Everyday Things*. Basic Books, USA, 2002.

[2] D. Robertson. A lightweight coordination calculus for agent systems. In *Declarative Agent Languages and Technologies II*. Springer Lecture Notes in Computer Science 3476, 2005.

[3] C. Walton and D. Robertson. Flexible multi-agent protocols. In *Procs. of UKMAS 2002*. UKMAS, 2002.

[4] B. Anton. Virtual Institution. In *PhD. Thesis*. Sydney, Australia, 2007.

[5] M. Yunus. *Creating a World Without Poverty: Social Business and the Future of Capitalism*. Public Affairs, USA, 2008.

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**
Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 2003 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail (mac@ime.usp.br).

Carlos H. Cardonha, Marcel K. de Carli Silva e Cristina G. Fernandes
*COMPUTAÇÃO QUÂNTICA: COMPLEXIDADE E ALGORITMOS*
RT- MAC 2005-01 – janeiro 2005

C.E.R. Alves, E. N. Cáceres and S. W. Song
*A BSP/CGM ALGORITHM FOR FINDING ALL MAXIMAL CONTIGUOS SUBSEQUENCES OF A SEQUENCE OF NUMBERS*
RT- MAC- 2005-02 – janeiro 2005

Flávio S. Corrêa da Silva
*WHERE AM I? WHERE ARE YOU?*
RT- MAC- 2005-03 – março 2005, 15pp.

Christian Paz-Trillo, Renata Wassermann and Fabio Kon
*A PATTERN-BASED TOOL FOR LEARNING DESIGN PATTERNS*
RT- MAC – 2005-04 – abril 2005, 17pp.

Wagner Borges and Julio Michael Stern
*ON THE TRUTH VALUE OF COMPLEX HYPOTHESIS*
RT- MAC – 2005-05 – maio 2005, 15 pp.

*José de Ribamar Braga Pinheiro Jr., Alexandre César Tavares Vida and Fabio Konl*
*IMPLEMENTAÇÃO DE UM REPOSITÓRIO SEGURO DE APLICAÇÕES BASEADO EM GSS – PROJETO TAQUARA*
RT- MAC – 2005-06 – agosto 2005, 21 pp.

Helves Domingues and Marco A. S. Netto
*THE DYNAMIC DEPENDENCE MANAGER PATTERN*
RT – MAC 2005-07 – dezembro 2005, 12 pp.

Marco A. S. Netto, Alfredo Goldman and Pierre-François Dutot
*A FLEXIBLE ARCHITECTURE FOR SCHEDULING PARALLEL APLICATIONS ON OPPORTUNISTIC COMPUTER NETWORKS*
RT- MAC 2006-01 – Janeiro 2006, 18 pp.

Julio M. Stern
*COGNITIVE CONSTRUCTIVISM AND LANGUAGE*
RT – MAC 2006-02 – Maio 2006, 67 pp.

Arlindo Flávio da Conceição and Fabio Kon
*EXPERIMENTS AND ANALYSIS OF VOICE OVER IEEE 802.11 INFRASTRUCTURED NETWORKS*
RT – MAC 2006-03 – Junho 2006,

Giuliano Mega and Fabio Kon
*DISTRIBUTED SYMBOLIC DEBUGGING FOR THE COMMON PROGRAMMER*
RT – MAC 2006-04 – Junho 2006

Pedro J. Fernandez, Julio M. Stern, Carlos Alberto de Bragança Pereira and Marcelo S. Lauretto
*A NEW MEDIA OPTMIZER BASED ON THE MEAN-VARIANCE MODEL*
RT – MAC 2006-05 – Junho 2006, 24 pp.

P. Feofiloff, C.G. Fernandes, C.E. Ferreira and J.C. Pina,
*"A NOTE ON JOHNSON, MINKOFF AND PHILLIPS' ALGORITHM FOR THE PRIZE-COLLECTING STEINER TREE PROBLEM"*
RT-MAC2006-06 – Setembro 2006, 11 pp.

Julio Michael Stern
*DECOUPLING, SPARSITY, RANDOMIZATION, AND OBJECTIVE BAYESIAN INFERENCE*
RT-MAC2006-07 – Novembro 2006, 36 pp.

Cristiane Maria Sato, Yoshiharu Kohayakawa
*ENTROPIA DE GRAFOS*
RT – MAC2006-08 – Dezembro 2006 , 44 pp.

Julio M. Stern
*LANGUAGE, METAPHOR AND METAPHYSICS: THE SUBJECTIVE SIDE OF SCIENCE*
RT-MAC-2006-09 – Dezembro 2006, 35 pp.

Thiago A. de André and Paulo J. S. Silva
*EXACT PENALTIES FOR KKT SYSTEMS ASSOCIATED TO VARIATIONAL INEQUALITIES*
RT-MAC-2007-01– Março 2007, 21 pp.

Flávio Soares Correa da Silva, Rogério Panigassi and Carlos Hulot
*LEARNING MANAGEMENT SYSTEMS DESIDERATA FOR COMPETITIVE UNIVERSITIES*
RT-MAC-2007-02 – Maio 2007, 12 pp.

Alexandre Freire da Silva, Fabio Kon, Alfredo Goldman
*THREE ANTI-PRACTICES WHILE TEACHING AGILE METHODS*
RT-MAC-2007-03 – Maio 2007, 20pp.

Silvio do Lago Pereira e Leliane Nunes de Barros
*PLANEJAMENTO BASEADO EM PROCESSOS DE DECISÃO MARKOVIANOS*
RT-MAC-2007-04 – Maio 2007, 17pp.

Silvio do Lago Pereira e Leliane Nunes de Barros
*DIAGRAMAS DE DECISÃO BINÁRIA*
RT-MAC-2007-05 – Maio 2007, 16pp.

Carlos Alberto de Bragança Pereira and Julio Michael Stern
*AN ESSAY ON THE ROLE OF BERNOULLI AND POISSON PROCESSES IN BAYESIAN STATISTICS*
RT-MAC-2007-06 – Junho 2007, 39pp.

Flávio Soares Corrêa da Silva
*ARGUMENTS IN FAVOR OF A CONTROLLED PLURALITY OF OFFICE FORMATTING STANDARDS*
RT-MAC-2007-07 – Junho 2007, 9pp.

Silvio do Lago Pereira, Leliane Nunes de Barros and Fábio Gagliardi Cozman
*STRONG PROBABILISTIC PLANNING*
RT-MAC-2007-08 – Junho 2007, 25pp.

Silvio do Lago Pereira and Leliane Nunes de Barros
*FORMALIZING PLANNING ALGORITHMS FOR TEMPORALLY EXTENDED GOALS*
RT-MAC-2007-09 – Junho 2007, 22pp.

Flávio S. Corrêa da Silva
*THEPROLOGPLAY: AN INTERACTIVE VIRTUAL ENVIRONMENT FOR ARTIFICIAL INTELLIGENCE AND RELATED FIELDS*
RT-MAC-2007-10 – Setembro 2007, 12pp

Vanessa Sabino and Fabio Kon
*LICENÇAS DE SOFTWARE LIVRE HISTÓRIA E CARACTERÍSTICAS*\*
RT-MAC-2009-01 - Março 2009, 40pp.

Alexandre Noma, Ana B. V. Graciano, Roberto M. César Jr., Luis A. Consularo and
Isabelle Bloch
*INEXACT GRAPH MATCHING FOR SEGMENTATION AND RECOGNITION OF OBJECT PARTS*
RT-MAC-2009-02 – Março 2009, 31pp.

Claudia J. Abro de Arajo and Flávio S. Corrêa da Silva
*GOVERNMENTAL VIRTUAL INSTITUTIONS*
RT-MAC-2009-03 – junho 2009, 19pp.