TAILORABLE E-GOVERNMENT INFORMATION SYSTEMS

George Lepouras, Anya Sotiropoulou, Dimitrios Theotokis, Costas Vassilakis
Department of Computer Science and Technology,
University of Peloponnese

INTRODUCTION

Real-world information, knowledge and procedures after which information systems are modelled are generally of dynamic nature and subject to changes, due to the emergence of new requirements or revisions to the initial specification. E-government information systems (eGIS), in particular, present a higher degree of volatility in their environment, since requirement changes may stem from a number of sources, including legislation changes, organisational reforms, end-user needs, technological developments, interoperability and distribution concerns and so on (Jansen, 2004; Prisma Project 2002; Scholl et al, 2005). To this end, the design and implementation of eGIS must adhere to paradigms and practices that facilitate the accommodation of changes to the eGIS as they occur in the real world. Object-oriented technologies have been extensively used in this context to encapsulate reusable, tailorable software architecture as a collection of collaborating, extensible object classes, however the inherent conflict between software reuse and tailorability has inhibited the development of frameworks and models that would effectively support all requirements exposed by eGIS (Demeyer et al. 1997). The lack of such frameworks has lead to situations where eGIS cannot easily be adapted to the new requirements, mainly because only the pre-determined specifications are taken into account and design decisions are fixed during the implementation phase (Stamoulis et al. 2003).

A key issue to a viable solution eGIS modelling is the provision of the ability to multiple public authorities (PAs) to represent different aspects of the same real-world entity, while maintaining at the same time the consistency of the information. Aspect representation is not only limited to data elements that describe the particular entity, but may extend to alterations of its behaviour, when the entity is examined in different contexts. For example, an entity representing the citizen is expected to assume the behaviour of *beneficiary*, when used in the context of the Ministry of Social Security, and the behaviour of *taxpayer*, when accessed from the Ministry of Finance’s eGIS. Distinct behaviours may rely on different data representations and/or respond differently in requests. In this work we present a role-based modelling and implementation framework, which can be used for building eGIS and we argue that this model promotes the tailorability and maintainability of eGIS, which are important aspects due the eGIS inherent complexity and their continuously changing requirements and specifications.

BACKGROUND

Currently, the representation of different real-world entity aspects is mainly achieved through the use of multiple, totally independent representations of the real-world entities, one for each PA eGIS. Each representation encompasses the data elements, and models the behaviour pertinent to the specific organisation. Note that these data and behaviour may include portions administratively regulated by other PAs: for instance the Ministry of Transport is administratively responsible for defining vehicle ownership-related data, however the Ministry of Finance eGIS should include such data, for taxation purposes.

At the other extreme of each eGIS storing its own copy of real-world entity representations, the approach of a single, centralised repository can be undertaken. According to this approach, some PA develops and maintains an eGIS, which is the authoritative source for both defining the schema and storing all the data values for real-world entities. The schema must consolidate all data-related requirements by all PAs, while security rules are introduced to limit the access of any PA to the schema elements pertinent to its task(s).

Between the fully replicated and the fully centralised approach, a federated database approach (Chorafas, 1993) could be adopted, according to which each PA eGIS defines some portion of the schema which is *exported*. A PA eGIS may also *import* schema portions that have been exported by other "federation members", with each import augmenting its locally defined schema accordingly. The federated approach decentralises the schema maintenance tasks, assigning to each PA the task of updating the global schema portion for which it is administratively responsible.

A methodology towards meeting both the requirements of multi-aspect modelling and context-specific behaviour, while enhancing the overall system maintainability is the adoption of *two base constructs* for entity modelling: the first construct models fundamental behavioural blocks of a system, providing only the essential behavioural elements of the most abstract version of the modelled entity; this construct is termed *ATOMA* (Theotokis 1997) and realises the most basic collaboration and reuse contracts (Codenie, W. et al., 1997). Enhanced and context-specific behaviours are modelled using a second construct, namely *roles*, which are attached to atoma for modelling functional behaviour related to a basic entity. Role attachment and removal can be performed dynamically, and multiple roles can be attached to a single entity, effectively modelling *facets* of this entity. Roles also implement their own collaboration and reuse contracts, through which operations are requested in the context of eGIS.

The Atoma framework is based on the concept of *separation of concerns* (Theotokis 2003), which is a key concept in realising deferred design decisions as it facilitates the notion of "injectable" behavioural adjustments in existing operational eGIS. The ATOMA model allows object-oriented design and code to be decomposed into units, describing basic behaviour, as this is captured during the initial design phase from the contractual requirements, and units that specify either variations or changes to these requirements, or new requirements, as these emerge in time. Both at design and implementation level, the former are represented as standard object-oriented classes, while the later are roles that, when composed with classes, realize the ever-emerging requirements. Each role can therefore be refined separately to a code artefact, and the details of the code composition can be derived from those of the design composition. There is excellent traceability and monitoring at this level, because code and design units correspond directly, and so do requirements and design. Furthermore, within a single composition, standard object-oriented design or code is used. This traceability and monitoring facilitates both evolution and "round-tripping": projecting design changes into the code and requirements or, for that matter, reflecting the changes made in the code back into the design and requirements.

THE ROLE-BASED MODEL FOR TAILORABLE E-GOVERNMENT SYSTEMS

eGIS Modeling using the Atoma Framework

According to the atoma framework, eGIS model the real-world entities using two fundamental constructs, namely *atoma* and *roles*. For each real-world entity, a single atoma construct is defined, which encapsulates all the fundamental data and functionality needed for managing these entities. The PA eGIS within which the construct will be defined may be chosen on the basis of various criteria, such as administrative responsibility (e.g. the Ministry of Transport defines constructs for vehicles), technical know-how (some PAs may have more experienced IT staff), expected access patterns (define the construct in the eGIS it is bound to be accessed from most frequently) and so forth. Once the construct for modelling a real-world entity has been defined, it may be *exported* for use by other eGIS, which will *import* this construct. Each eGIS will also provide implementations for the roles that will be assigned to the atoma that will be used within its context, either locally defined or imported. For example, the Ministry of Transport may define the roles *taxi* and *bus* that will be assigned to atoma of type *vehicle* (locally defined), while the Ministry of Finance may define the role *taxpayer,* which will be assigned to atoma of type *person*, a type that is imported from the Ministry of Social Security. Note that the correspondence between roles and atoma types is not necessarily one to one: the Ministry of Finance may define the *taxable good* role, which can be assigned to atoma of various types, such as *building*, *car*, *value-added service* and so on. The communication between a role and the atoma construct to which it has been assigned is based on the collaboration and reuse contracts provided by the atoma construct; thus, the *taxable good* role can be assigned to any atoma construct that implements the collaboration contract *price*, which will be invoked to obtain the net price (before taxation) of the relevant real-world entity. During the period that an entity is assigned the *taxable good* role, its behaviour is enhanced with the operations defined in the role (e.g. *tax amount*), while some of its operations may be overridden by respective ones implemented by the role (e.g. the *price* behaviour will be redefined to return the net price plus the tax due). Note that both the role and the underlying entity may independently evolve without any disruption in their communication, as long as the collaboration contracts are respected. Moreover, the interaction between a role and the eGIS it is used within will function properly, as long as the existing elements of the collaboration contract implemented by the role are not altered (new elements may be added with no side-effects).

Similarly to atoma constructs, roles can also be *exported* by the PA eGIS that has defined them and imported by other PA eGIS for use, if the functionality they provide is useful in the context of the importing eGIS. For instance, the *taxable good* role, defined by the Ministry of Finance, may be imported by the Ministry of Commerce and assigned to atoma constructs, either locally defined (e.g. *value-added service*) or imported (e.g. *car*).

Maintaining role-based eGIS

Regarding system maintenance, when a new requirement is added or an existing one is changed, a new design aspect (a role) is created to address it. The new design aspect can then be composed with the existing design. Afterwards, the design aspect can be refined to a code aspect (role implementation), which is similarly composed with the existing code. The changes are localized without risking the rest of the system's stability, so there is no tangling, and both traceability and monitoring are preserved. In addition to this, dealing with the alignment problem, the decomposition into roles alleviates the monolithic nature of the design, and allows for concurrent development, while the composition underlying the ATOMA model provides a powerful mechanism for integration, evolution, customization, adaptation, tailorability and improved reuse.

It is in fact this composition mechanism that provides a remedy to the problem of constructing rigid systems. The necessary inflexible mapping of requirements to design units in no longer required, and design decisions, which fix requirements into design, and subsequently code, are no longer necessary. By separating the various cross-cutting aspects of a system, modelling them independently and composing them back to deliver the required function of the eGIS, we advocate that deferred design decisions can be realized, and thus enable the construction of truly tailorable eGIS

Using atoma and roles: a case study

Figure 1 presents an example of an eGIS, with three constituent PA eGIS. The case study has been drawn from the Greek public administration, by studying the related legislation and practises. Atoma are represented as rounded rectangles, whereas parallelograms are used for roles. Solid lines indicate that the construct has been defined within the eGIS it is depicted in, while dashed lines indicate imported constructs. Role assignment to atoma is shown through arrows. Only one atoma construct is defined in this example, namely Person. This construct is defined within the Ministry of Social Security, which additionally defines the *beneficiary* role that it assigns to the person construct. In this manner, a *Person* exhibits the behaviour of *beneficiary* only when it is treated within the context of the Ministry of Social Security, whereas outside the scope of this eGIS only the basic *Person* behaviour is available. The two other ministries (Health and Finance) import the Person construct and assign to it locally defined roles (Patient and Taxpayer, respectively). The Ministry of Finance additionally imports the Patient role from the Ministry of health, and assigns it to the Person construct, since the expenses for hospitalisation included in the Patient role's contract are needed within the context of the Ministry of Finance's eGIS, to enable their deduction from the person's taxable income.

Since roles are independent of one another, it is possible that two or more roles implement different behaviours under the same collaboration contract. For instance in Figure 1, it is possible that the Patient role includes the *Checkup* contract, which triggers the performance of clinical tests to the patient, and the role *TaxPayer* includes too a *Checkup* contract, which initiates a detailed audit of the person's tax records. Since within the Ministry of Finance's eGIS both roles have been assigned to the Person construct, both contracts are valid for use and an ambiguity issue is raised, regarding which behaviour should be selected. This ambiguity is resolved by defining priorities for role assignments, effectively dictating the order in which role contracts will be scanned for elements matching any individual contract use. In the aforementioned example, the Taxpayer role will have a greater priority than the Patient role, thus the behaviour most pertinent to the business model of the Ministry of Finance will be selected. It is generally expected that locally defined roles should be assigned higher priorities than imported roles, however no such restriction is imposed by the model. The atoma framework encompasses even more powerful methods for atoma and role combinations, such as molecules, which can be used in more complex cases (Theotokis, 2001).

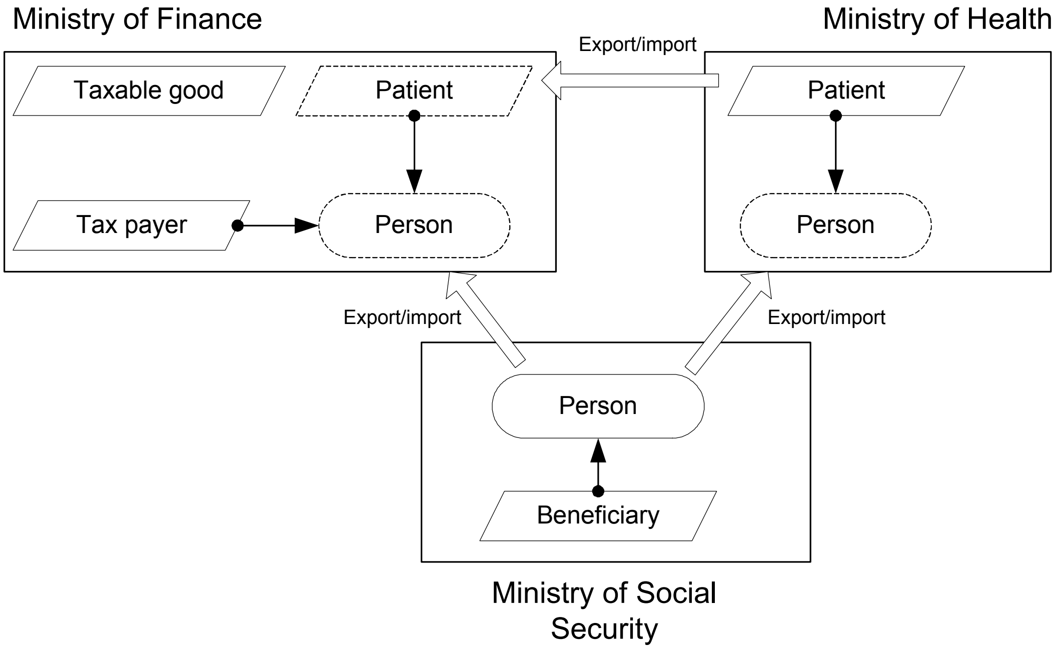

Figure 1 – Definitions, exports and imports of atoma and roles

Comparing the Atoma framework and other approaches

As noted in the background section, the dominant approaches for modelling multiple aspects of real world entities within eGIS belonging to different PAs are (a) the use of totally independent systems (b) the introduction of a centralised repository where all schemata are consolidated and (c) the use of a federated schema approach. These practises, however, have a number of disadvantages seriously impeding the design process, modelling of context-specific behaviour, tailorability and maintainability.

Firstly, the practice of using independent systems introduces multiple autonomous representations, which are a potential source of inconsistencies, both at the level of schema representation and at the level of stored data values (Lenz, R., 1996; Wiesman, M. et al., 2000). Indeed, if a change that affects the schema of the data used for a specific purpose is decided by the administratively responsible PA, this change should be communicated to all other PAs that handle such data, and their eGIS should be accordingly modified; the distributed nature and the scale of these maintenance activities increases the associated costs and the probability of errors. At data instance level, changes to the data values stored in one eGIS do not affect the corresponding values in other eGIS, leading to discrepancies in the representation of the same real-world entity. Finally, multiple copies of the same information are maintained, increasing the overall storage requirements.

The centralised repository approach, eliminates on the one hand inconsistency problems (only one schema is defined and a single copy of each datum is stored), on the other hand, however, it hinders the modelling of context-specific behaviour for the entities: if entity behaviour is modelled within the global repository (encapsulated with the data items), a single behaviour would be provided for all eGIS using the entity, regardless of the context. If entity behaviour were provided separately by each PA eGIS, changes to the "global schema" would necessitate maintenance activities for all affected PA eGIS. Note also that global schema amendments and the PA eGIS maintenance should be performed "almost synchronously", since it is imperative that the PA eGIS data model should exactly match the repository data model upon any access.

Finally, the federated database approach it does not address the issue of context-specific behaviour: if each PA eGIS should fully model behaviour entities should have in its context, changes in some schema portion would again call for maintenance activities to all PA eGIS importing the modified schema; if the behaviour were provided by schema publishers, context-specificity would be hindered and an additional concern would be raised, regarding where to code behaviour that relies on multiple schema portions, exported by different PA eGIS.

The role-based model based on the Atoma framework, successfully tackles all these issues, by allowing organisations to model both the data items and the behaviour pertinent to the entity within the role concept, which is attached to the basic atoma construct corresponding to the real-world entity. Roles can be shared between eGIS, at both schema and instance-data level, removing thus any inconsistencies due to multiple copies. Context-specific behaviour can be directly modelled through new context-specific roles and/or by fine-tuning the assignment of roles to atoma. Finally, maintenance is facilitated, since changes in existing requirements or emergence of new ones can be addressed through the modelling of new roles. Note that the creation and assignment of new roles (as well as the de-assignment of existing ones) can be performed at any time instant, removing the need for fixing design decisions at early stages, and increasing thus the overall system flexibility.

FUTURE TRENDS

eGIS have emerged in the past few years, aiming to exploit information and communication technologies for the provision of public service with improved quality and reduced costs (Forman, 2002). In the area of services, most of the work has been devoted to the development of isolated informational and transactional services, but service interoperation and integration, which provides added value for the service consumers [e.g. for handling *life-events* (Wimmer, 2002)], has not received comparable attention. It is expected that the research and implementation agenda for eGIS will include these aspects, for which the work presented herein provides the necessary technical infrastructure. Of equal importance is the provision of a methodological framework that would support the phases of eGIS requirements analysis and design, enabling the involved stakeholders to unambiguously distinguish between basic and context-specific functionalities of real-world entities, providing appropriate input to the implementation phase. Methodologies extending beyond traditional service-provision systems, to include the dimensions of managerial effectiveness improvement and democracy promotion (Gil-Garcia R., 2004; Cohen, S., 2002), providing a holistic framework for e-government, will also be of essence.

CONCLUSION

In this article we have presented a framework for constructing tailorable eGIS. The proposed framework employs two basic constructs for modelling real-world entities, namely *atoma* for encapsulating the basic representational needs and the fundamental functionality, and *roles* for representing context-specific behaviour, which is attached to atoma on demand, according to the needs of the eGIS within which the entity is examined. By employing this design and implementation frameworks organisations enhance the overall system maintainability, since atoma and roles may evolve independently, provided that the reuse and collaboration contracts are respected along the evolution. New behaviours, necessitated by the change of existing requirements or the emergence of new ones can also be seamlessly integrated into the system, by modelling new roles and attaching them to the pertinent atoma. Future work will address the formulation of a methodology for eGIS design according to the atoma framework. Migration and mobility of atoms and roles, as well as versioning and persistency issues will be also examined.

REFERENCES

Chorafas D. N., Steinmann, H. (1993). Solutions for Networked Databases: How to Move from Heterogeneous Structures to Federated Concepts, Academic Press

Codenie, W., De Hondt, K., Steyaert, P., Vercammen, A. (1997). From Custom Applications to Domain-Specific Frameworks. Communications Of The ACM Vol. 40, No. 10, October 1997

Cohen, S. (2002). The Future Of E-Government: A Projection Of Potential Trends And Issues. Price Waterhouse Coopers. http://www.columbia.edu/~sc32/future%20of%20egov.pdf

Demeyer, S., Meijler, T. D., Nierstrasz, O., Steyaert, P. (1997). Design Guidelines for 'Tailorable' Frameworks. Communications Of The ACM Vol. 40, No. 10, October 1997

Forman, M. (2002). E-Government Strategy - Implementing the President's Management Agenda for E-Government. http://www.whitehouse.gov/omb/inforeg/egovstrategy.pdf

Gil-Garcia, R. (2004). Electronic Government. ISWorld Encyclopedia. http://ispedia.terry.uga.edu/?title=Electronic_Government

Jansen Bas (2004). Capturing legislation for eGovernance. White paper, http://www.egovernance-consortium.org/WhitePapers/Janssen2004a.pdf

Lenz, R. (1996). Adaptive distributed data management with weak consistent replicated data. Proceedings of the 1996 ACM symposium on Applied Computing table of contents, Philadelphia, Pennsylvania, United States, pp. 178 - 185

Prisma Project (2002). Prisma Project Deliverable Deliverable D4.1 - Report on pan-European scenario-building. http://www.prisma-eu.net/deliverables/D4-1.PDF

Scholl H. J., Klischewski, R., Moon M. J. (2005). Minitrack: "E-Government Infrastructure and Interoperability". Proceedings of the 38th Hawaii International Conference on System Sciences

Stamoulis, D., Theotokis, D., Martakos, D., Gyftodimos, G., (2003). Ateleological Development of "Design-Decisions-Independent" Information Systems. In Adaptive Evolutionary Information Systems, Nandish Patel (Editor), Idea Publishing Group. ISBN 1-59140-034-1. pp 81-104.

Theotokis, D. (2001). Object-oriented development of Dynamically modifiable Information Systems using Components and Roles, PhD Thesis, Dept. of Informatics and Telecommunications, Univ. of Athens (in Greek)

Theotokis, D., et al. (1997). ATOMA: A Component Object-Oriented Framework for Computer Based Learning. Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLIS '97), De Montford University, Leicester, UK, pp. G5-G15

Wimmer, M., Tambouris, E. (2002). Online One-Stop Government: A working framework and requirements. In Proceedings of the IFIP World Computer Congress, Montreal

Wiesman, M. et al. (2000). Database Replication Techniques: A Three Parameter Classification. 19th IEEE Symposium on Reliable Distributed Systems Nürnberg, Germany

TERMS AND DEFINITIONS

Atoma: the most abstract level of representation of a real-world entity, encompassing only the most basic data and functionality for representing and manipulating the entity

Role: a situation- or context-specific aspect of a real-world entity. A role may be dynamically associated and removed from an atoma construct when either the real-world entity evolves accordingly (e.g. a *person* becomes an employee or looses this property) or when the behaviour modelled by the role becomes/ceases to be pertinent to the current context.

Export/import: Information systems may define atoma and roles and then *export* them to be used by other information systems. An information system may *import* any of the exported atoma and roles and use it in its own context.

Tailorability: the dynamic accommodation of context-dependent behavioural variations in an existing software system

Representational diversity: the practice according to which real-world entities are represented through totally independent and unsynchronised models in different information systems, leading to inconsistencies and maintenance problems.

Centralised representation repository: A single database in which all real-world entity representation models are stored, and any information system willing to access a model retrieves if from there. This approach keeps the models consistent but shifts the maintenance issues to the client information systems.

Federated representational model: An approach for building representational models according to which each information system defines some aspects of an entity model, contributing thus to a “global” entity model.