A Framework for Managing the Lifecycle of Transactional e-Government Services

C. Vassilakis[1], G. Laskaridis[1], G. Lepouras[1], S. Rouvas[1], P. Georgiadis[1]
[1]e-Gov Lab, Department of Informatics, University of Athens, 15784, Greece
{costas, glask, gl, rouvas, p.georgiadis}@e-gov.gr

Abstract: Filling and submission of electronic forms is a key issue for e-government, since most electronic services offered in this context include some variant of electronic forms. Insofar, IT experts are placed in the centre of electronic forms services lifecycles, undertaking the analysis, design, implementation and maintenance phases. This practice, however, implies various impediments, such as the need for multitudinous teams with diverse skills. In this paper we present experiences from developing and maintaining a set of electronic services for the Greek Ministry of Finance, and propose an approach to handling electronic services' lifecycle that balances responsibilities between domain experts and IT professionals. This approach enables a more holistic management of the electronic service lifecycle, by employing modelling and representation in high levels of abstraction and incorporating tools for automatically generating operative service instances from these high-level descriptions.

Keywords: e-government, electronic service, lifecycle framework

1 Introduction

Quoting from the European Commission's Green paper on Public Sector Information in the Information Society "The emerging Information Society, largely driven by an ever increasing and pervasive use of information and communication technologies is more and more affecting the public sector. Administrations follow the example of the private sector and benefit from the enormous potential of these technologies to improve their efficiency. This development is often labelled "Electronic Government" and covers both the internal and external application of Information and Communication Technologies (ICT) in the public sector" [European Commission, 1999].

An appropriate set of indicators may be used to quantify the development of "Electronic Government" and to measure comparative progress. *eEurope* has published a list of 20 basic public services [eEurope, 2000], which should be considered as the *first steps* towards "Electronic Government", along with a methodology for assessing the status of government online services [eEurope, 2001]. It is worth noting that among the basic public services listed in [eEurope, 2000], 15 of them (75%) are *transaction services*, i.e. services that involve filling-in and submission of electronic forms. The citizen interface (front-end) of the transaction services must be connected to the organisation's back-office to complete the processing cycle and offer the rich spectrum of services that customers want and governments have promised [Jupp, 2001].

Insofar, however, the development of transactional services lags behind the expected level, although the technological potential offered is adequate. This is owing to the following reasons:

- From the organisational point of view structural reforms, and the adoption of a customer-centric model are required in order to enable the delivery of high-quality services to the citizens [Robben, 2001]
- From the IT perspective, the development and maintenance processes for such services are quite complex: firstly, service requirements must be analysed, involving IT staff and domain experts. Secondly, the service has to be designed, taking into consideration the functional requirements that emerged from the previous stage, user interface aspects, as well as administrative issues that pertain to service operation. Subsequently, the service has to be implemented and deployed; for these two tasks, it will be necessary to link the e-service platform to installed IT systems, in order to save and retrieve data. Finally, when changes to the service are required, the whole process must be carried out from the beginning, resulting in significant costs and time delays.

In this paper we present experiences from developing, operating and maintaining the electronic tax return service of the Greek Ministry of Finance, and propose a new approach, complete with the associated tools, which enables public sector employees, having the necessary domain knowledge and only basic IT skills, to develop and maintain transactional services. The proposed approach and tools have been developed by the paper authors and employed for developing new services for the Greek Ministry of Finance, and have proven insofar to provide a far more intuitive and maintainable framework for electronic services.

The rest of this paper is organised as follows: section 2 outlines the approach taken to implement the electronic tax return service for the fiscal year 2001 and the problems faced; section 3 provides an overview of the state-of-the-art, regarding the technological areas involved in electronic service development and deployment. Sections 4 and 5 present the overall architecture and tools for new approach to handling the electronic services lifecycle. Section 6 outlines the advantages of the proposed approach and finally, section 7 concludes and draws future research directions.

2 Experiences from an electronic service’s life cycle

The development of the electronic tax return service for year 2001 followed the waterfall software engineering paradigm [Sommerville, 2000], starting of with the user requirements analysis. Four major requirement categories were identified in this procedure and more specifically:

1. Provision of the appropriate forms through which users would fill in their tax return.
2. Implementation of *input validation checks*, which would verify that user input conforms to the tax return submission rules, as specified by the current taxation legislation.
3. Forwarding of collected data to the back-end system for the tax computation process.
4. Collection of the results of the tax computation process and notification of each user about the result of the processing of his/her tax return.

These four requirement dimensions are in fact the “computerised” counterparts of the paper-based tax return submission and tax computation process. The implementation team, however, had to face some additional issues, which were due to the fact that tax declarations were submitted electronically:

1. Prospective service users should be provided with *identification credentials* (usernames and passwords), through which they could authenticate themselves to the service and submit their tax return.
2. Users should be able to perform a limited administration of their personal data and update some information, such as passwords and e-mail addresses.
3. The Ministry's administration team should be equipped with sufficient management tools for user account management, database backup and recovery, batch generation of personalised e-mail messages, statistics reports etc.

The overall service operation scheme is illustrated in Figure 1.

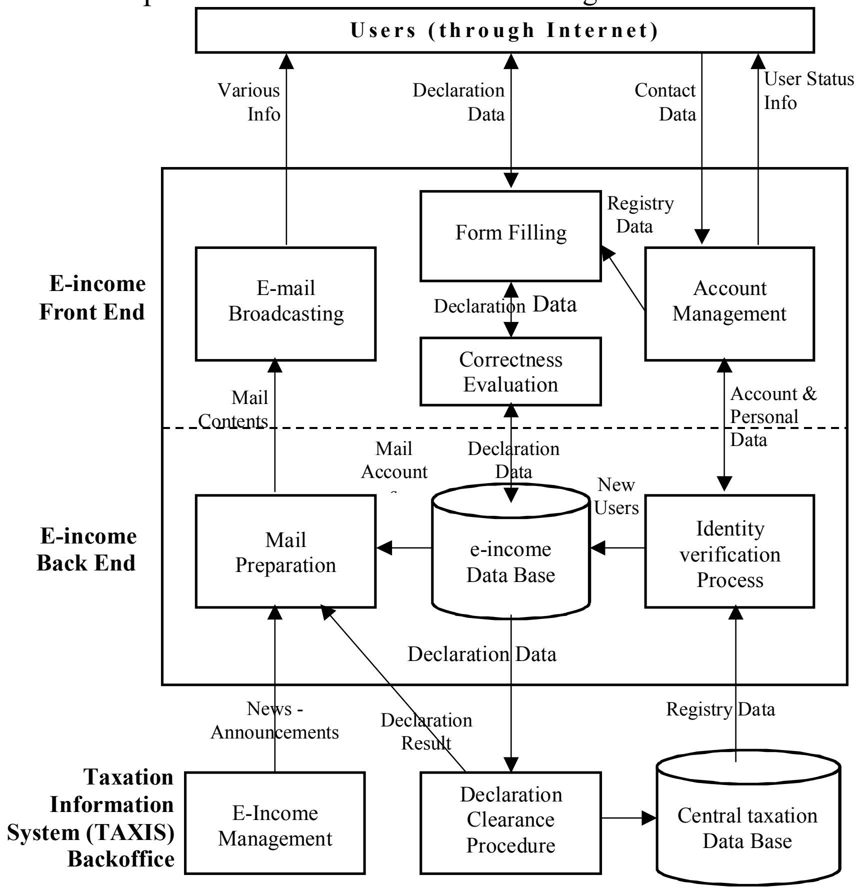


Figure 1 - Operation scheme for the electronic tax return submission service

After the user requirements phase was completed, the design phase was initiated, in which detailed specifications of the HTML forms, the input validation programs and the data interchange procedures with the back-end system were produced. Moreover, in this phase the interfaces between the different stages of the information flow were specified. Subsequently, each portion of the work was implemented "autonomously", and an integration step consolidated the different modules into a single, operational platform. After the service became operational, some maintenance tasks were performed, mainly for the purpose of correcting appearance problems and modifying or enhancing input validation checks.

During the life cycle of the project, a number of shortcomings of the followed approach were identified, which led to increased product delivery times and the need to involve more staff, usually with diverse field expertise. These shortcomings are briefly discussed in the following paragraphs.

2.1 Implicit knowledge

One of the major problems faced during the life cycle of the project was that knowledge existed *implicitly* within the organisation, usually under the possession of experienced individuals, rather than stored in some publicly accessible repository in an explicit form. This mainly affected the subtask dealing with the user input validation, since this portion is closely coupled with the organisation's *business logic*. The analysis team that interviewed domain experts, in order to extract the user requirements, often collected partial, or even contradicting descriptions of the rules that applied to different cases. In some cases, this was due to the fact that even experienced personnel may not take into account certain regulations that apply only to a small number of citizens; in other cases, some individuals were not even aware that some regulations applied, or that some piece of legislation was revised.

Although it may be argued that this issue should be solved at *organisational level*, it is worth noting that the user requirement analysis process that took place within this project amassed an amount of knowledge, which was recorded as project-specific, low-level user requirements, rather than as high-level knowledge. In this sense, the collected knowledge was non-reusable and difficult to maintain.

2.2 User interaction

A second issue that was not satisfactorily addressed by the "traditional" approach was that of the user interaction. Although a team of experts undertook the coding of the HTML forms, certain aspects could not be addressed using technical expertise alone, but required substantial domain expertise. For instance, the range of values that are usually entered in some field is a significant aid to HTML designers, in their effort to keep the forms compact. Moreover, fields that are semantically related should be placed close together on the form, in order to ease the form filling process. In both of these cases, domain expertise is required, however the collaboration of HTML experts and domain experts in this level of detailed has proven to be quite tedious.

User interaction with the service was hindered by the fact that users did not have access to expert assistance, while filling in their tax return: contrary to the submission to the tax office, where the citizen could obtain expert and specific for their cases information from the tax officers, users submitting through the electronic version of the service were limited to accessing generic help documents. In some cases, this has proven to inhibit citizens from using the service, since they preferred to submit their tax return to the tax office, where the required help was available. User interaction issues, as recorded by a user poll, are summarised in [General Secretariat for Information Systems, 2001a] and measures are proposed in [General Secretariat for Information Systems, 2001b].

It must be pointed out that in most cases, the "expert assistance" required by the users, was exactly the knowledge accumulated during the user requirements analysis phase. This knowledge, however, was mapped in the design and implementation phases to low-level system specification and programming primitives, and remained inaccessible to the users.

Another potential problem for user interaction is that in many cases, electronic forms are modelled after the paper forms; this is not always the best option, since paper forms are often lengthy and field completion in some location may require looking up the value filled in some field elsewhere. Although this is not a major problem for paper-based systems, electronic forms should remain compact –preferably with no scrolling at all– and the need for navigation between pages for field value lookup should be minimised.

2.3 Code reusability

By breaking down the code written within the project into categories, it was found that 40% of the code implemented the organisation's business logic, 30% dealt with administration procedures, 15% catered for data interchange with the back-end system and 15% addressed various issues related to the operation of the service.

When designing enhanced versions of the service for the fiscal year 2002, it was an unpleasant surprise to realise that although more than 60% of the existing code was relevant to the designed extensions, it required considerable revisions –and in some cases recoding from scratch. This was owing to the fact that the code was written having in mind the project-specific aspects of the organisation's and the service's requirements, rather than a more holistic view of the organisational knowledge and electronic service issues. Moreover, due to some execution platform changes, code that would otherwise remain intact had to be translated to fit into the new platforms. This was found out to be a tedious and error-prone process.

2.4 Communication with back-end systems

In order to complete a full transaction processing cycle, most transactional electronic services need to exchange data with some information system installed within the organisation. The organisation's information system may be either directly accessible, or physically isolated from the machines delivering the electronic service, due to security reasons. In both cases, building custom interfaces, tailored for each case, has proven not to be a good practice, since it requires substantial programming effort to cope with the idiosyncrasies of the various modules of the back-end system.

2.5 Paper workflows instead of data workflows

The adopted approach for the design and implementation of the online tax declaration service was the provision of electronic counterparts for the existing procedures. The main problem that arises from this approach is that existing procedures are *paper oriented*, i.e. they were designed having in mind paper distribution, paper attachments, paper receipts, physical signatures, imprints etc. The procedures used for paper-based services, however, are not necessarily the ones best suited to electronic information flows. Thus, some *business process reengineering* activities are required to upgrade the overall workflow schema of the specific administration. Although theory suggests that business processes must be re-engineered before design and implementation commence, strict service delivery deadlines together with bureaucratic delays often enforce a reverse sequence in time. To this end, the delivered platform must be flexible enough to be able to adapt in a changing business environment with a minimum of changes.

2.6 Security issues

One important aspect that should not be overlooked in this stage is the provision of appropriate *security measures* and the clear definition of *roles*. Indeed, security in paper-oriented systems is usually implemented through lockers and safes; however such measures are not applicable to securing electronically stored information. Roles must also be defined and the authorities granted to each role must be stated explicitly.

The electronic service development and delivery platform must include appropriate concepts and constructs to allow for the definition and enforcement of security rules.

3 State of the art

With respect to the issues presented in the previous sections, a number of technological solutions are available nowadays, which are usually employed in the electronic service lifecycles. Recording knowledge in an explicit and reusable format is a topic traditionally addressed by knowledge management tools. A plethora of such tools is commercially available, with some of them specifically addressing the Public Sector (e.g. [Cognos, 2002; Computer Sciences Corporation, 2002; Information Management Research, 2002; TheBrain Technologies Corporation 2002]) while a number of active research projects are introducing novel methods to knowledge management (e.g. [DECOR Project 2002; Know-Net Consortium, 2002]). It is worth noting, however, that despite knowledge is a valuable asset for public administrations, the number of organisations that have set some policy for recording it explicitly still remains low. Moreover, development of electronic services is usually treated as an isolated software project, thus information extracted from the involved domain experts is recorded as low level "user requirements", rather than as high-level organisational knowledge.

For the service development and deploying phases, numerous solutions are available, including some commercial products. XMLForms™ by Schemantics [Schemantics Inc. 2001], Accelio Capture™, Accelio Integrate™ and Accelio Present™ by Accelio Worldwide [Accelio Worldwide, 2001] and Oracle E-Business Suite™ by Oracle Corporation [Oracle Corporation, 2001] are examples of such platforms; the Open Source world has also provided a number of software packages as well.

Recently, the W3 consortium has published the *XForms* specification [W3 Consortium, 2002]. *XForms* is an XML-based standard for specifying Web forms. Each form has three main components, namely "form purpose", "form presentation", and "form data", thus the content, structure and user input are clearly separated.

Finally, regarding service deployment to the end-users, a number of user interface related issues have to be taken into account. Employing a user-centric approach (ISO 13407 [International Standards Organisation, 1999]) will provide the users with a more intuitive, comprehensible and friendly interface, while suggestions published by the Web Accessibility Initiative (WAI [Web Accessibility Initiative, 1999]) must also be considered, in order to make services available to everybody.

4 Proposed approach

Although the state-of-the-art provides sufficient tools for tackling the various phases of electronic service lifecycle, these phases are still handled in isolation, rather than as an integrated process. Besides requiring numerous implementation teams with diverse skills, this approach introduces an *impedance mismatch* situation, where the same information is represented using different concepts, terminology and structures in various phases, leading to increased development times, difficulty in communication between teams and reduced maintainability.

To provide an alternative solution for the problems described above, one has to adopt a layered approach that would introduce higher levels of abstraction, enhancing thus maintainability and re-usability, isolate knowledge from the code, allow the asynchronous development of different modules. The approach also places domain experts in the centre of the development cycle, rather than treating them as mere input providers for the user requirements stage. Domain experts possess the knowledge about the data that need to be collected by the electronic service and the processing that must be applied on them.

Under this approach, an electronic service consist of the following components:

1. *The data*, which corresponds to the input boxes of paper-based forms, which the citizens are asked to fill in. In the proposed environment, however, each piece of data may be supplemented with additional information, such as type information (text, numeric etc), text labels, instructions, examples, references to supporting legislation, presentation options (input field length, number format), qualitative characterisations (e.g. the surname field may be characterised as "personal data", whereas the acreage of a house may be characterised as "estate property"). Data may be either provided by the user or retrieved from administrative registries. Multiple individual data items may be *grouped* for easier reference and management.
2. *The input form(s)*, through which users will be prompted to enter the relevant data. In practice, an input form is a set of references to data items, together with placement information and navigation links, and may be statically designed or dynamically created, based on the characterisations of the data. In this manner, one service may be presented differently to various users, to cater for personal preferences and convenience, as well as functional needs. Interface flexibility is important for value added professionals that will be using the service on behalf of their clients, since this user group would opt for more "light", batch-oriented submission methods, *au contraire* to individual users who would prefer a rich, menu-driven environment with visual enhancements and detailed help.
3. *The validation checks* that user input must pass for a submission to be considered successful. Validation checks may range from simple tests, such as data type and format checking, value ranges and input for mandatory fields, to complex assertions involving multiple fields (e.g. *if user selects field 170 and 180 and has less than two children, then field 190 must be completed according to the law 2076/1998*), or even multiple forms (e.g. *Form2 cannot be submitted until Form1 is submitted* or *the value of Field1 in Form1 must be less than the value of Field2 in Form2*).
4. *The post-processing* that must be applied on the collected data so as to complete the transaction and/or forward the submitted document to the next step of the organisation's workflow chain. This step may include storage of data to the organisation's databases, invocation of external programs, production of new documents, sending of alerts to the organisation staff that must examine the submitted data, etc.

For a service to be developed and deployed, the organisation's domain experts, managers and IT staff should contribute to the definition of the four dimensions identified above. More specifically:

1. Managers determine the services that must be rolled out through the electronic platform.
2. Domain experts define the data that will be used in the context of each service. Usually these data should be the same to the data appearing on the paper-based versions of the service. Domain experts supplement definitions with appropriate information, as described above, and provide the validation checks that should be applied on user input. Since definitions are stored in the central knowledge repository, they may be reused in the context of other services; for

instance, if an organisation defines a data item group named "Personal data" in the context of service A, the data item may be directly reused in the context of service B. Additionally, domain experts create the statically defined input forms and outline the post-processing procedure for the successfully submitted forms.

3. IT staff complement the work of domain experts by providing database connectivity for retrieving data from administrative registries and storing the submitted data to the organisation's database. Moreover, IT staff may assist in the implementation of validation checks that prove too complex for domain experts to handle, optimise the interfaces created by domain experts and implement the external programs that may be needed to run as part of the post-processing procedure.

It is important to note that throughout the definition phase *semantically rich elements* are used, e.g. "the total husband's income from salaries", rather than implementation-oriented terminology, such as *table4_field22*. This enhances readability and allows for easier maintenance, since the level of abstraction remains high and semantic information is retained in this representation. We also note that security management is a horizontal activity spanning across all user groups. For each piece of information within the knowledge base, its creators and administrators may designate the access rights each user has on it and whether it will be accessible by the end users of the service (i.e. the citizens). Security information is also applied to elements of the post-processing stage, such as database tables and columns, new documents produced etc.
The process described above is illustrated in Figure 2.

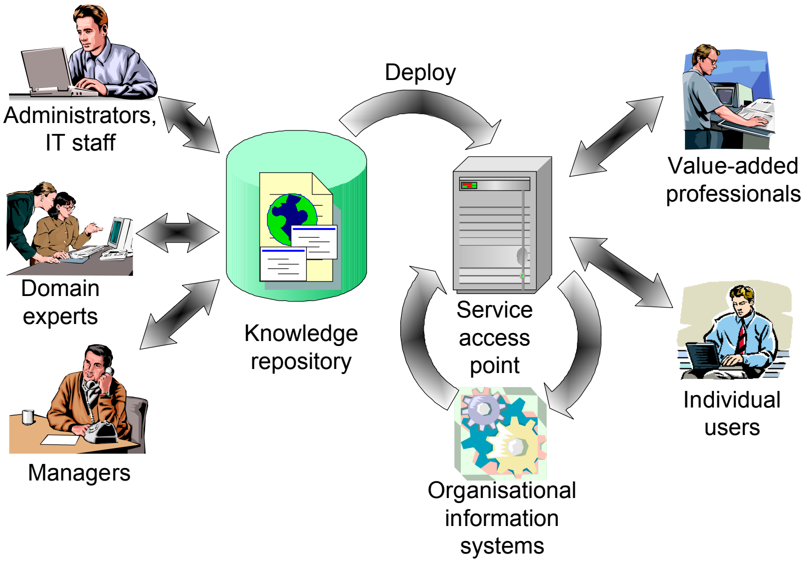


Figure 2 - System Architecture

Each user group, besides providing its expertise for the implementation of the service, may define information that should be collected during service operation, but is not *per se* part of the service. For example, managers may require the collection of statistics regarding the user groups accessing the service or other data that will guide decision making procedures; domain experts may benefit from the collection of data for submissions that have failed due to validation errors, in order to provide better-suited examples and more extensive help; IT staff would like to collect performance data, so as to optimise and fine-tune the platform.

When all the appropriate information for the service has been specified, the service may be *instantiated*, through the automatic generation of:

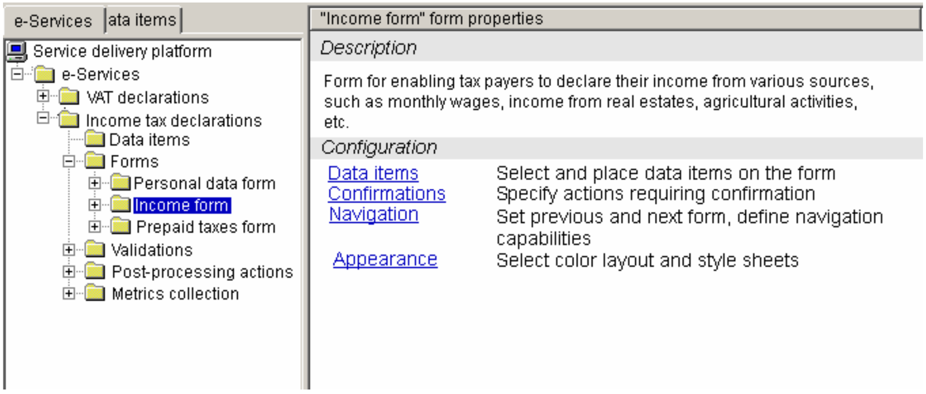
1. *User interfaces (forms)*. Statically designed forms are directly instantiated, and additionally, data items are grouped based on their qualitative characterisations, creating thus “wizards” through which users will be guided throughout the process of data input. For example, a “wizard” for a tax return would gather all “personal data” item in page 1, income sources in page 2, pre-paid taxes in page 3 and would connect the pages with navigation links. Both statically designed and dynamically created forms include information associated with the data items appearing on the form, either directly (e.g. textual labels describing the field) or in the form of links (e.g. links to examples, legislation etc). Consistency in user interfaces may be achieved through the use of templates, which specify presentation attributes and suggest layouts.
2. *Non-interactive submission interfaces.* This feature aims to assist value added professionals, allowing them to make automated submissions on behalf of their clients through appropriate software. For instance, accountants may submit their clients’ tax return directly through the accounting applications they use to record incomes and expenses, without having to fill in the interactive forms. Non-interactive submissions are based on XML.
3. *Validation programs*, which will ascertain that user input conforms to the checks provided by domain experts. These programs are generated by translating the high-level descriptions provided by the domain experts to executable code.
4. *The databases* that will store the data pertaining to successful submissions.
5. *Metrics collection code* for gathering the statistical information designated by user groups.
6. Optionally, the instantiation process may generate user registration, login, and account management forms to further automate the service deployment procedure.

The instantiated service may be deployed through installation of the generated files on an appropriate service delivery platform, such as a web server.

5 Supporting Tools

In order to perform tasks described in the previous section, each user group must have at its disposal the appropriate tools for accessing the knowledge repository and defining the various service aspects. These tools have been designed to be friendly and intuitive, since a significant portion of its expected users (i.e. domain experts and managers) is not usually (and need not be) too familiar with information technology programming issues concerning the implementation of an electronic service.

The tools complementing the platform provide users with a personalised view of the knowledge base, arranging for a suitable presentation of the aspects of the knowledge base that are of interest to each user, depending on group membership. Information is organised in a hierarchical fashion, and users start off from electronic services located at the top level and drill down to the elements making up each service. Complementary to hierarchical navigation, users may browse through the set of data items already defined in the knowledge base, and facilities for searching and correlating existing information are provided. Searching and correlation allows for identification of “similar” data items, enabling thus domain experts to merge them, before information duplication becomes a source of inconsistency. A screenshot of the tool for exploring the knowledge base (an early prototype version) is depicted in Figure 3.



e-Services | ata items

Service delivery platform
- e-Services
  - VAT declarations
  - Income tax declarations
    - Data items
    - Forms
      - Personal data form
      - Income form
      - Prepaid taxes form
    - Validations
    - Post-processing actions
    - Metrics collection

"Income form" form properties

Description

Form for enabling tax payers to declare their income from various sources, such as monthly wages, income from real estates, agricultural activities, etc.

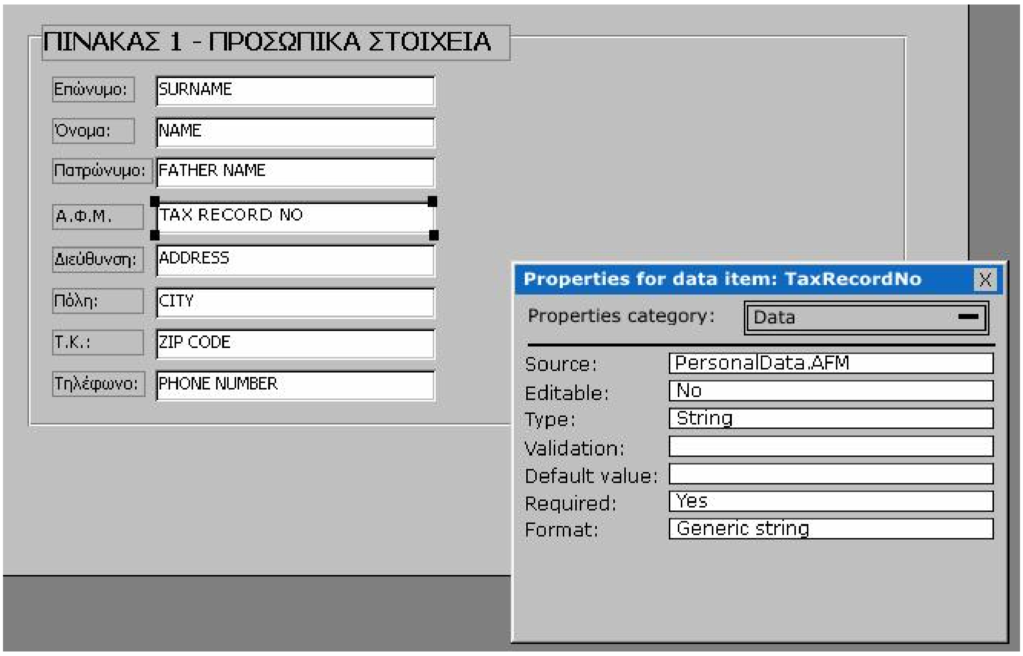
Configuration

Data items	Select and place data items on the form
Confirmations	Specify actions requiring confirmation
Navigation	Set previous and next form, define navigation capabilities
Appearance	Select color layout and style sheets

Figure 3 – Tool for exploring the knowledge base

Once the appropriate piece of information has been identified, the user may query or set its characteristics through an appropriate property sheet. Property sheets are dynamic, presenting each user with the properties related to the groups he/she belongs to. We note here that each piece of information may be accessed through *multiple paths*; for instance, a data item may be accessed either by locating it within the pool of data items, or by browsing to it through the set of data items of a service that the data item participates in, or through a predefined form it has been placed on. Provision of multiple paths facilitates the exploration of the knowledge base and the formulation of a semantically rich network of interlinked service elements.

A screenshot the form definition component is depicted in Figure 4.



ΠΙΝΑΚΑΣ 1 - ΠΡΟΣΩΠΙΚΑ ΣΤΟΙΧΕΙΑ

Επώνυμο:	SURNAME
Όνομα:	NAME
Πατρώνυμο:	FATHER NAME
Α.Φ.Μ.	TAX RECORD NO
Διεύθυνση:	ADDRESS
Πόλη:	CITY
Τ.Κ.:	ZIP CODE
Τηλέφωνο:	PHONE NUMBER

Properties for data item: TaxRecordNo

Properties category: Data

Source:	PersonalData.AFM
Editable:	No
Type:	String
Validation:	
Default value:	
Required:	Yes
Format:	Generic string

Figure 4 – Defining field properties in the development tool

The proposed approach, together with the associated tools, allows domain experts and managers to directly participate in the creation process of electronic services, by accessing and defining relevant information in a user-friendly and intuitive manner.

6 Assessment of the proposed approach

The proposed approach has been assessed, in order to measure its impact both on the various stages of the electronic service's lifecycle and on the quality of the final products (i.e. the electronic services). The assessment took into consideration objective data (such as the total time to develop a service, the overall amount of person hours for a specific task) as well as subjective perspectives (e.g. user-friendliness of the approach, quality of the product etc). Objective measurements were gathered by monitoring the pertinent quantities, whereas for subjective topics user group opinions were collected, using interviews and questionnaires. The most important findings of this assessment process are summarised in the following paragraphs.

1. *Creation of a knowledge repository.* The creation of a knowledge repository is probably the most important gain from this approach, since all tacit and implicit knowledge possessed by domain experts and pertaining to the developed service will be recorded in an explicit, searchable and reusable form. The knowledge repository is extensively used in the design, development and maintenance phases and has been favourably commented in most questionnaires.
2. *Faster development.* The proposed approach enables the direct use of previously developed and tested components of electronic services in new ones. This offers the possibility for implementing and publishing new services with a minimum amount of effort and in shorter time, as compared to the waterfall approach. The latter can be also attributed to the fact that domain experts are directly involved in all stages of the development process, hence the iteration process for producing code is much faster. Depending on the amount of reused components, the gains in service development time may range from 15% to 40%.
3. *Ease of maintenance.* Besides from easing development, the separation of knowledge from the code also facilitates the maintenance process. Maintenance requests take place whenever a change in a regulation, directive or law, forces a change in the corresponding service. With the proposed approach such alterations can be carried out primarily by domain experts who can update the processing rules and, if necessary, modify the form layout.
4. *Consistency in electronic services.* With the proposed approach accumulated knowledge is maintained in a central repository, facilitating consistency in the way knowledge is represented and updated. This has a direct impact on the way services are implemented, since a single change in the knowledge base will affect all related electronic services. For example, if the method for validating a field is changed due to a legislation alteration, all forms and subsequently all services encompassing this field will be automatically adjusted without any further need for modifications. Furthermore, this centralised approach can also assure the uniformity in the look and feel of all the electronic services. With previous approaches, each electronic service was implemented separately and special care had to be taken in order for the look and feel to remain consistent. Whenever a need for a change occurs a single modification will affect the look and feel of all electronic services.
5. *Enhanced user interfaces.* The proposed platform enables the direct exploitation of the recorded knowledge for the creation of user interfaces that will provide assistance to service users in the form filling process. Non-interactive submission interfaces are also a significant aid to value-added professionals, since they can automate submissions through the service, skipping the tedious process of form filling. End-user polls have shown that these two features were of major importance to service users.

6. *Support of multiple dissemination platforms.* The architecture of the system enables the implementation of electronic services to a variety of media. Through appropriate content generators that can be included in the service access point, electronic services can be delivered through a variety of channels, such as the WWW and WAP. This feature has only partially been assessed, since the restricted capabilities of WAP client devices only allow the delivery of simple services to them.

7 Conclusions

In this paper we presented experiences from developing and maintaining a set of electronic services for the Greek Ministry of Finance. The traditional software engineering approaches employed in the first development phases proved to be inadequate in handling all aspects related to the lifecycle for electronic services. In the second phase we used a new approach, together with appropriate software tools, which allowed for using higher levels of abstraction, enhancing thus the maintainability, portability and reusability of the project's results, and reducing overall development time.

Future plans include integration with legislation databases, a feature that will provide the service users with direct links to documents governing the service use and semantics, but will also enable the automatic tracking of portions of the services that have become outdated due to changes in the supporting legislation. The integration with the organisation's workflow systems needs also to be elaborated on, enabling the unification of the processes that are applied to information flows emanating from paper-based services and electronic services.

8 References

Accelio Worldwide, 2001. Accelio products presentation. http://www.accelio.com/products/index.cfm

Cognos, 2002. Cognos Business Inteligence. http://www.cognos.com/km/gcn_excerpt.html

Computer Sciences Corporation, 2002. CSC Knowledge Management. http://www.csc.com/solutions/knowledgemanagement/

DECOR Project, 2002. Delivery of context-sensitive organisational knowledge. http://www.dfki.uni-kl.de/decor/KM-EUROPE-2000/decor-flyer.jpg

eEurope, 2000. Common list of basic public services. http://europa.eu.int/information_society/eeurope/action_plan/pdf/basicpublicservices.pdf

eEurope, 2001. eGovernment indicators for benchmarking eEurope. http://europa.eu.int/information_society/eeurope/action_plan/pdf/egovindicators.pdf

European Commission, 1999. Public Sector Information: A Key Resource for Europe, Green paper on Public Sector Information in the Information Society. http://europa.eu.int/ISPO/docs/policy/docs/COM(98)585/

General Secretariat for Information Systems, 2001a. Results of the User Poll Regarding the Electronic Tax Return Submission Service. http://www.gsis.gov.gr/04/poll/e1poll.html (in Greek)

General Secretariat for Information Systems, 2001b. Suggestions for Improving the Electronic Tax Return Submission Service. http://www.gsis.gov.gr/05/e1-eval/assess.html (in Greek)

Information Management Research, 2002. Alchemy product white paper. http://www.imrgold.com

International Standards Organisation, 1999. ISO 13407:1999: Human-centred design processes for interactive systems.

Jupp V., 2001. eGovernment – Lessons Learned, Challenges Ahead, eGovernment Conference: From Policy to Practice, 29-30 November 2001, Charlemagne, Brussels. http://europa.eu.int/information_society/eeurope/egovconf/documents/eGovernment%20Conference%20-%20Speech%20of%20Vivienne%20Jupp.doc

Know-Net Consortium, 2002. Manage Knowledge for Business Value. http://www.know-net.org/overview.shtml

Oracle Corporation, 2001. Oracle E-Business Suite. http://www.oracle.com/applications /index.html?content.html

Robben F., 2001. (Re)-organising for better services, eGovernment Conference: From Policy to Practice, 29-30 November 2001, Charlemagne, Brussels. http://europa.eu.int/information_society/eeurope/egovconf/documents/ppt/ Annex_2_Benchmarking_Robben_presentation_30-11-2001.ppt

Schemantics Inc. 2001. XMLForms. http://www.schemantix.com/product /xmlforms.html

Sommerville I. 2000. Software Engineering (6th Edition). Addison-Wesley Pub Co, ISBN: 020139815X.

TheBrain Technologies Corporation 2002. TheBrain Enterprise Knowledge Platform. http://www.thebrain.com/

W3 Consortium, 2002. XForms - The Next Generation of Web Forms. http://www.w3c.org/MarkUp/Forms

Web Accessibility Initiative, 1999. Web Content Accessibility Guidelines 1.0. http://www.w3.org/TR/WCAG10/