André Vasconcelos, Pedro Sousa, José Tribolet

Enterprise Architecture Analysis

An Information System Evaluation Approach

The definition of an Enterprise Architecture (EA) has a central role in implementing Information Systems (ISs) that proactively contribute to business and Information Technology (IT) alignment. In this paper, using a set of key concepts for the Information System Architecture (ISA) specification (formalized in a UML profile), 16 metrics are proposed for ISA suitability assessment considering a set of required quality characteristics. This paper also proposes an exploratory approach for the ISA definition process. This approach combines EA primitives and metrics in order to support multi-criteria ISA selection. The proposals presented in this research are applied in the definition, evaluation and comparing of different architectural options for the new Portuguese National identification document project (the Citizen Card project).

1 Introduction

The Information System Architecture (ISA) definition has a key role in the implementation of Information Systems (ISs) that actively contribute to business and Information Technology (IT) alignment – ensuring that the ISs are robust, IT independent, flexible and adaptable to business needs.

The research described in this paper, supported on a set of information system founding concepts (or architectural primitives), contributes on the estimation of the ISA suitability for a set of quality characteristics.

Using a UML profile for ISA modelling, we will present a set of metrics for assessing the ISA qualities; these metrics are formalized using OCL and are integrated in an approach for constructing an ISA, with specific architecture evaluation tasks.

In the next section an ISA modelling framework, that considers the enterprise, business, organizational, information system, information, application and technology concepts, is introduced. In the third section we formalize a coherent set of ISA evaluation metrics, supported in the framework introduced in the previous section. Next section describes the metrics application to the Portuguese Citizen Card Project. Finally we draw the major conclusions of this research and propose future exploration paths.

2 ISA Modelling

Nowadays the ISA is considered a critical success factor for the implementation of the organizational business strategy. According to [Zach97], ISA is "the determining factor, the factor that separates the winners from the losers, the successful and the failures, the acquiring from the acquired, the survivors from the others". In spite of the potential advantages of defining an ISA - namely better alignment between business and IT, interface and integration cost reductions, coherent data sharing and cheaper IT maintenance - there is currently no standard organizational praxis to define it. Moreover, the tools for representing an ISA, at informational, application and technological levels, and its dependences with business level, in a standard, normalized and simple way are not available off-the-shelf for Enterprise Architects [Zach99], [Boar99].

Organizational Engineering Center (CEO), considering other authors' research ([ErPe00], [MCL+99], [OMG06a]), in [VCN+01], proposed a framework for enterprise modelling (CEOF 2001). The CEOF 2001 provided a restricted set of business objects, defined in a UML profile [OMG06a], used for Enterprise modelling. CEOF 2001 has been extended for better supporting ISA modelling, at informational, application and technological levels, as well as its relationships with the business model.

Figure 1 introduces the core concepts proposed in the CEOF (simplified) metamodel for ISA modelling – integrated in an Enterprise Architecture (EA) modelling framework. The three major architecture levels that define the EA are: the Business, Organizational and Information Systems Architectures. At ISA level the concepts of information entity («Information Entity»), application block («IS Block»), technology block («IT Block») and business service («Business Service») are the root primitives for expressing the Information Architecture, the Application Architecture, the Technology Architecture and the relations between the ISA and the business, respectively. Thus, CEOF proposes to describe the ISA from its information, application and technology components. Furthermore, the relation between the ISA and the business model is accomplished using services that directly support business process needs (even between the ISA primitives the service concept is also used to express the dependencies and relations).

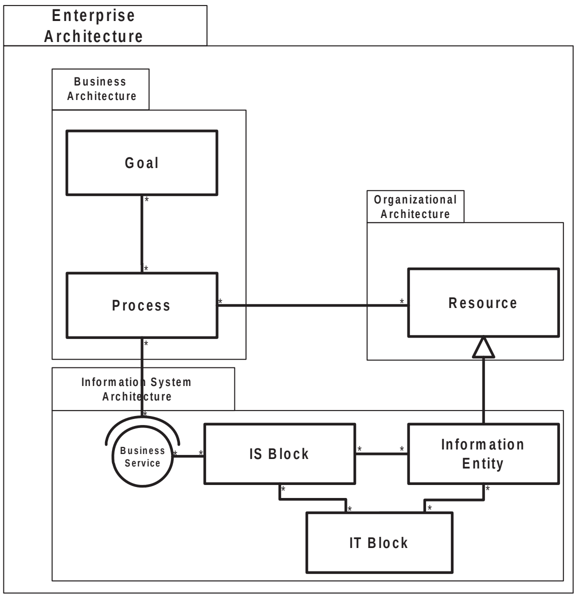

Figure 1: Simplified metamodel of CEO Framework

For further detail on the CEO framework architectural primitive (including its UML profile, attributes, OCL restrictions, associations, shapes, and presentation options) see [Vasc06], [STV+06], [VaST03] and [VCN+01].

3 ISA Evaluation

Though Information System Architecture (ISA) is currently recognized as an essential step in the process of building Enterprise Information Systems (EIS) aligned with business needs, there are not tools that support the Information System (IS) architect in accessing (during "design time") the impact of his or her decisions on the global ISA. Moreover, other ISA stakeholders that might have limited knowledge on ISA matters (as business people, software engineers, infrastructure experts) do not have simple methods or tools to quickly and automatically evaluate an ISA considering a set of desired IS qualities driven from the business context.

Using the CEO UML profile for ISA (described in the previous section), we defined a set of metrics, specified in OCL [OMG06b] and supported on the architectural primitives and attributes of CEOF. The metrics are defined considering a set of ISA qualities and principles next introduced.

3.1 Information System Qualities

In the software engineering domain there is some consensus on the set of quality characteristics desired in a software program; these characteristics are specified in the international standard ISO 9126 [ISO01].

In the Information System domain this consensus and normalization do not exist; although related research, mostly, at technological level of the ISA, proposes some IS qualities – for example [KhHo04] presents a set of quality attributes for large scale software systems (as performance, scalability, portability, robustness, accuracy and reliability), the Open Group (using TOGAF *Framework)* also describes a set of qualities that the IS should present [OpGr03]. Though the technology architecture might be analyzed considering most of the ISO 9126 qualities, the application and information architectures qualities and the alignment between the enterprise architectures are not expressed in ISO 9126 qualities. Thus, in this paper, we extend the ISO 9126 for ISA evaluation. Besides changing the scope of each quality characteristic for the IS domain – namely, changing the words "product" or "software product", in each ISO 9126 quality, to "Information Systems" – we propose the following changes to the ISO 9126 qualities[1]:

- *Functionality* – Information Systems ability to provide services that fulfil strategies and business goals;
- *Interoperability* – Information Systems ability to interact at technical and semantic level [IDAB03].
- *Usability* – since this characteristic is not measurable in an ISA, it won't be considered when analyzing an ISA (as its sub characteristics).

1. Only the changes to ISO9126 qualities and new qualities are described.

In order to consider the alignment in an ISA a new quality characteristic is introduced: Alignment – ability of the ISA components to operate according to the requirements/resources requested/provided in some other architectural level with the goal to improve organizational performance over time (the alignment is consider between business, application, information and technology architectures).

3.2 Metrics for ISA Evaluation

According to Tom DeMarco "You cannot control what you cannot measure" [DeMa82]; the metrics aim is to provide information to support quantitative decision-making during the system lifecycle, in order to "measure" the architecture (and control it!).

In this paper, the authors enforce the metrics coherence and correctness using the rules and principles proposed in [AbCa94] and [Hend96].

The metrics next described were developed iteratively, using, as starting point, the research from specialists in specific domains and the adaptation and extension of several software engineering metrics to the ISA domain (providing the chance of reusing the consolidate knowledge from a related research area - software engineering).

This paper proposes 16 metrics for ISA evaluation that pretend to quantify the set of quality characteristics for ISs (based on ISO 9126) and on the architectural primitives specified in the CEOF UML profile. Each of the metrics proposed is focused on the evaluation of an ISA quality at a defined architectural level (information, application or technology architectures). Table 1 presents an integrated overview on the metrics proposed and the qualities addressed.

In order to ensure the metrics correctness and minimize misinterpretations, all the metrics are specified using OCL (for space limitations the OCL code is not presented in the article) and supported on the CEOF UML primitives (previously described).

The metrics are next described, according to the ISA quality assessed.

3.2.1 Functionality Metrics

BSRPF - Business Service Required and Provided Factor

Computation

The Business Service Required and Provided Factor is computed considering the number of business services required and not provided by the Information Systems.

$$BSRPF = 1 - \frac{\sum_{i=1}^{\#«Business\ Process»} \#«Business\ Service»RNI_i}{\sum_{i=1}^{\#«Business\ Process»} \#«Business\ Service»R_i}$$, where

#«Business Service»RNI_i – is the number of «Business Service» Required for supporting process i and Not Implemented.
#«Business Service»R_i – is the number of «Business Service» Required for supporting process i.
#«Business Process» – is the number of Business Processes

Support

This metric measures the alignment and the suitability of the services provided by the information systems to the business processes. This analysis is accomplished considering all the services required by processes that are not support by «business services» (or which «business services» are not implemented in any application component «IS Block») [MRTG00].

DIIEF - Different Implementations of Information Entity Factor

Computation

The Different Implementations of Information Entity Factor is computed counting, for each «Information Entity» the number of possible implementations in «Low Level Information Entity».

$$DIIEF = \frac{\#«Information\ Entity»}{\sum_{i=1}^{\#«Information\ Entity»} NLLIE_i}$$, where

$NLLIE_i$ – is the number of «Low Level Information Entity» associated to the «Information Entity»i through the «implements» relation.
#«Information Entity» – is the total number of «Information Entity» in the ISA.

Support

This metric measures the number of different implementations that exist for each information entity. According to [Inmo00], for each information entity ("top level") there might be other entities that implement it ("low level information entity"). The existence of different «Low Level Information Entities» points to semantic problems for that «Information Entity» (e.g., by using different formats or attributes in the implementation of the information entity).

Table 1: Relation between the metrics proposed and the ISA architectural level and qualities

ISA Quality		Metric	Arch. Level
Functionality	Suitability	BSRPF - Business Service Required and Provided Factor	Business Application
	Interopera-bility (semantic and technical)	DIIEF - Different Implementations of Information Entity Factor	Information
		DTISSF - Distinct Technologies for IS Services Factor	Application Technology
	Security	SCBITABF - Security Components Between «IT Application Block» Factor	Technology
		IASF – Information-Application Security Factor	Information Application
Reliability	Fault Tolerance	ITRF – IT Redundancy Factor	Technology
Efficiency	Resource Behaviour	SITPLBF - Stateful «IT Presentation Block» and «IT Logic Block» Factor	Technology
Maintainability	Analyzability	SCCF - Service Cyclomatic Complexity Factor	Business Application
	Changeability	LCOISF - Lack of COhesion in «IS Block» Factor	Application Information
		NOISF - Number of Operations in «IS Block» Factor	Application
	Testability	RSF - Response for a Service Factor	Application
Portability	Adaptability	POSF - Possible Operating Systems Factor	Technology
Alignment	Business/ System Alignment	CPSMF - Critical Process - System Mismatch Factor	Business Application
	Information/ Application Alignment	NAIEF - Number of Applications for «Information Entity» Factor	Application Information
	Information/ Technology Alignment	LLIEITBDTMF - Low Level Information Entity – IT Block Data Type Mismatch Factor	Information Technology
	Application/ Technology Alignment	CSTMF – Critical System - Technology Mismatch Factor	Application Technology
Dimension		NE – Number of Entities	Information
		NA – Number of Applications	Application
		NITB - Number of IT Blocks	Technology

DTISSF - Distinct Technologies for IS Services Factor

Computation

The Distinct Technologies for IS Services Factor is computed counting for each «IS Service» the number of «IT Service» of type "Integration Service".

$$DTISSF = 1 - \frac{\#«IS\ Service»}{\sum_{i=1}^{\#«IS\ Service»} \#«IT\ Service»_{Integration\ i}}$$, where

#«IT Service» Integration $_i$ – is the number of «IT Service», which attribute serviceType is equal to "Integration Service" that implements the «IS Service» i.
#«IS Service» – is the number of «IS Service» in the ISA.

Support

The technical interoperability of a software architecture increases by providing the same interface in different technologies [SaSu03]. In the same way, with this metric the technical interoperability and portability of an EIS is analyzed as the average of the Technologies that each application interface provides.

SCBITABF - Security Components Between «IT Application Block» Factor

Computation

The Security Components Between «IT Application Block» Factor is computed counting, for each «IT Application Block» the minimal number of «IT Block», which attribute "securityElement" is true, that is between the path of that block and each of the remaining «IT Application Block».

$$SCBITABF = \frac{\sum_{i=1}^{\#IT\ Application\ Block} \left[\sum_{j=1}^{\#«IT\ Application\ Block»} Min\ \{\#SITB_{ij}\} \right]}{\#«IT\ Application\ Block» \times \#«ITBlock»}$$, where

Min{ $\#SITB_{ij}$ } – is the minimal number of instances of «IT Block», which attribute "securityElement" has the value "true", and is in the path between «IT Application Block»i and «IT Application Block»j.
#«IT Application Block» – is the number of instances of «IT Application Block».
#«IT Block» – is the number of instances of «IT Block».

Support

The ISA security is increased by putting security elements on it, as IDS, firewalls, etc. Thus, this metric, is not limited to counting the number of security components but it also considers, for each application component, the number of security components that isolate it from other components.

IASF – Information-Application Security Factor

Computation

The Information-Application Security Factor is computed considering the number of information entities with high level security requirements supported in «IS Blocks» that also support information entities without high security requirements and vice versa.

$$IASF = 1 - \frac{\#\{InformationEntity_s \in ISBlock_{NS}\} + \#\{InformationEntity_{NS} \in ISBlock_S\}}{\#«InformationEntity»}$$, where

$\#\{InformationEntity_S \in ISBlock_{NS}\}$ – is the number of «Information Entities» that its Security attribute value is {Yes} supported in «IS Blocks» that support other «Information Entities» which Security attribute value is {No}; where an «Information Entity» is "supported" by an «IS Block» if and only if exists at least one «operation» provided by the «IS Block» that CUD the «Information Entity».
$\#\{Informatio\ nEntity_{NS} \in ISBlock_S\}$ – is the number of «Information Entities» that its Security attribute value is {No} supported in «IS Blocks» that support other «Information Entities» which Security attribute value is {Yes}; where an «Information Entity» is "supported" by an «IS Block» if and only if exists at least one «operation» provided by the «IS Block» that CUD the «Information Entity».
#«Information Entity» – is the number of information entities

Support

According to [SoPM04] applications should manage information entities of the same security level.

3.2.2 Reliability Metrics

ITRF – IT Redundancy Factor

Computation

The IT Redundancy Factor is computed counting the «IT Block» which attribute "redundantElement" is true.

$$ITRF = \frac{\#RITB}{\#«IT\ Block»}$$, where

#RITB – is the number of «IT Block» which attribute "redundantElement" has the value true.
#«IT Block» – is the number of «IT Block».

Support

The Fault tolerance of an ISA tends to increase by using redundant elements [Varg00]. This metrics considers this fact.

3.2.3 Efficiency Metrics

SITPLBF - Stateful «IT Presentation Block» and «IT Logic Block» Factor

Computation

The Stateful «IT Presentation Block» and «IT Logic Block» Factor is computed counting the number of «IT Presentation Block» and «IT Logic Block» that its attribute "state" value is "stateful"

$$\text{SITPLBF} = 1 - \frac{\#SITPLB}{\#«IT\ PresentationBlock» + \#«IT\ LogicBlock»}$$, where

#SITPLB – is the number of «IT Presentation Block» and «IT Logic Block» that its attribute "state" value is "stateful".
#«IT Presentation Block» – is the number of «IT Presentation Block».
#«IT Logic Block» – is the number of «IT Logic Block».

Support

The Scalability of an EIS is increased if business and presentation components do not keep the state (since it will be easier for implementing new parallel instances of these ISA components) [BEA06].
The Scalability of an ISA tends to grow if the «IT Presentation Blocks» and the «IT Logic Blocks» do not preserve the application state (stateless) – the «IT Data Blocks» should be the ones to keep application state.

3.2.4 Maintainability Metrics

SCCF - Service Cyclomatic Complexity Factor

Computation

The Service Cyclomatic Complexity Factor is computed considering the number of dependencies between «IS Blocks» subtracted by the number of «IS Blocks» that support the service, for each service.

$$SCCF = \frac{\#«Business\ Service» + \#«IS\ Service»}{\sum_{i=1}^{\#«Business\ Service» + \#«IS\ Service»} |e_i - n_i + 2|}$$, where

e_i – is the number of dependencies between «IS Block» for the service i.
n_i – is the number of «IS Blocks» that support the service i.
#«Business Service» – is the number of «Business Services».
#«IS Service» – is the number of «IS Services»

Support

Like [McCa76], for the software engineering area, considering that the higher the number of paths in a program, the higher its control flow complexity probably will be, in [VaST05] is proposed a similar metric for evaluate the complexity of an ISA in the support of the business services – considering that the complexity, for each service, is measure by the difference between the number of dependencies and applications involved.

LCOISF - Lack of COhesion in «IS Block» Factor

Computation

The Lack of COhesion in «IS Block» Factor is computed counting the number of sets of information entities that are used by distinct functionalities of the same application (provided by operations in «IS Blocks»)

$$LCOISF = 1 - \frac{\sum_{i=1}^{\#IS\ Block} \#LCOIS_i}{\#«IS\ Block» \times \#«IS\ Operation» \times \#«Information\ Entity»}$$, where

$\#LCOIS_i$ – is the number of sets of «Information Entities» that are used by «operations» distinct of the «IS Block» i;.
#«IS Block» – is the number of «IS Blocks».
#«IS Operation» – is the number of «IS Operation».
#«Information Entity» – is the number of « Information Entity».

Support

This metric measure the correlation between application blocks and the information entities used in that application block.
It is quantified by the average of the number of sets of information entities that are used by distinct operations of the same application [VaST05].

NOISF - Number of Operations in «IS Block» Factor

Computation

The Number of Operations in «IS Block» Factor is computed dividing the number of «IS Blocks» by the number of operations on each «IS Block».

$$NOISF = \frac{\#«IS\ Block»}{\sum_{i=1}^{\#IS\ Block} \#«IS\ operation»_{«ISBlock»i}}$$, where

$\#«operation»_{«IS\ Block»i}$ – is the number of operations on «IS Block» i.
#«IS Block» – is the number of «IS Block»

Support

The simplicity to adapt/improve operations in an ISA to new business demands is maximized when the impact of changing each operation is reduced to a single application block («IS Block»). This metric measures this fact.
This metric was defined considering the similar software engineering metric "Average number of methods per class", that considers the existing methods in each class [AbCa94].

RSF - Response for a Service Factor

Computation

The Response for a Service Factor is computed by considering the average of the number of «IS Blocks» that might be used to support each «Service».

$$RSF_{ISA} = \frac{\#«Business\ Service»+\#«IS\ Service»}{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} \#«IS\ Block»_i}$$, where

#«IS Block»$_i$ _ is the number of «IS Blocks» involved in supporting service i.
#«Business Service» – is the number of «Business Services».
#«IS Service» – is the number of «IS Services».

Support

This metric is similar to the software metric "Response For a Class" – see [ChKe94] and [BaBM96] for further details – that computes the number of methods that can potentially be executed in response to a message received. In [VaST05] this metric is proposed (Average Response for a Service) and it computes the number of «IS Blocks» that might be used to support a service.
Recent research [SoPM04] suggests that each business process should be supported by the less number of applications as possible – which is measured by this metric.

3.2.5 Portability Metrics

POSF - Possible Operating Systems Factor

Computation

The Possible Operating Systems Factor is computed by counting, on each application («IT Application Block»), the number of possible operating systems (families).

$$POSF = 1 - \frac{\#«IT\ Application\ Block»}{\sum_{i=1}^{\#«IT\ Application\ Block»} NPOS_i}$$, where

$NPOS_i$ – is the number of possible operating systems families that the «IT Application Block»$_i$ supports.
#«IT Application Block» – is the number of «IT Application Block» in the ISA.

Support

The portability and Technical Interoperability in an ISA increase with the number of possible platforms where ISA components are able to operate [SaSu03].

3.2.6 Alignment Metrics

CPSMF - Critical Process - System Mismatch Factor

Computation

The Critical Process - System Mismatch is computed by counting the number of critical business processes supported by «IS Blocks» that also support non-critical business processes and the number of non-critical business processes supported by «IS Blocks» that also support critical business processes.

$$CPSMF = 1 - \frac{\#\{Process_C \in ISBlock_{NC}\} + \#\{Process_{NC} \in ISBlock_C\}}{\#«Process»}$$, where

$\#\{Process_C \in ISBlock_{NC}\}$ – is the number of critical processes supported by «IS Blocks» that support other non-critical processes.
$\#\{Process_{NC} \in ISBlock_C\}$ – is the number of non-critical processes supported by «IS Blocks» that support other critical processes.
#«Process» – is the number of processes.

Support

As described in [SoPM04] the critical business processes should be supported by different applications than non-critical business processes.

NAIEF - Number of Applications for «Information Entity» Factor

Computation

The Number of Applications for «Information Entity» Factor is computed counting the average number of applications («IS Blocks») that through its «operations» support each «information entity».

$$NAIEF = \frac{\#«Information\ Entity»}{\sum_{i=1}^{\#Information\ Entity} \left[\#ISBlocks \in \{\exists«ISOperation»\ CUD\ «InformationEntity»_i\}\right]}$$, where

$\#ISBlocks \in \{\exists«operation»\ CUD\ «InformationEntity»_i\}$ – is the number of «IS Blocks» in which exists an «operation» that CUD (Creates, Updates or Deletes) the «information entity» i.

#«Information Entity» – is the number of «Information Entities».

Support

According to [SoPM04] each information entity should be managed by a single application.

LLIEITBDTMF - Low Level Information Entity – IT Block Data Type Mismatch Factor

Computation

The Low Level Information Entity – IT Block Data Type Mismatch Factor is computed counting the number of «Low Level Information Entity» that are "primitive" that (according to the attribute "informationEntityData") supported in «IT Block» that also support «Low Level Information Entity» that are "derived" and vice versa.

$$\text{LLIEITBDTMF} = 1 - \frac{\#\{LowLevelInformationEntity_P \in ITBlock_{NP}\}}{\#«LowLevelInformationEntity»} + \frac{\#\{LowLevelInformationEntity_D \in ITBlock_{ND}\}}{\#«LowLevelInformationEntity»}$$

, where

$\#\{LowLevelInformationEntity_P \in ITBlock_{NP}\}$ – is the number of «Low Level Information Entity» which informationEntityData attribute is {Primitive} supported in «IT Block» that support other «Low Level Information Entity» which attribute informationEntityData is different from {Primitive}.
$\#\{LowLevelInformationEntity_D \in ITBlock_{ND}\}$ – is the number of «Low Level Information Entity» which informationEntityData attribute is {Derived} supported in «IT Block» that support other «Low Level Information Entity» which attribute informationEntityData is different from {Derived}.
#«Low Level Information Entity» – number of low level information entities.

Support

According to Inmon the primitive and derived data present important differences on performance, accessing patterns, availability, among others issues [Inmo92].
Thus, is considered a "good architectural practice" to use different technology components to support primitive and derived data – using this metric it is possible to measure the alignment between the technology and the information architectures.

CSTMF – Critical System - Technology Mismatch Factor

Computation

The Critical System - Technology Mismatch Factor is computed counting the number of «IS Block» considered critical (for supporting exclusively critical processes) supported in «IT Block» that support «IS Block» not critical and vice versa.

$$CSTMF = 1 - \frac{\#\{ISBlock_C \in ITBlock_{NC}\} + \#\{ISBlock_{NC} \in ITBlock_C\}}{\#«ISBlock»}$$

, where

$\#\{ISBlock_C \in ITBlock_{NC}\}$ – is the number of «ISBlock» considered critical supported in «IT Block» that support other «ISBlock» besides the critical ones (where an «IS Block» is considered critical if it supports exclusively critical «process»).
$\#\{ISBlock_{NC} \in ITBlock_C\}$ – is the number of «ISBlock» considered not critical supported in «IT Block» that support other «ISBlock» besides the non-critical ones (where an «IS Block» is considered not critical if it supports exclusively not critical «process»).
#«IS Block» – is the number of «IS Block».

Support

As critical business processes should be supported by different applications than non-critical business processes [SoPM04], these applications should be implemented in technology components different than the ones that implement applications that support non-critical business processes.

3.3 An Exploratory Approach for Building an ISA

Considering the evaluation metrics previously introduced the authors propose an exploratory approach for defining and selecting an ISA according to a set of quality characteristics.

The "traditional" process for building an ISA is described in Figure 2 (a); in the "traditional" approach the ISA is defined and implemented without considering the qualities it should provide. In Figure 2 (b) a new approach is proposed that – besides modelling the ISA using a set of formalized UML primitives – has explicit control and evaluation tasks on the architecture quality (providing feedback to the ISA definition process).

This (exploratory) approach provides a set of conceptual tools for the "IS architects" use the primitives and metrics developed in our research. The approach is based on the major steps and tasks of Spewak Enterprise Architecture Planning (EAP) methodology [SpHi92] (however it could be applied to other archi-

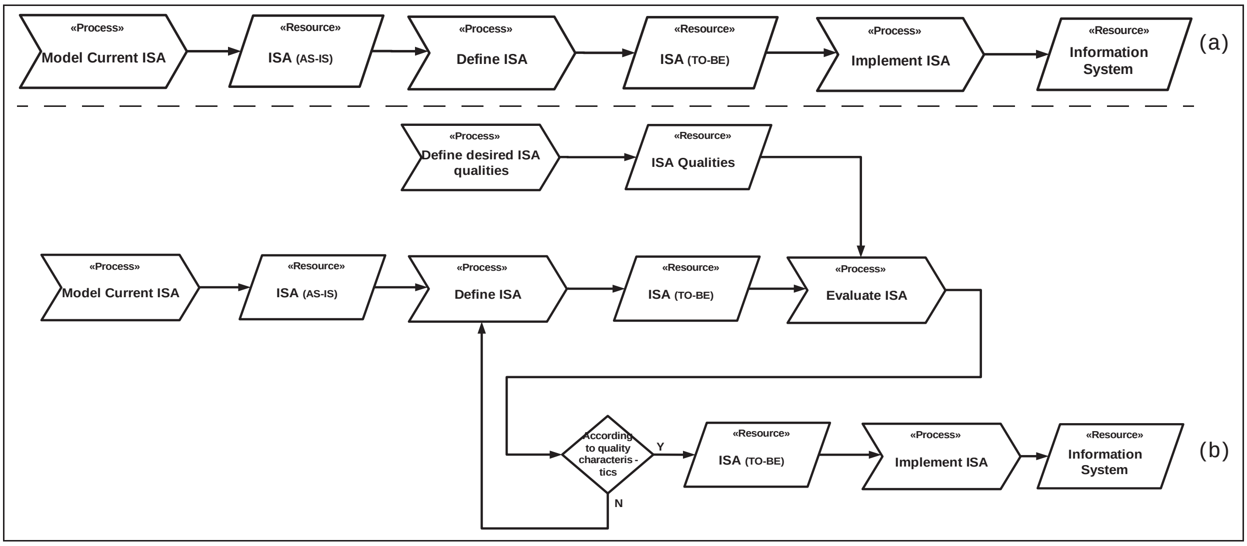

Figure 2: Traditional (a) and proposed (b) approach for building an ISA

tecture building approaches). Next we will just identify the changes proposed to the EAP approach.

Thus, for modelling the current ISA and business processes the task suggested by [SpHi92] should be accomplished, but using the CEOF modelling primitives to represent the current situation.

In the definition of the ISA, a new step should be added: the identification of the qualities that the ISA should fulfil. The architecture team should weight each quality (according to its importance), considering the qualities and ISA levels identified in Table 1.

After this step, the definition of the ISA should be developed according to EAP methodology, by modelling the information, application and technology architectures (always using CEOF UML primitives); each architecture (at information, application and technology levels) should be evaluated (using the metrics described in the previous section) and compared to the current ("AS-IS") architecture or against other possible design options presented by the project team (see Figure 3 (a)).

Finally, the project team should apply all the metrics introduced in the previous section (including the dimension metrics), at each (possible) architecture.

After the relative weight of each quality characteristic is combined with each metric, a final value is obtained. In order to select the "best" architecture it is recommended that the project team develop a cost-benefit analysis (obtaining the cost of each architecture using the dimension metrics) and perform a sensibility analysis (in order to better understand the impact of the architecture's design choices on the alternative ISA) - see Figure 3 (b).

This approach is used in the Portuguese Citizen Card Architecture described next.

4 The Portuguese Citizen Card Project

This section presents the result of using the primitives, metrics and approach, previously described, for defining the ISA of a Portuguese Government project.

4.1 The Project Context

The Portuguese Government decided to implement a new mandatory national identification card that combines and replaces five identification cards (that current the citizens need for interact with the public administration). This card is a physical high secure document (that allows the visual identification of a citizen) and it is also a digital document (that allows the citizen to identify himself/herself and to electronically sign documents) – see Figure 4.

Considering the technological, architectural, legal, and political challenges of the project and the short time available for its implementation (the project definition started in June 2005 and the system should be issuing real cards by the end of 2006) the project team decided to develop a Proof of Concept (PoC), based on the know-how of different companies and public agencies and on the best practices of other similar projects. The ISA defined for the PoC was focussed on "demonstrating the concepts" of electronic identity and interoperability, and was not much

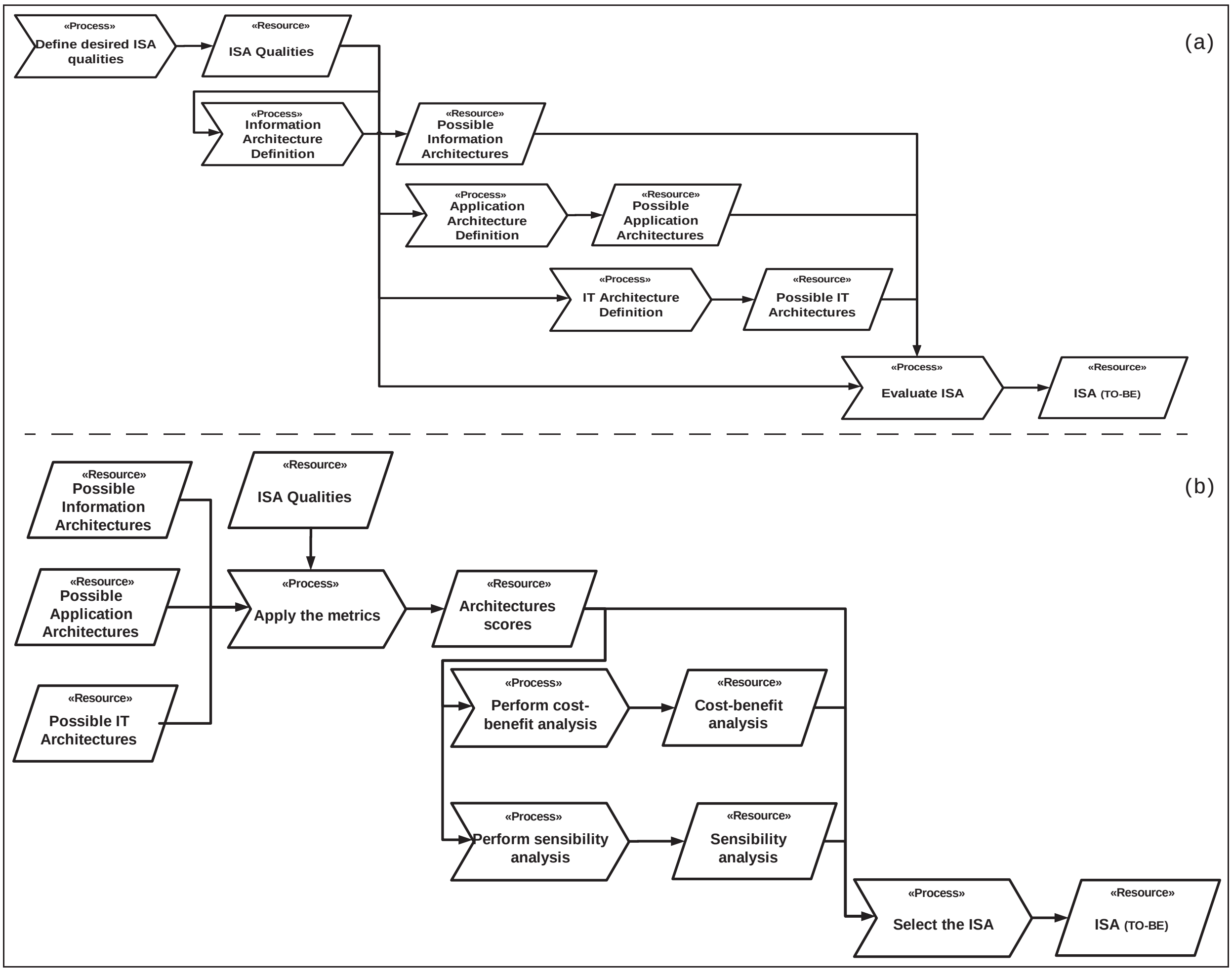

Figure 3: Detailing sub-processes of the "Define ISA" (a) and "Evaluate ISA" (b) processes (part of the building ISA approach proposed)

concerned on the reliability, security, portability or efficiency of the resulting information systems.

Simultaneously with the PoC ISA implementation (for demonstration purposes) the project team modelled the business processes and specified an ISA (that considered the PoC architecture with some changes) – we will label this the "final ISA". The final ISA was defined considering the impact of other architectural options on the PoC architecture – according to the metrics, notation and approach described in the previous sections.

We describe this approach and the architectural results accomplished[2].

Figure 4: Portuguese Citizen Card (CC)

4.2 The Business Architecture

Figure 5 presents the top level business process of the citizen card request and use. The project team detailed this (and other) process – however considering the focus of this paper we will not present further details on the business architecture (but on the ISA).

2. Considering the size and complexity of this project, only part of the metrics applied are presented in this paper.

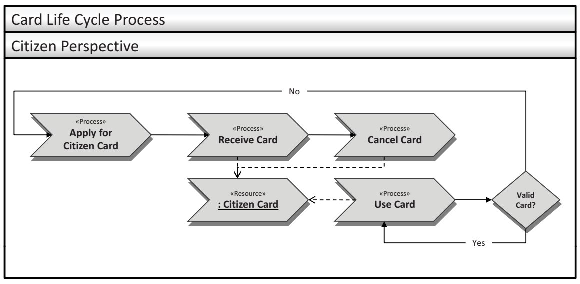

Figure 5: Citizen Card high level business process

Table 2: Qualities weights defined for the CC ISA

	Import. (0-10)	Sub-Quality	Import. (0-10)	Architectural Level	Import. (0-10)
Functionality	9	Suitability	9	Business Application	--
		Semantic Interoperability	3	Information	--
		Technical Interoperability	3	Application Technology	--
		Security	10	Technology	8
				Information Application	5
Reliability	5	Fault Tolerance	--	Technology	--
Efficiency	4	Resource Behaviour	--	Technology	--
Maintainability	6	Analyzability	3	Business Application	--
		Changeability	7	Application Information	5
				Application	5
		Testability	8	Application	--
Portability	3	Adaptability	--	Technology	--
Alignment	7	Business/ System Alignment	8	Application Information	--
		Information/ Application Alignment	8	Information Technology	--
		Information/ Technology Alignment	8	Application Technology	--
		Application/ Technology Alignment	8	Application Information	--

4.3 Selecting the "right" ISA

4.3.1 *ISA Qualities Identification*

According to the approach previous described, the project team defined weights for each quality characteristic (according to the project priorities and goals) – Table 2 presents the result. This table will support the selection of the ISA to implement.

4.3.2 *Information Architecture*

The information architecture defined for the PoC is modelled in Figure 6 (a). For the final ISA (Figure 6 (b)) the "Delivery Note" and "Receipt" were added to the architecture – that previously, for demonstration purposes, weren't considered important.

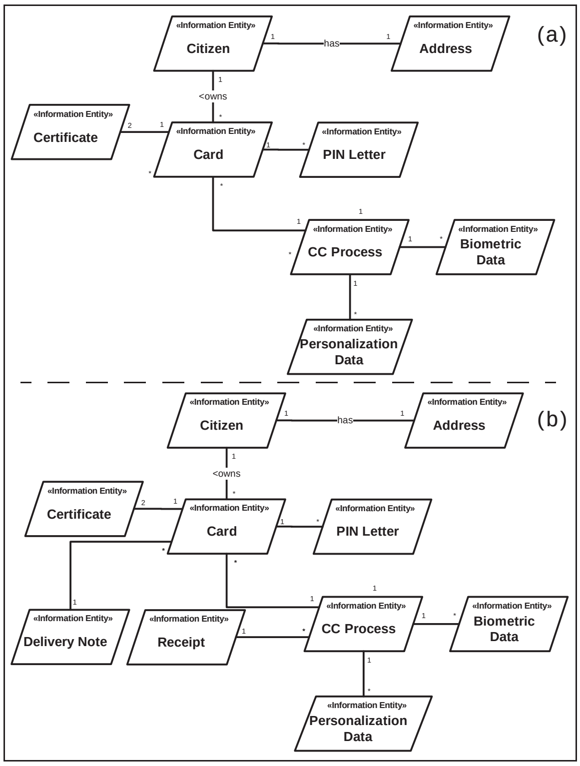

Figure 6: Citizen Card PoC (a) and final (b) Information Architecture

The «Low Level Information Entities», derived from each «information entity» were next identified. The PoC and the final information architectures present some differences at this level; for example, the format of the "Address" information entity, in the PoC was considered similar for all the IS (Figure 7 (a)); however, in the final ISA, most of the existing IS already had distinct attributes for the address (besides the need for keeping the successive citizen addresses during his/her life) – Figure 7 (b).

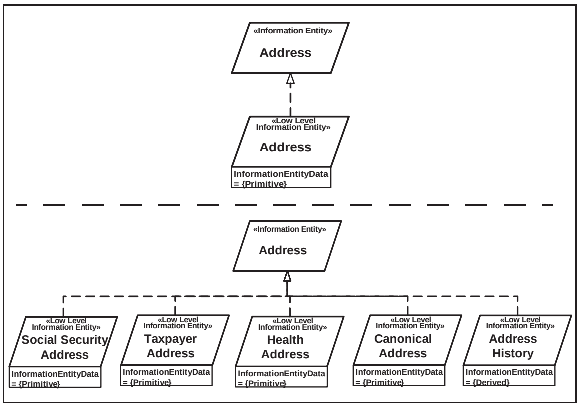

Figure 7: Address Low Level Information Entities for PoC (a) and final (b) ISA

Considering all the information entities and low level information entities, in order to quantify the semantic interoperability of the PoC and the final architecture, the metric DIIEF ("Different Implementations of Information Entity Factor") was computed.

For the PoC ISA:

$$DIIEF(PoC) = \frac{\#«Information\ Entity»}{\sum_{i=1}^{\#«Information\ Entity»} NLLIE_i} = \frac{8}{5+1+1+1+1+1+1+1} = \frac{2}{3} = 0{,}67$$

And for the Final ISA:

$$DIIEF(Final) = \frac{\#«Information\ Entity»}{\sum_{i=1}^{\#«Information\ Entity»} NLLIE_i} = \frac{10}{6+2+4+2+5+1+1+1+1+1} = \frac{5}{12} = 0{,}42$$

Thus, the semantic interoperability of the PoC is higher than the final ISA. However, the implementation of the PoC would imply changing the existing information systems (of the public agencies involved) – which was not considered in the project timeframe.

4.3.3 Application Architecture

The PoC and final application architectures are modelled in Figure 8 (a) and Figure 8 (b), respectively. As Figure 8 (a) presents, some application components were not consider in the PoC architecture (since they were not important to demonstrate the concepts) but revealed to be important to support the Citizen Card business processes (like transportation or payment activities).

Figure 9 and Figure 10 describe the application architecture support for the processes "Ask for Citizen Card", "Personalize Card" and "Get Citizen Card", in the PoC and final architectures – once again, in the PoC some business process components are not supported by the «IS Block».

Considering all the processes and their relations – besides the processes model in Figure 9 and Figure 10 – using the metric *BSRPF (Business Service Required and Provided Factor*) we have for PoC and final architectures:

$$BSRPF(PoC) = 1 - \frac{\sum_{i=1}^{\#«Business\ Process»} \#«Business\ Service»RNI_i}{\sum_{i=1}^{\#«Business\ Process»} \#«Business\ Service»R_i} = 1 - \frac{18}{32} = \frac{9}{16} = 0{,}56$$

$$BSRPF(Final) = 1 - \frac{\sum_{i=1}^{\#«Business\ Process»} \#«Business\ Service»RNI_i}{\sum_{i=1}^{\#«Business\ Process»} \#«Business\ Service»R_i} = 1 - \frac{0}{32} = 1$$

This metric values point that the ISA implemented in the PoC is less suitable to the business needs than the final ISA (that completely supports the business services required).

The dependencies between the information and application architectures are described in Figure 11. The information-application architectures alignment is analyzed using the *NAIEF* ("Number of Applications for «Information Entity» Factor") metric.

$$NAIEF\ (PoC) = \frac{\#«Information\ Entity»}{\sum_{i=1}^{\#Information\ Entity} \left[\#\ ISBlocks \in \{\exists «ISOperation»\ CUD\ «InformationEntity»_i\}\right]} \Leftrightarrow$$

$$NAIEF\ (PoC) = \frac{8}{6+2+1+1+1+1+1+1} = \frac{8}{14} = 0{,}57$$

$$NAIEF\ (Final) = \frac{\#«Information\ Entity»}{\sum_{i=1}^{\#Information\ Entity} \left[\#\ ISBlocks \in \{\exists «ISOperation»\ CUD\ «InformationEntity»_i\}\right]} \Leftrightarrow$$

$$NAIEF\ (Final) = \frac{10}{6+2+1+1+1+1+1+1+1+1} = \frac{10}{16} = 0{,}63$$

The *NAIEF* metric points out that the information-application alignment is higher in the final ISA.

The changeability, at information-application level, for both architectures is high, according to the LCOISF (Lack of COhesion in «IS Block» Factor) metric.

$$LCOISF(PoC) = 1 - \frac{\sum_{i=1}^{\#IS\ Block} \#LCOIS_i}{\#«IS\ Block» \times \#«IS\ Operation» \times \#«Information\ Entity»} \Leftrightarrow$$

$$LCOISF(PoC) = 1 - \frac{1+1+1+1+1+1+1+1+2+1+1}{10 \times 23 \times 8} = 1 - \frac{11}{1840} = 0{,}99$$

$$LCOISF(Final) = 1 - \frac{\sum_{i=1}^{\#IS\ Block} \#LCOIS_i}{\#«IS\ Block» \times \#«IS\ Operation» \times \#«Information\ Entity»} \Leftrightarrow$$

$$LCOISF(Final) = 1 - \frac{1+1+1+1+1+1+1+2+1+2+1+1+1}{12 \times 31 \times 10} = 1 - \frac{14}{3720} = 1{,}00$$

The high value of this metric, in both architectures, is a consequence of the high affinity of the operations for each «IS Block» - since, almost always, the operations in each «IS Block» share information entities (between themselves).

The changeability at application level can be estimated using the NOISF (Number of Operations in «IS Block» Factor) metric.

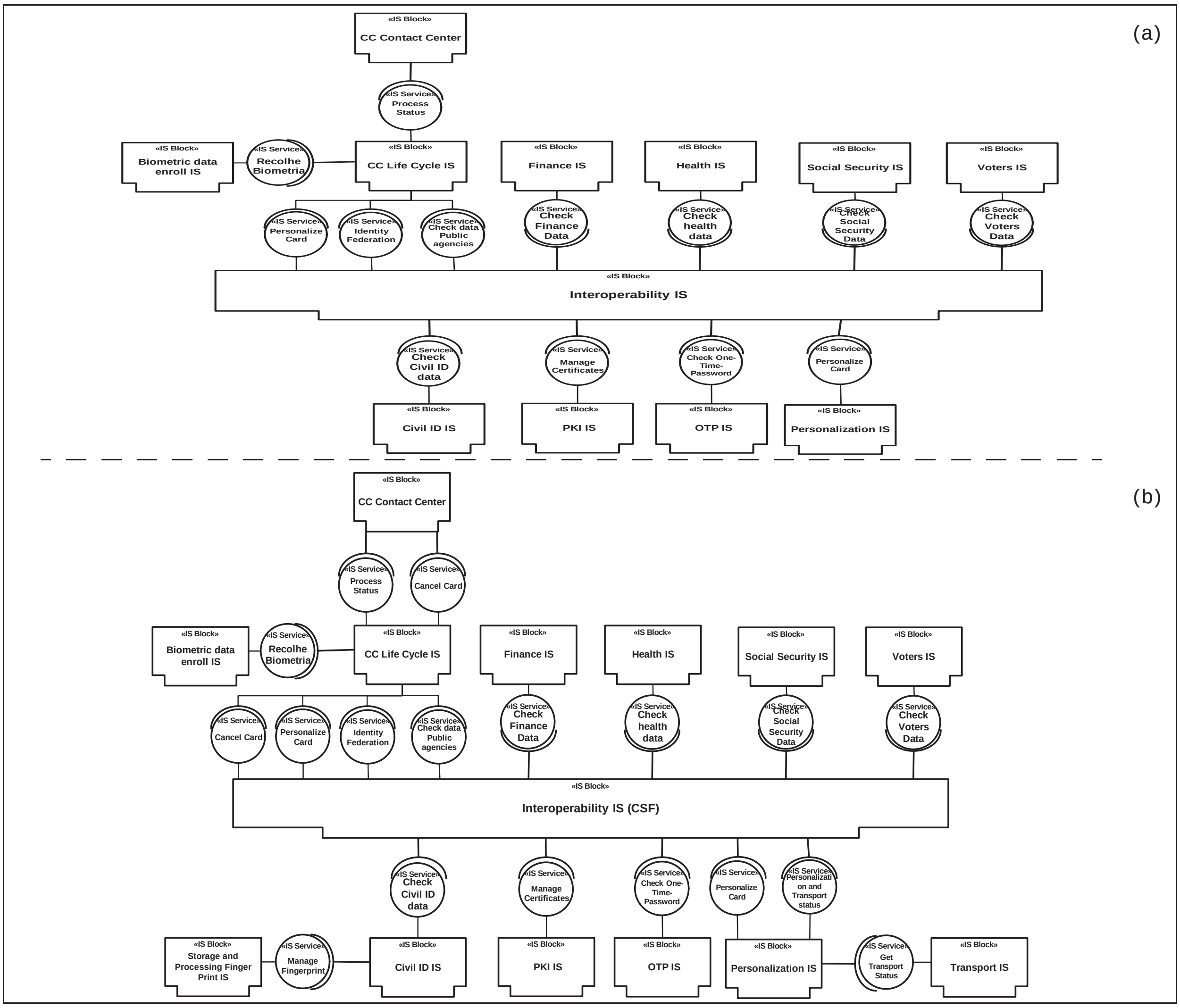

Figure 8: Citizen Card Application Architecture: PoC (a) and final (b) architecture

$$NOISF\ (PoC) = \frac{\#«IS\ Block»}{\sum_{i=1}^{\#IS\ Block} \#«IS\ operation»_{«ISBlock»i}} = \frac{10}{23} = 0{,}43$$

$$NOISF\ (final) = \frac{\#«IS\ Block»}{\sum_{i=1}^{\#IS\ Block} \#«IS\ operation»_{«ISBlock»i}} = \frac{12}{31} = 0{,}39$$

This metrics points out that the PoC ISA has a higher changeability (at application level), considering the smaller number of operations, per «IS Block».

The information-application security is high in both ISAs since all the information entities required to be managed in a high security environment. Thus the metric IASF (Information-Application Security Factor) presents a maximum value for both architectures:

$$IASF(PoC) = 1 - \frac{\#\{InformationEntity_s \in ISBlock_{NS}\} + \#\{InformationEntity_{NS} \in ISBlock_S\}}{\#«InformationEntity»} = 1 - \frac{0+0}{8} = 1$$

$$IASF(Final) = 1 - \frac{\#\{InformationEntity_s \in ISBlock_{NS}\} + \#\{InformationEntity_{NS} \in ISBlock_S\}}{\#«InformationEntity»} = 1 - \frac{0+0}{10} = 1$$

Figure 12 and Figure 13 present two sequence diagrams, for the PoC and the final architectures, which describe the system interaction for supporting the "ask for citizen card" business process.

Considering all the business processes (of the Citizen Card project), and the resulting sequence diagrams the project team estimated the following qualities for both architectures:

The Analyzability quality is estimated using the SCCF (Service Cyclomatic Complexity Factor) metric.

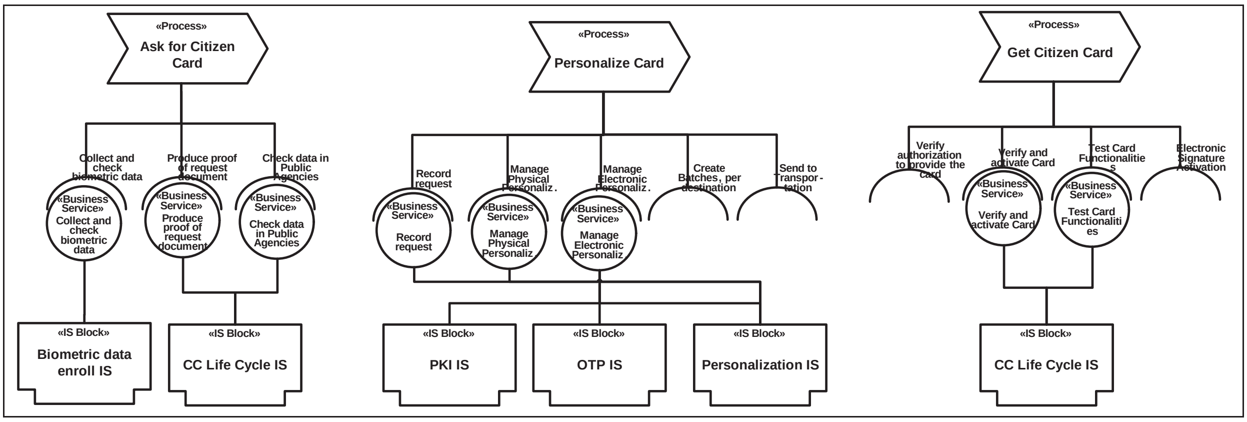

Figure 9: Business-application architecture dependencies (for the ask for citizen card, personalize card and get citizen card processes) of the PoC ISA

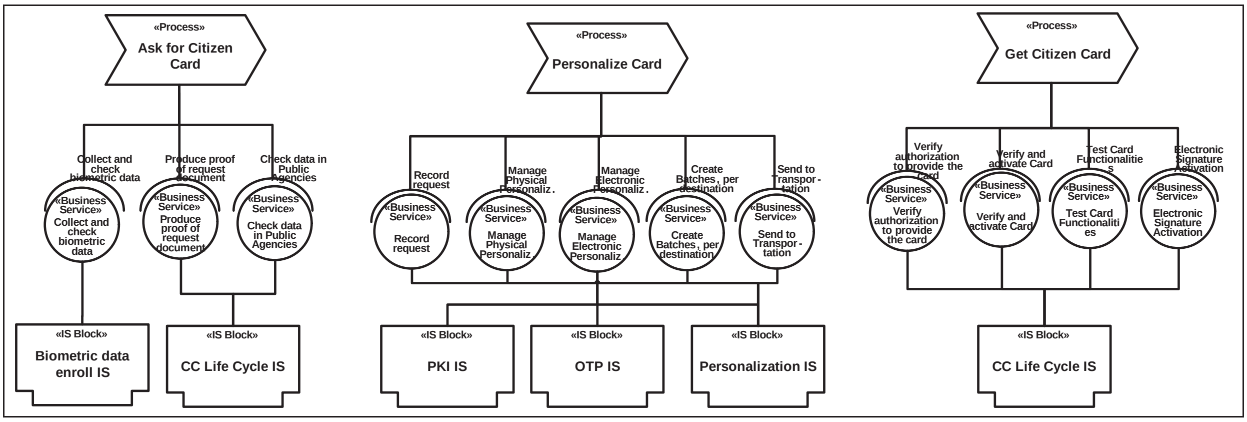

Figure 10: Business-application architecture dependencies (for the ask for citizen card, personalize card and get citizen card process) of the final ISA

$$SCCF\ (PoC) = \frac{\#«Business\ Service»+\#«IS\ Service»}{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»}|e_i - n_i + 2|} \Leftrightarrow$$

$$SCCF\ (PoC) = \frac{14+13}{(3+5+36+2+2+2+7+2+3+2+6+9+8+4)+(4+4+9+8+8+4+4+4+4+8+4+4+6)}$$

$$SCCF\ (PoC) = \frac{27}{162} = 0{,}17$$

$$SCCF\ (Final) = \frac{\#«Business\ Service»+\#«IS\ Service»}{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»}|e_i - n_i + 2|} \Leftrightarrow$$

$$SCCF\ (Final) = \frac{32+18}{(3+5+36+2+2+2+3+11+2+3+3+2+2+4+9+1+2+2+2+8+2+8+4+3+4+3+4+3+3+2+3+5)} +$$
$$+ \frac{32+18}{(4+5+4+5+9+8+8+4+4+4+4+7+5+8+4+4+6+7)} =$$

$$SCCF\ (Final) = \frac{50}{248} = 0{,}20$$

Thus, the final ISA, even supporting a higher number of business processes, presents a greater analyzability than the PoC architecture.

The maintainability of the architectures, namely the Testability is predicted using RSF (Response for a Service Factor) metric.

For the PoC architecture we have:

$$RSF(PoC) = \frac{\#«Business\ Service»+\#«IS\ Service»}{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»}\#«IS\ Block»_i} \Leftrightarrow$$

$$RSF(PoC) = \frac{14+13}{(2+1+7+5+5+5+7+1+1+3+3+5+4+1)+(2+1+3+6+6+1+1+1+1+2+1+1+2)} \Leftrightarrow$$

$$RSF(PoC) = \frac{27}{78} = 0{,}346$$

And for the final architecture:

$$RSF(PoC) = \frac{\#«Business\ Service»+\#«IS\ Service»}{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»}\#«IS\ Block»_i} \Leftrightarrow$$

$$RSF(PoC) = \frac{14+13}{(2+1+7+5+5+5+7+1+1+3+3+5+4+1)+(2+1+3+6+6+1+1+1+1+2+1+1+2)} \Leftrightarrow$$

$$RSF(PoC) = \frac{27}{78} = 0{,}346$$

Therefore the final ISA is easier to test than the PoC architecture.

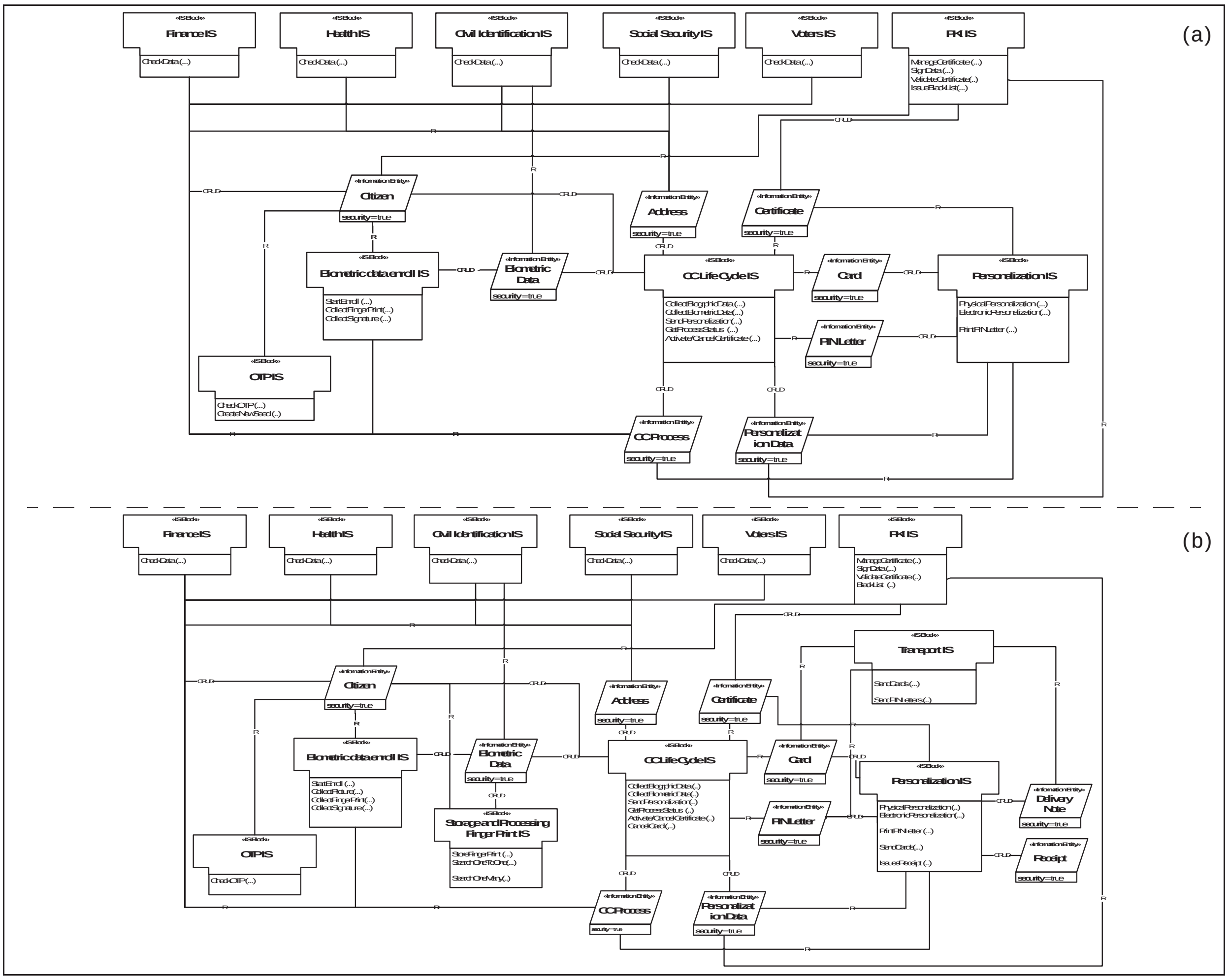

Figure 11: PoC (a) and final (b) Information and Application Architectures

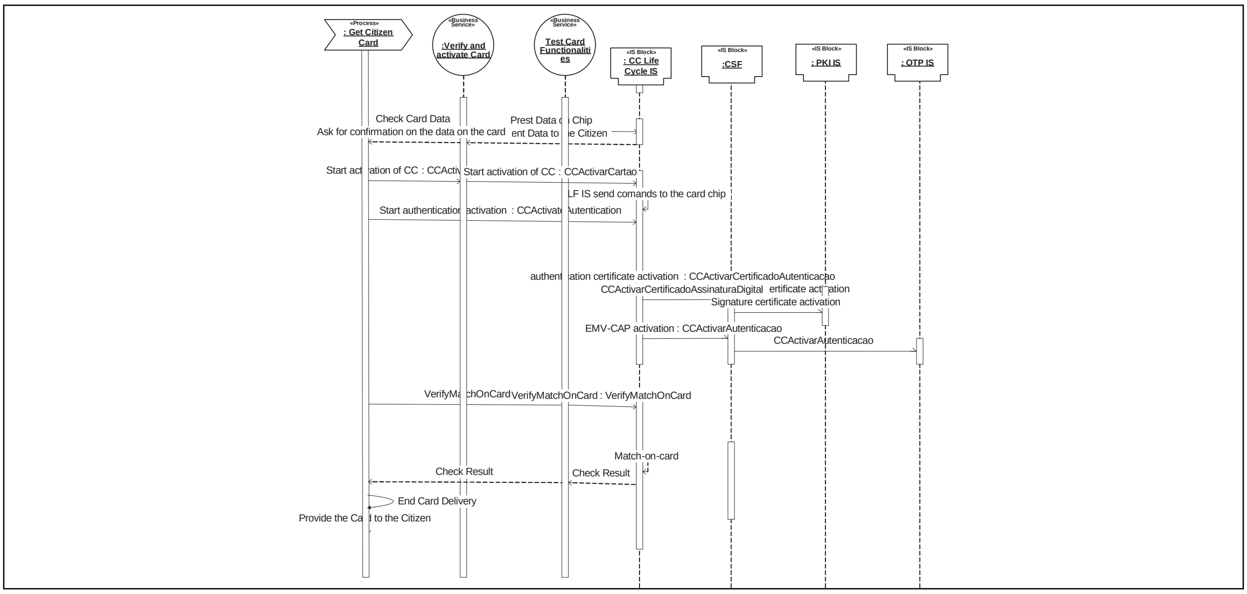

Figure 12: Sequence diagram for the Get Citizen Card Process of the final PoC

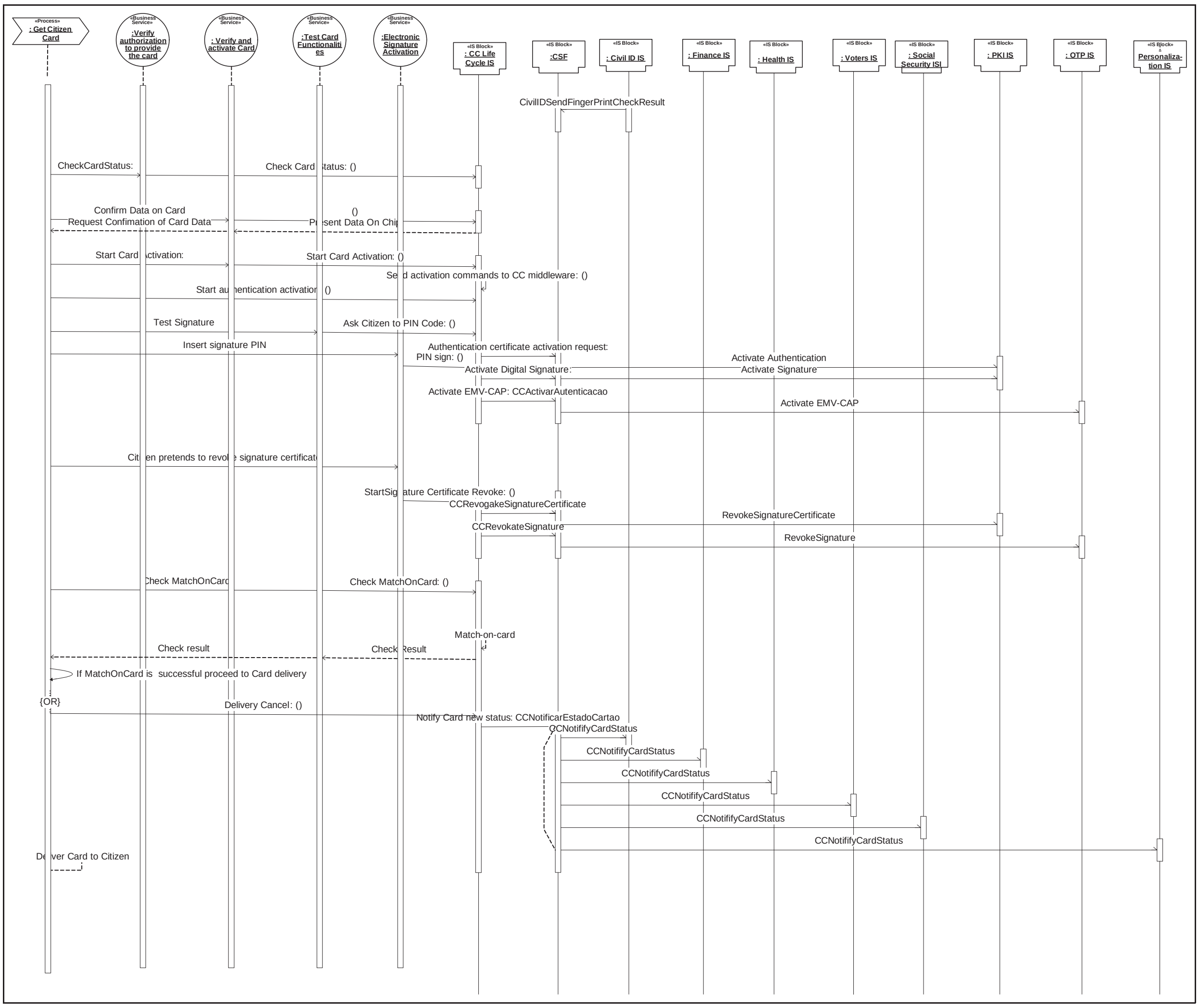

Figure 13: Sequence diagram for the Get Citizen Card Process of the final ISA

4.3.4 Technology Architecture

Considering the application components (defined in the application architecture previous described) the software components used for its implementation («IT Application Block») were then defined – Figure 14 presents the “CC Life Cycle IS” «IS Block» implementation at technology level for the PoC and the final ISAs.

The technical interoperability for both ISAs was assessed, considering all the dependencies between the application and technology architectures, using the DTISSF (Distinct Technologies for IS Services Factor) metric.

$$DTISSF(PoC) = 1 - \frac{\#«IS\ Service»}{\sum_{i=1}^{\#«IS\ Service»} \#«IT\ Service»_{Integration\ i}} = 1 - \frac{13}{13} = 0$$

$$DTISSF(Final) = 1 - \frac{\#«IS\ Service»}{\sum_{i=1}^{\#«IS\ Service»} \#«IT\ Service»_{Integration\ i}} = 1 - \frac{18}{20} = \frac{1}{10} = 0{,}1$$

Most of the «IS Services», in both architecture are implemented, at technological level, using webservices. This option ensures a technological independency allowing other systems (implemented in different technologies) to use the service. Nevertheless, in the final ISA, for example, the biometric data capture equipment provides, besides a webservice interface, another .Net interface - increasing the technical interoperability of the final architecture.

Figure 15 presents the PoC technology architecture and Figure 16 presents a partial view on the final technology architecture. The technological security of both architectures are estimated using the SCBITABF (Security Components Between «IT Application Block» Factor) metric.

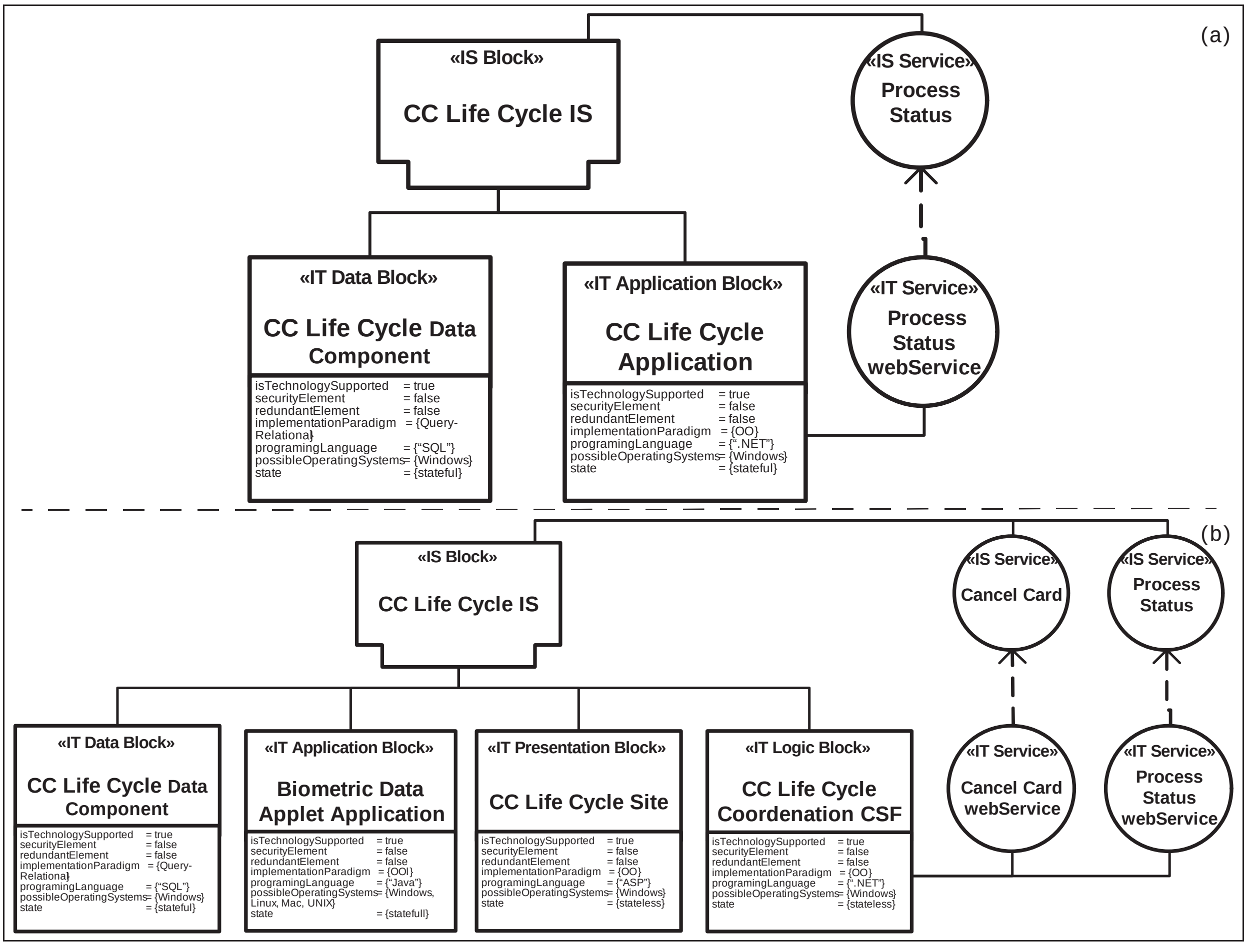

Figure 14: Dependencies between application and technological components (CC Life Cycle IS) for the PoC (a) and final (b) ISA

$$SCBITABF(PoC) = \frac{\sum_{i=1}^{\#IT\ Application\ Block}\left[\sum_{j=1}^{\#«IT\ Application\ Block»} Min\ \{\#SITB_{ij}\}\right]}{\#«IT\ Application\ Block» \times \#«ITBlock»} = \frac{702}{27 \times 82} = 0{,}317$$

$$SCBITABF(Final) = \frac{\sum_{i=1}^{\#IT\ Application\ Block}\left[\sum_{j=1}^{\#«IT\ Application\ Block»} Min\ \{\#SITB_{ij}\}\right]}{\#«IT\ Application\ Block» \times \#«ITBlock»} = \frac{37054}{108 \times 448} = 0{,}766$$

Therefore the (technological) security in the final ISA is considerable higher than in the PoC ISA.

The reliability of the architectures is estimated using the ITRF (IT Redundancy Factor) metric. Considering that in the PoC only two «IT Block» present redundant characteristics, the ITRF value is:

$$ITRF(PoC) = \frac{\#RITB}{\#«IT\ Block»} = \frac{2}{82} = 0{,}024$$

In the final ISA, since most of the servers have redundant characteristics the ITRF value is:

$$ITRF(Final) = \frac{\#RITB}{\#«IT\ Block»} = \frac{81}{448} = 0{,}181$$

The architectures efficiency was estimated using the SITPLBF (Stateful «IT Presentation Block» and «IT Logic Block» Factor) metric. However, since the presentation and logic components of both architectures do not keep the state (the state is only managed by the data components), both architectures present maximum values for this metric.

$$\text{SITPLBF}\ (PoC) = 1 - \frac{\#SITPLB}{\#«IT\ PresentationBlock» + \#«IT\ LogicBlock»} = 1 - \frac{0}{1} = 1$$

$$\text{SITPLBF}\ (Final) = 1 - \frac{\#SITPLB}{\#«IT\ PresentationBlock» + \#«IT\ LogicBlock»} = 1 - \frac{0}{8} = 1$$

The portability of each ISAs was estimated using the *POSF* (Possible Operating Systems Factor) metric.

$$POSF(PoC) = 1 - \frac{\#«IT\ Application\ Block»}{\sum_{i=1}^{\#«IT\ Application\ Block»} NPOS_i} = 1 - \frac{25}{119} = 0{,}790$$

$$POSF(Final) = 1 - \frac{\#«IT\ Application\ Block»}{\sum_{i=1}^{\#«IT\ Application\ Block»} NPOS_i} = 1 - \frac{51}{275} = 0{,}815$$

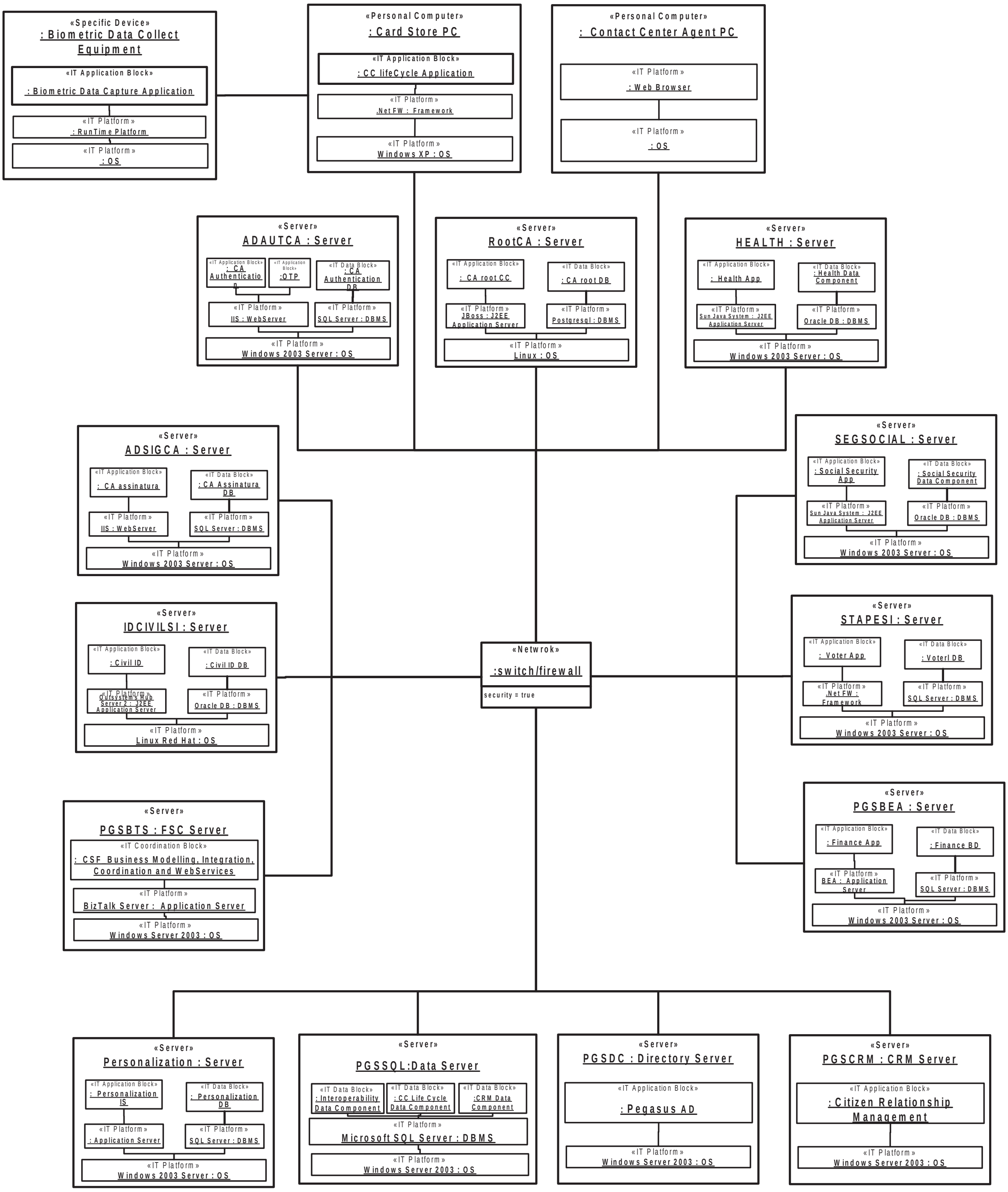

Figure 15: PoC Technology architecture

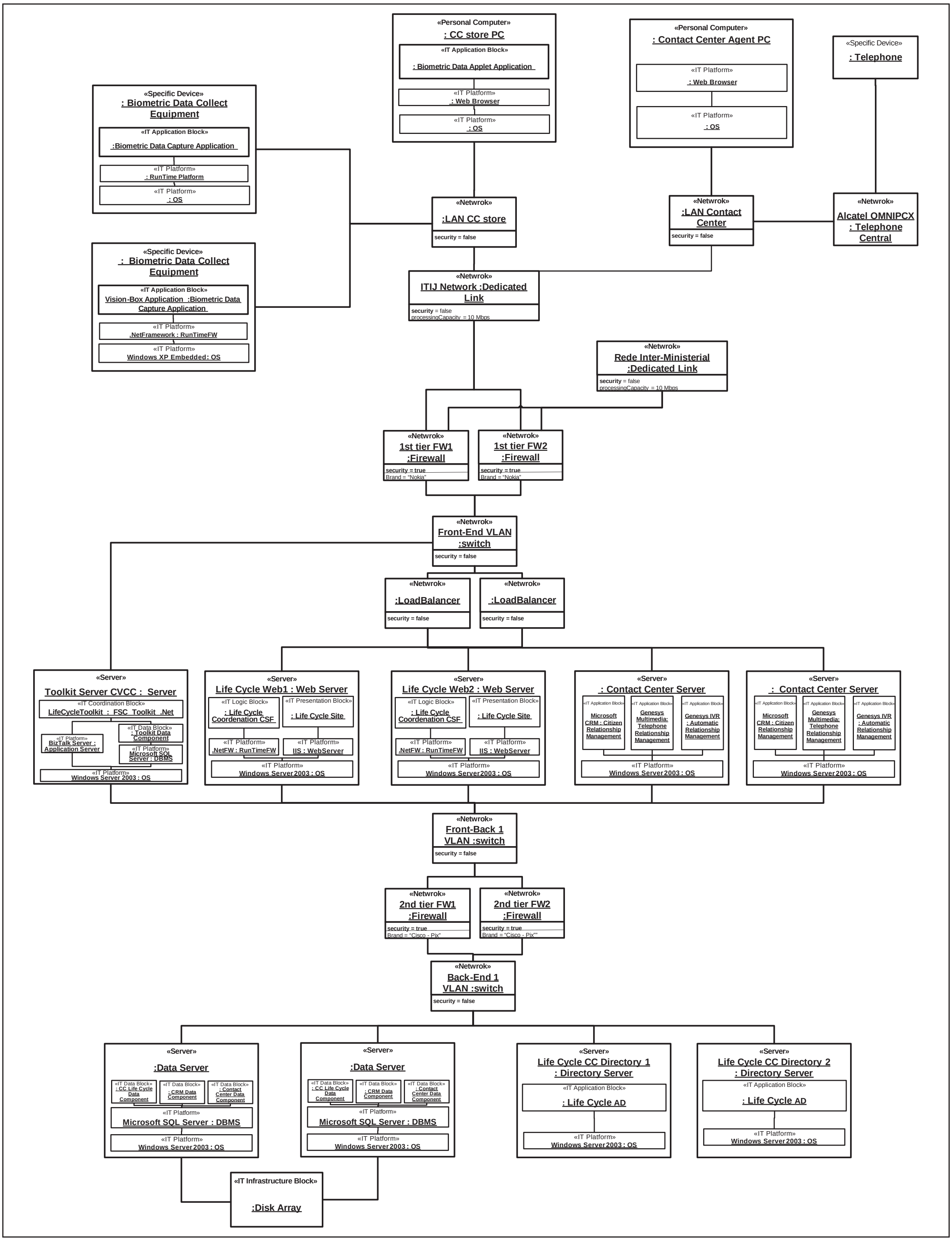

Figure 16: Final Technology architecture (partial view)

In order to estimate the information-technology alignment the *LLIEITBDTMF* (Low Level Information Entity – IT Block Data Type Mismatch Factor) metric was computed.

$$\text{LLIEITBDTMF (PoC)} = 1 - \frac{\#\{LowLevelInformationEntity_P \in ITBlock_{NP}\}}{\#«LowLevelInformationEntity»} + \frac{\#\{LowLevelInformationEntity_D \in ITBlock_{ND}\}}{\#«LowLevelInformationEntity»} \Leftrightarrow$$

$$\text{LLIEITBDTMF (PoC)} = 1 - \frac{0}{12} = 1$$

$$\text{LLIEITBDTMF (final)} = 1 - \frac{\#\{LowLevelInformationEntity_P \in ITBlock_{NP}\}}{\#«LowLevelInformationEntity»} + \frac{\#\{LowLevelInformationEntity_D \in ITBlock_{ND}\}}{\#«LowLevelInformationEntity»} \Leftrightarrow$$

$$\text{LLIEITBDTMF (final)} = 1 - \frac{17}{24} = 0{,}29$$

Since in the PoC no data was kept for statistics, or auditing, no low level information entities for managing derived data existed (like the consecutive addresses or consecutive cards for each citizen). As this was mandatory in the final ISA – and these derived entities are managed by the same applications responsible for managing the primitive entities – a misalignment between the information and technology levels of the final ISA exists.

Finally, the application-technology alignment was computed using the *CSTMF* (Critical System - Technology Mismatch Factor) metric. However, since all the processes in this project were considered critical (in both ISAs) the CSTMF presents a maximum value for both ISAs.

$$CSTMF\ (PoC) = 1 - \frac{\#\{ISBlock_C \in ITBlock_{NC}\} + \#\{ISBlock_{NC} \in ITBlock_C\}}{\#«ISBlock»} \Leftrightarrow$$

$$CSTMF\ (PoC) = 1 - \frac{0}{10} = 1$$

$$CSTMF\ (Final) = 1 - \frac{\#\{ISBlock_C \in ITBlock_{NC}\} + \#\{ISBlock_{NC} \in ITBlock_C\}}{\#«ISBlock»} \Leftrightarrow$$

$$CSTMF\ (Final) = 1 - \frac{0}{12} = 1$$

4.3.5 Alternative ISAs Evaluation

In order to perform a cost-benefit analysis on the PoC and final ISAs, the cost for each architectural level was computed. Thus, at information level the NE (Number of Entities) metric values are: *NE (PoC)* = 8, *NE (final)* = 10; for the application architecture the NA (Number of Applications) metric values are: *NA (PoC)*=10, *NA (final)*=12; and for the technology architecture the NITB (Number of IT Blocks) metric values are: *NITB (PoC)*=75, *NITB (final)*=448.

Combining the cost metrics with the benefit estimation metrics (computed in previous subsections) a cost-benefit analysis can be performed – in Figure 17 this analysis is carried out for the application architecture. The final application architecture is estimated to have higher quality characteristics than the PoC, except regarding changeability (at application level).

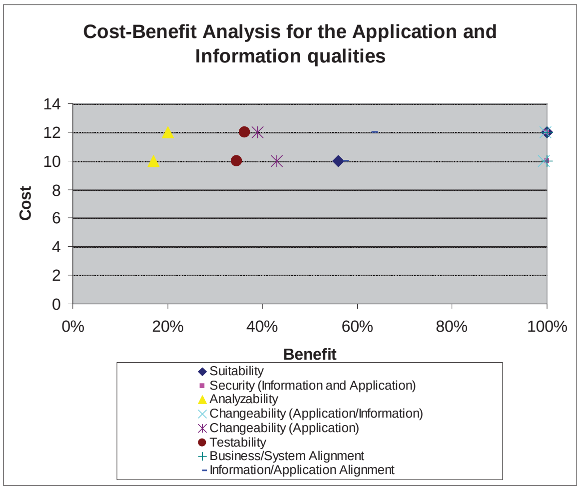

Figure 17: Application architecture Cost-Benefit analysis

The technological qualities characteristics are also compared in Figure 18. The final technological architecture qualities are estimated to be superior to the PoC ones, excepted for the information/technology alignment (as explained in previous subsection).

Considering the influence of each metric on each quality (previously defined in Table 2), the architectural qualities, for each ISA, are compared in Figure 19. The Functionality, Reliability, Maintainability and Portability are higher in the final ISA than in the PoC. The Efficiency, according to the metrics, is similar in both ISAs. Finally, the Alignment is estimated to be superior on the PoC ISA, since (as discussed in the previous subsection) no derived information entities were considered on the PoC ISA – however for the production (final) system this was a mandatory requirement (namely for process security and statistical reasons). Thus, the separation in different technological components, for the final ISA, is being considered for improving the final ISA quality.

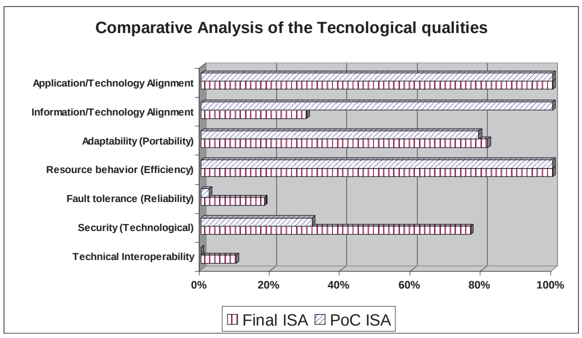

Figure 18: Technological qualities comparative analysis

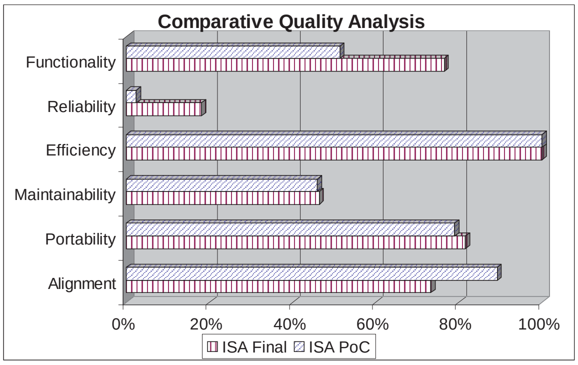

Figure 19: Comparative Analysis of PoC and Final ISAs' qualities

Considering for each quality its relative weight in the global ISA quality, in Table 3 is assessed the total quality of the final and PoC ISAs. Thus, the final ISA presents a total quality superior than the PoC ISA (of 65% and 59%, respectively).

Figure 20 presents a cost-benefit analysis of the top qualities (the cost was normalized in the scale [0;1] considering similar weights for each architectural level – information, application and technology).

As a final remark on this case study, it is important to emphasis that this approach, besides demonstrating that the final ISA quality is higher than the PoC architecture (something that was expected by the project team), pointed out some improvement opportunities for the final ISA. Namely, this evaluation revealed that the final ISA could improve its semantic interoperability and information-technology alignment.

The semantic interoperability improvements would imply some normalization on the information entities formats of the different public agencies involved (e.g., the address format is different in most of the existing systems). In order to improve the information-technology alignment the primitive and derived data should be managed by different IT Blocks (e.g., separation in different IT components of the change of address process and the previous citizen addresses).

Table 3: Global evaluation table for the Final and PoC architectures

	Metric	Quality Weight		ISA PoC	ISA final
Functionality		**26%**		**51.4%**	**76.5%**
Suitability	BSRPF	36%		56%	100%
Semantic Interoperability	DIIEF	12%		67%	42%
Technical Interoperability	DTISSF	12%		0%	10%
Security		40%		58%	86%
Technological	SCBITABF		62%	32%	77%
Information Application	IASF		38%	100%	100%
Reliability	ITRF	**15%**		**2.4%**	**18.1%**
Efficiency	SITPLBF	**12%**		**100.0%**	**100.0%**
Maintainability		**18%**		**45.9%**	**46.4%**
Analyzability	SCCF	17%		17%	20%
Changeability		39%		71%	69%
Information Application	LCOISF		50%	99%	100%
Application	NOISF		50%	43%	39%
Testability	RSF	44%		35%	36%
Portability	POSF	**9%**		**79.0%**	**81.5%**
Alignment		**21%**		**89.3%**	**73.3%**
Business/System Alignment	CPSMF	25%		100%	100%
Information/Application Alignment	NAIEF	25%		57%	63%
Information/Technology Alignment	LLIEITBDTMF	25%		100%	30%
Application/Technology Alignment	CSTMF	25%		100%	100%
Total (ISA Quality)				**59%**	**65%**

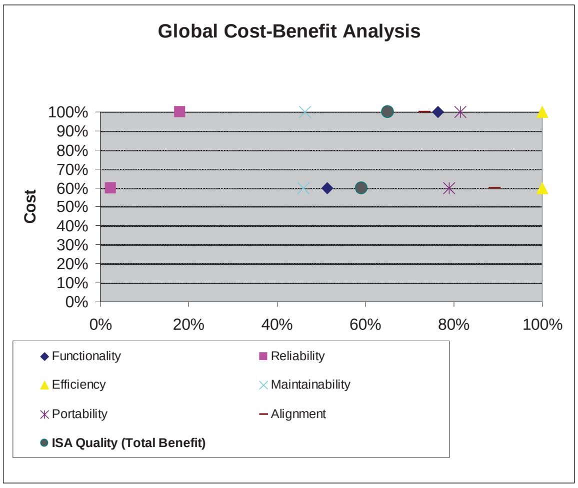

Figure 20: Global Cost-Benefit analysis

5 Conclusion and Future Work

This paper proposes a set of 16 metrics to assess 6 ISs' qualities (adapted and extended from the ISO9126 software quality standard), namely: Functionality, Reliability, Efficiency, Maintainability, Portability, and Alignment. The metrics, supported in the CEOF UML profile, are also formal defined in OCL (OCL description is not presented in the paper for space limitations), avoiding any interpretation or application ambiguities.

The results support that: 1) the metrics are applicable since the early stages of the ISA definition; 2) the metrics proposed are able to correctly order the ISA, according to the selected qualities; 3) the metrics are independent of the ISA size - for example, in the Citizen Card project the PoC and the final architectures have very dissimilar sizes (namely the technology architectures) but it was possible to compare the different architecture qualities (without being influenced by the ISA dimension); 4) the metrics do not consider the possible effort of changing an existing reality – they just assess the "best" ISA, independently of the existing ISs – for example in the Citizen Card project the metric DIIEF (Different Implementations of Information Entity Factor) revealed this fact.

Using the CEOF architectural primitives and metrics it is described an exploratory approach for supporting the ISA definition process. This approach, combines Spewak EAP methodology [SpHi92], with the CEOF primitives and defines clear evaluation tasks, using the metrics proposed.

Currently we are performing further experimental research, in order to verify the independence between the metrics described. Another future research topic is the experimental verification of the correlation between the metrics and the estimated ISA qualities.

Improving the metrics in order to consider/assess other ISA patterns ("best practices") in the metrics formulas', is another planned future research path. The approach for constructing an ISA should also be applied to several more projects and organizations in order to validate and generalize it.

Finally, another projected work is the improvement of the existing software tools for ISA modelling and assessment. We plan to integrate ISA monitoring activities (using the CEOF primitives and metrics) on the regular organizational tasks, by providing software monitoring tools for determining the current ISA (in real time) at technology, application and information levels.

6 Acknowledgements

The case study research presented in this paper was possible thanks to the support of AMA – Agency for the Public Services Reform.

References

[AbCa94] Abreu, F.; Carapuca, R.; Candidate metrics for object-oriented software within a taxonomy framework, J. Syst. and Software 23, 87-96, 1994.

[BaBM96] Basili, V. R.; Briand, L. C.; Melo, W. L.; A validation of object-oriented design metrics as quality indicators; In: IEEE Trans. Software Eng. 22, 1996.

[BEA06] BEA; Scaling EJB Applications; available in http://edocs.bea.com/wle/tuning/tsejb.htm, May 2006.

[Boar99] Boar, B.: Constructing Blueprints for Enterprise IT Architecture, John Wiley & Sons, 1999.

[ChKe94] Chidamber, S.; Kemerer, C.; A metrics suite for object-oriented design. IEEE Trans. on Software Engineering, 20(6):476–493, 1994.

[DeMa82] De Marco, T.; Controlling software projects, Yourdon Press, New York, 1982.

[ErPe00] Eriksson, H.; Penker, M.: Business Modeling with UML: Business Patterns at Work, John Wiley & Sons, ISBN 0-471-29551-5, 2000.

[Hend96] Henderson-Sellers, B.; Object-Oriented Metrics – Measures of Complexity, The Object-Oriented Series, Prentice Hall PTR, EUA, ISBN 0-13-239872, 1996.

[IDAB03] IDABC; The European Interoperability Framework - for improved pan-European government e-services; Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens, June 2003, available in http://ec.europa.eu/idabc/en/document/1439

[Inmo00] Inmon, W.H.; "Using The Generic Data Model", White Paper, 2000, http://www.inmoncif.com/library/whitepapers/

[Inmo92] Inmon, W. H; Data architecture: the information paradigm; QED Information Sciences, Inc. Wellesley, MA, USA, 1992.

[ISO01] ISO 9126, "Information Technology – Software Product Evaluation – Software Quality Characteristics and Metrics", International Organization for Standardization, 2001.

[KhHo04] Khaddaj, S.; Horgan, G.; The Evaluation of Software Quality Factors in Very Large Information Systems; In: The Electronic Journal of Information System Evaluation, Vol.7, Issue 1, 2004.

[McCa76] McCabe, T.J.; A complexity measure, IEEE Transactions on Software Engineering, SE-2, Volume 4, 1976, pp. 308-320.

[MCL+99] Malone, T.; Crowston, K.; Lee J.; et al.,: Tools for inventing organizations: Towards a handbook of organizational processes, Management Science, 1999.

[MRTG00] Maes, R.; Rijsenbrij, D.; Truijens, O.; Goedvolk, H.; Redefining Business – IT Alignment Through a Unified Framework, White Paper, May 2000, http://www.cs.vu.nl/~daan/

[OMG06a] OMG, "Unified Modeling Language: Superstructure, version 2.1", Object Management Group, 2006.

[OMG06b] OMG, "Object Constraint Language: OMG Available Specification Version 2.0", Object Management Group, 2006.

[OpGr03] Open Group; The Open Group Architectural Framework (TOGAF) – Version 8.1, chapter 19, December 2003.

[SaSu03] Sarkis, J.; Sundarraj, R.; Evaluating Componentized Enterprise Information Technologies: A Multiattribute Modeling Approach, Information Systems Frontiers, Kluwer Academic Publishers, 2003, section 3.2.1, pp. 303–319.

[SoPM04] Sousa. P.; Pereira, C.; Marques, J.; Enterprise Architecture Alignment Heuristics, Journal4, Microsoft Architects Journal , October 2004.

[SpHi92] Spewak, S.; Hill, S.; Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology, Wiley-QED, ISBN 0-471-599859, 1992.

[STV+06] Sousa, P,; Tribolet, J.; Vasconcelos, A.; Caetano, A.; Pereira, C.; Enterprise Modeling and Computing with UML, chapter of book "Enterprise Modeling and Computing with UML" edited by Peter Rittgen, University College Borås, Sweden, ISBN: 1-59904-175-8, 2006

[Varg00] Vargas, H.; High Availability Fundamentals, Sun BluePrints OnLine, November 2000, available in http://www.sun.com/blueprints/1100/HAFund.pdf

[Vasc06] Vasconcelos, A.; CEO Framework for Information System Architecture: An UML profile, CEO Report, available at http://ceo.inesc.pt/andre/reports/CEOF_UML_Profile_v1_2.pdf

[VaST03] Vasconcelos, A.; Sousa, P.; Tribolet, J.; Information System Architectures; In: Proceedings of Business Excelence 2003, International Conference on Performance Measures, Benchmarking and Best Practices in New Economy, Portugal, June 2003.

[VaST05] Vasconcelos, A.; Sousa, P., Tribolet, J.; Information System Architecture Evaluation: From Software to Enterprise Level Approaches; In: 12th European Conference On Information Technology Evaluation (ECITE 2005), Turku, Finland, September 2005.

[VCN+01] Vasconcelos, A.; Caetano, A,; Neves, J.; Sinogas, P.; Mendes, R; Tribolet, J.: A Framework for Modeling Strategy, Business Processes and Information Systems; In: 5th International Enterprise Distributed Object Computing Conference EDOC, Seatle, USA, September 2001.

[VSFT04] Vasconcelos, A.; Silva,M; Fernandes, A.; Tribolet., J.; An Information System Architectural Framework for Enterprise Application Integration; In: Proceedings of 37th Annual Hawaii International Conference On System Sciences (HICCS37), Hawaii, USA, 2004.

[Zach97] Zachman, J.: Enterprise Architecture: The Issue of the Century, Database Programming and Design, March 1997, pp. 11.

[Zach99] Zachman, J.: Enterprise Architecture: The Past and the Future". In: DM Review Magazine, December 1999.

André Vasconcelos, Pedro Sousa, José Tribolet

Instituto Superior Técnico, Technical University of Lisbon
CEO - Centro de Engenharia Organizacional - INESC
Rua Alves Redol 9
Lisbon
Portugal
{andre.vasconcelos|pedro.sousa|jose.tribolet}@dei.ist.utl.pt