Information System Architecture Metrics: an Enterprise Engineering Evaluation Approach

André Vasconcelos, Pedro Sousa and José Tribolet
Technical University of Lisbon, Portugal
andre.vasconcelos@dei.ist.utl.pt
pedro.sousa@dei.ist.utl.pt
jose.tribolet@dei.ist.utl.pt

Abstract: Although some important technological developments have been achieved during last decade, information systems still do not answer efficiently enough to the continuous demands that organisations are facing – causing a non-alignment between business and information technologies (IT) and therefore reducing organisation competitive abilities. This paper proposes sixteen metrics for the Information System Architecture (ISA) evaluation, supported in an ISA modelling framework. The major goal of the metrics proposed is to assist the architect previewing the impact of his/her ISA design choices on the non-functional qualities of the Enterprise Information System (EIS), ensuring EIS better align with business needs. The metrics proposed are based on the research accomplished by other authors, from the knowledge in other more mature areas and on the authors experience on real world ISA evaluation projects. The metrics proposed are applied to an e-government project in order to support the definition of a suitable ISA for a set of business and technological requirements.

Keywords: Information system architecture metrics, information system architecture evaluation, enterprise information system, ceo framework, e-government project evaluation.

1. Introduction

Though Information System Architecture (ISA) is currently recognised as an essential step in the process of building Enterprise Information Systems (EIS) aligned with business needs, there are not tools that assist the Information System (IS) architect in accessing (during "design time") the impact of his or her decisions on the global ISA qualities. Moreover, other ISA stakeholders that might have limited knowledge on ISA matters (as business people, software engineers, infra-structure experts) do not have simple methods or tools to quickly and automatically evaluate an ISA in respect to a set of desired IS qualities driven from the business context. The authors' research pretends to provide ISA stakeholders the tools for assessing ISA qualities ensuring EIS suitable to business needs. Firstly, recognising the need for a coherent way of representing ISA, in Vasconcelos et al. (2001), the authors proposed a set of Enterprise modelling primitives (the CEO Framework), extended later into an UML profile for ISA modelling – regarding information, application and technological information system concerns (Vasconcelos et al. 2003). Afterward the ISA modelling framework have been tested in real world case studies (Vasconcelos et al. 2004a) and enriched considering other IS characteristics (Vasconcelos et al. 2004b) – this research step confirmed the need for tools capable of supporting the architect while building the ISA and quickly accessing his or her design choices. More recently, considering that the evaluation topic is a quite mature issue on the software engineering domain, the authors classify several software evaluation approaches in order to consider its applicability for ISA evaluation and adapted some software metrics to the information system context (Vasconcelos et al. 2005). In this paper the authors present theirs recently developments on ISA evaluation by proposing and explaining the foundation of a set of metrics for ISA evaluation. The ISA modelling framework that supports the evaluation metrics is introduced in section 2. In section 3 the authors proposed a coherent set of ISA evaluation metrics, relating ISA qualities and ISA components. In section 4 the metrics proposed are applied to an e-government ISA project. The conclusions and future work are presented in section 5.

2. CEO framework for ISA modelling

In section 2.1 we introduce the Information System Architecture (ISA) major concepts and relations. In section 2.2 we introduce the CEO framework for ISA modelling; this framework is used in section 3 for supporting the metrics proposed for ISA evaluation.

2.1 Information system architecture

Information System Architecture (ISA) is a part of a vaster field of architectures and models relevant for the organisation. Considering the architectural level, one can distinguish the following architectures:

- Enterprise Architecture.
- Information System Architecture (ISA).
- Software Architecture (SWA)

Reference this paper as:
Vasconcelos A, Sousa P and Tribolet J (2007) "Information System Architecture Metrics: an Enterprise Engineering Evaluation Approach" *The Electronic Journal Information Systems Evaluation* Volume 10 Issue 1, pp 91 - 122 , available online at www.ejise.com

Software Architecture (SWA) main study area is on how programs or application components are internally built (Carnegie 2000). At this level it is import to considered the objects and classes needed for implementing the software. SWA is a quite stable and mature field. Enterprise Architecture is a group of models defined for getting a coherent and comprehensible picture of the enterprise (Tissot et. al. 1998). The models define different “perspectives or viewpoints from which the company is considered, focusing on some aspects and ignoring others in order to reduce complexity” (Vernadat 1996). Thus, a model of the company can contain several activity, process, organisation, information and behaviour diagrams of the company. Enterprise architecture is considered a vaster concept than ISA, which includes business strategies and processes, besides Information System (IS) models that support them. Usually, at enterprise architecture level, IS are consider “simple” resources used in business (as people, equipment and material, etc.) – e.g., (Eriksson et. al. 2000) and (Marshall 2000).

Finally, Information System Architecture (ISA) addresses the representation of the IS components structure, its relationships, principles and directives (Garlan et. al. 1995), with the main propose of supporting business (Maes et. al. 2000). Spewak in Spewak et. al. (1992), argues that the ISA description is a key step in ensuring that IT provides access to data when, where and how is required at business level. Thus some of the potential benefits of an ISA are:

- IS complexity and interfaces cost reduction (Cook 1996), Spewak et. al. (1992).
- Ensures IS flexible, durable and business oriented (Zijden et. al. 2000) .
- Allows the evolution and introduction of new technologies according to the strategy of the business plan (Cook 1996), (Spewak et. al. 1992).
- Provides the means for business, IS and IT components alignment (Zijden et. al. 2000) .
- Ensures greater efficiency using IT, namely by providing: a controlled development and maintenance cost, more application portability, and more flexibility in changing and upgrading technological components (Open 2003).

In the 80’s, software architecture (SWA) and ISA where considered synonymous. Only in last decade the need for manipulation of concepts that overwhelm the description of how a system is internally built emerged. Zachman framework (Zachman 1987), is defined has the first important sign that SWA has not enough.

Quoting IEEE (1998), ISA level should be high. Thus, ISA is distinguished from software representation and analysis methods (as E-R diagrams, DFD), presenting an abstraction of internal system details and supporting organisation business processes (Zijden et. al. 2000) .Sassoon, discusses the concept of “IS urbanisation”, emphasising, like in city planning, the need for models that guide the evolution and growth of IS robust and independent of technological trends (Sassoon 1998).

ISA usually distinguish three aspects, defining three “sub architectures” Spewak et. al. (1992):

- Informational Architecture, or Data Architecture. This level represents the main data types that support business.
- Application Architecture. Application architecture defines applications needed for data management and business support.
- Technological Architecture. This architecture represents the main technologies used in application implementation and the infrastructures that provide an environment for IS deployment.

Identifying and defining the major data types that support business development is Informational Architecture major propose Spewak et. al. (1992), DeBoever (1997). Inmon (2000) characterises data (the support of the information architecture) through different dimensions: primitive vs. derived, private vs. publics and historical vs. operational vs. provisional data. Inmon (2000) argues that the ISA should be influence by the data characteristics. The second architecture level, defined by DeBoever (1997), is the application (or system) architecture. This architecture defines the main applications needed for data management and business support. This architecture should not be a definition of the software used to implement systems. The functional definition of the applications that should ensure access to data in acceptable time, format and cost is this architecture main focus Spewak et. al. (1992).Application architecture defines the major functional components of the architecture. Spweak proposes a methodology – Enterprise Architecture Planning (EAP) – able to define application architecture from informational and business requirements Spewak et. al. (1992). Using Spewak methodology and Zachman framework several institutions have been proposing adaptations that best answer to its needs – interesting case studies are Information System Architectures in the American Federal Government (FEAF 1999), DoD Technical Reference Model (DOD 2002), Treasury Enterprise Architecture Framework (Business 2002), among others.

Technological architecture defines the major technologies that provide an environment for application building and deployment. At this level, the major technological concepts relevant for the IS are identified – as network, communication, distributed computation, etc. Spewak et. al. (1992).

2.2 The CEO framework

The CEO Framework (Figure 1) aims at providing a formal way of describing business goals, processes, resources and information systems and the dependencies between them. It is composed of three separate levels, each of which provides adequate forms of representing the notions about the layer being described (Vasconcelos et al. 2001).

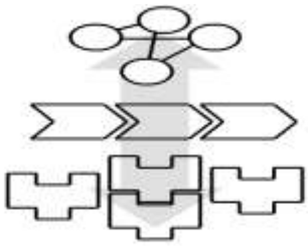


Figure 1. Goal / process / system framework

In the first level, the aim is at describing the current set of goals that drive business. These goals must be achieved through one or more business process. The business processes are described at the second level and must exist in order to satisfy one or more goals. Besides serving goals, business processes interact with resources in order to do work and may be supported by information systems. The information systems layer aims at modelling the components of the system that support business.
The modelling language used to implement the CEO Framework was UML (Unified Modelling Language). As UML was initially designed to describe aspects of a software system, it had to be extended to more clearly identify and visualise the important concepts of business, namely by use of stereotypes – for further detail on UML extension mechanisms see OMG (2004). Due to size restrictions, we will not do a full presentation on the CEO Framework (for further reading, refer to Vasconcelos et al. (2001)). **Figure 2** presents the UML metamodel defined for the CEO Framework.

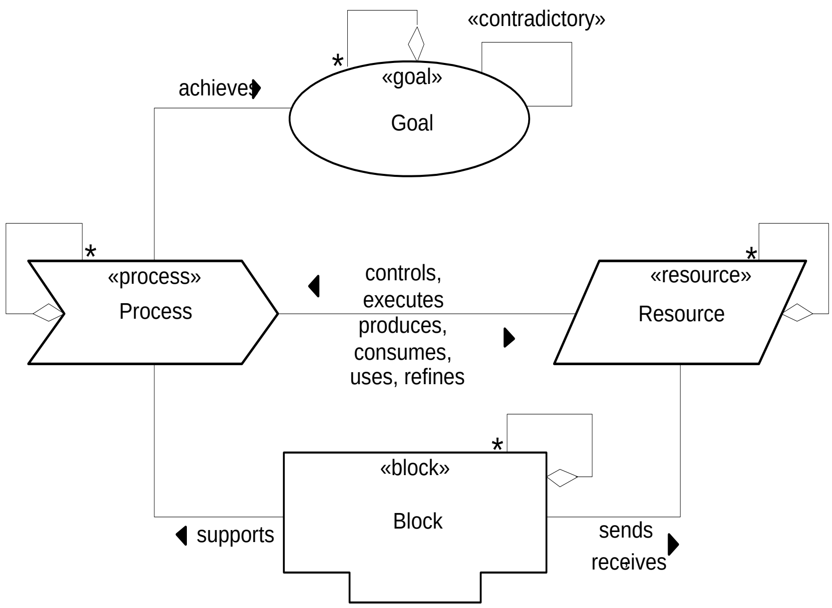


Figure 2 - UML metamodel of the CEO framework

In order to model ISA key concepts, the «Block» component was specialised. The key concepts for the Information System Architecture are:

- Information Entity – person, place, physical thing or concept that is relevant in the business context;
- IS Block – Application architecture main aim is on the functional components characterisation. At application level, the IS Block (or Application Block) notion is the founding concept. IS Block is defined as the collection of mechanisms and operations organised in order to manipulate organisation data.
- IT Block – Technological architecture addresses a large variety of notions, caused, on the one hand, by the continuous

technological evolutions and, on the other hand, by the need for different specialised IT architectural views – as security, hardware and software development architectures. In order to encapsulate this diversity, this framework uses the "IT Block" concept. IT Block is the infrastructure, application platform and technological/software component that realises (or implement) an (or several) IS Block(s). IT Block defines three major sub-concepts:

- IT Infrastructure Block – represents the physical and infra-structural concepts existing in an ISA: the computational nodes (as servers, personal computers or mobile devices) and the non-computational nodes (as printers, network, etc.) that support application platforms.
- IT Platform Block – stands for the implementation of the services used in the IT application deployment.
- IT Application Block, the technological implementation of an IS Block. At this level is relevant to consider the kind of IT Application Block (namely presentation, logic, data and coordination block), and its "technological principles" (like if it is implemented using components, modules, OO principles, etc.), among other characteristics.

- Service – is an aggregation of a set of operations provided by an architectural block. A generalisation of the web service notion (W3C 2002). We consider three distinct services in an ISA:
  - Business Service. A business service is a collection of operations provided by IS Blocks that support business processes.
  - IS Service. The set of operations provided by an IS Block to others IS Blocks defines the IS service.
  - IT Service. The technological services provided by application platforms are the IT services (Open 2003).
- Operation, the abstract description of an action supported by a service. Thus, operations are the minor level concept relevant in an ISA.

Figure 3 describes how these high-level primitives are related, in a UML profile for ISA. For further detail please refer to Vasconcelos et al (2003).

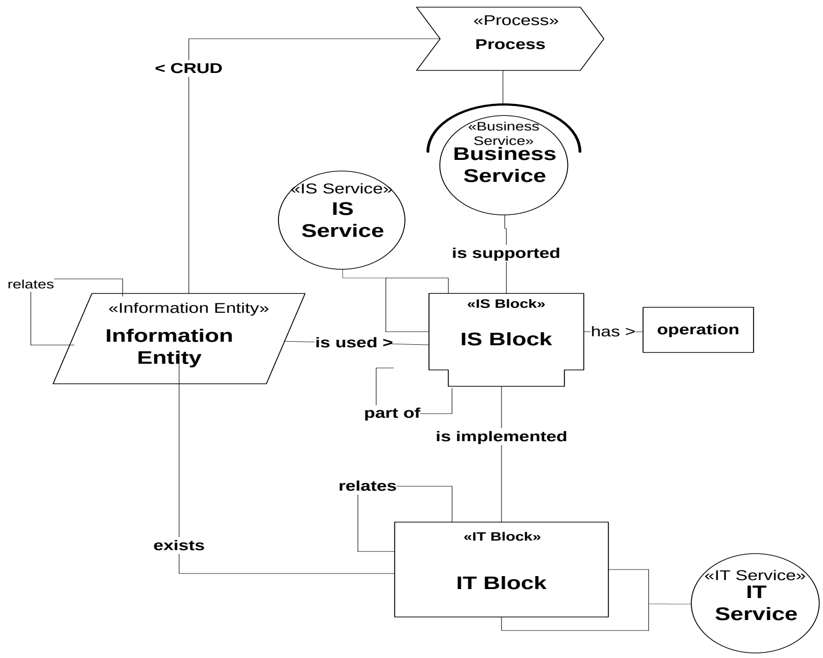


Figure 3. Information System CEO framework metamodel (Vasconcelos et al. 2003)

3. Information system architecture evaluation

In this section we propose a set of metrics for ISA evaluation. In section 3.1 we identify the information system qualities that might be measure in an ISA. In section 3.2 the authors propose a set of ISA evaluation metrics.

3.1 Information system qualities

As discussed in Vasconcelos et al. (2005), the qualities attributes that are important in software evaluation are also significant in ISA evaluation. The accuracy and suitability of an architecture is analysed considering several quality attributes. Bass (1998) and Clements (2002) propose the following:

- Usability – user's ability to utilise a system effectively;
- Performance – responsiveness of the system – the time required to respond to stimuli or the number of events processed in some interval of time;
- Reliability – ability of the system to keep operating over time;
- Availability – proportion of time the system is up and running;
- Security – system's ability to resist unauthorised attempts at usage and denial of service while still providing its services to legitimate users;
- Functionality – ability of the systems to do work for which it was intended;
- Modifiability – ability to make changes to a system quickly and cost effectively;
- Portability – ability of the system to run under different computing environments;
- Variability – how well the architecture can be expanded or modified to produce new architectures that differ in specific, preplanned ways;
- Subsetability – ability to support the production of a subset of the system;Testability – ability to observe results and control the components internal state in order to identify system faults;
- Conceptual Integrity – vision that unifies the design of the system at all levels (ability of the architecture do similar things in similar ways);
- Building simplicity – ability to implement the defined architecture;
- Cost – System Cost;
- Time to market – Time required to implement the architecture.

At Enterprise System Architecture level, other qualities become relevant, as Information System/Business alignment (ISA accurateness to the business needs), Information System/Strategy alignment (ISA support for the enterprise strategy) or Interoperability (ability of an ISA to interact or support different technologies – technical Interoperability – or support different information architecture implementations – syntactic Interoperability). These quality attributes might be: observable during execution (as usability, performance, reliability, availability, security and functionality), non-observable during executing (as Modifiability, portability, variability, subsetability and testability), architectural quality attributes (as conceptual integrity and building simplicity) or business quality attributes (as cost and time to market). The qualities are interrelated, and enhancing one will likely degrade or enhance others – for instance performance is likely to degrade scalability – for further detail on this subject please refer to Gillies (1992) or Khaddaj and Horgan (2004).

3.2 Information system evaluation metrics

In this section the authors propose a set of ISA evaluation metrics. These metrics were defined based on the research accomplished by other authors (specialists in certain areas – e.g., security, scalability, portability), on the adaptation of the evaluation knowledge from other more mature areas (e.g., software engineering) and on the authors experience on real world ISA evaluation projects. The authors argue that with these metrics the architect has a set of indicators on the impact of each of his or her decisions during the process of building an ISA and, therefore, he or she will be better equipped to build EIS align with a set of desired qualities. The following template is used to describe the metrics proposed.

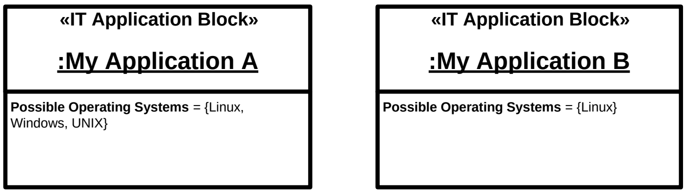
Table 1. ISA metric template

Acronym	Metric Acronym
Name	Metric Name
Computation	Description on the metric algorithm or formula
Scale	Scale of possible values for the metric
Architectural Levels	Architecture levels relevant for this metric
ISA Primitives and attributes	Architectural primitives and attributes used in the metric computation
ISA Qualities	Enumeration of the "architectural qualities" related with the metric
Support	Rational that supports the metric proposed and its relevance for measuring the ISA qualities
Example(s)	Presentation of ISA evaluation simple examples by applying the proposed metric

Acronym	*NPOS* (or $NPOS_{ISA}$)
Name	*Average Number of Possible Operating Systems*
Computation	The Average Number of Possible Operating Systems is computed by counting, on each application (*«IT Application Block»*), the number of possible operating systems (families) and dividing it by the number of applications $NPOS_{ISA} = \frac{\sum_{i=1}^{\#«IT\ Application\ Block»} NPOS_i}{\#«IT\ Application\ Block»}$, 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
Scale	$[1, +\infty[$
Architectural Levels	Technological Architecture
ISA Primitives and attributes	Primitive: *«IT Application Block»* - Attribute: *possible operating systems*
ISA Qualities	The Portability and Technical Interoperability of an EIS tend to increase with this metric
Support	The portability and Technical Interoperability in an ISA increase with the number of possible platforms where ISA components are able to operate (Sarkis and Sundarraj 2003, section 3.2.1). From a software engineering perspective, the portability of an operating system is a major indicator on an application portability (Roulo 1997); in the same way, the technical portability of an EIS, represented by an ISA, is measure by this metric as the average of the software applications' (*«IT Application Block»*) portability.

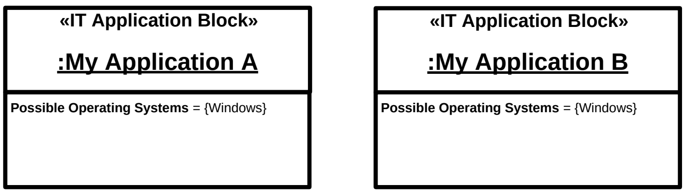
Example(s)

Example ISA A

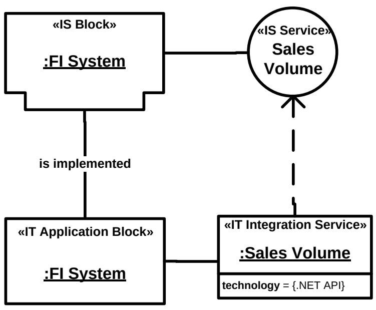
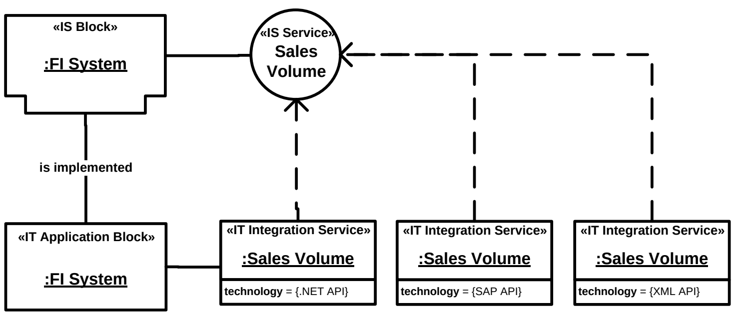


$$POS_{ISA} = \frac{3+1}{2} = 2$$

Example ISA B



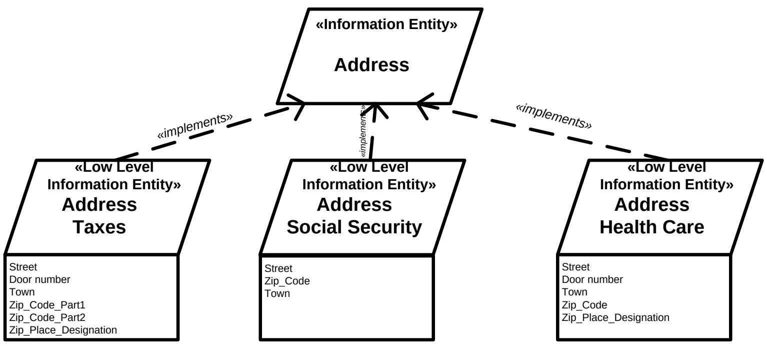
$$POS_{ISA} = \frac{1+1}{2} = 1$$

Acronym	*NDTIS* (or $NDTIS_{ISA}$)
Name	*Average Number of Distinct Technologies for IS Services*
Computation	The Average Number of Distinct Technologies for Information System Services is computed by counting for each *«IS Service»* the number of *«IT integration Services»*. $NDTIS_{ISA} = \frac{\sum_{i=1}^{\#«IS\ Service»} \#«IT\ Integration\ Service»_i}{\#«IS\ Service»}$, where: $\#«IT\ Integration\ Service»_i$ – is the number of *«IT Integration Service»* that implement the *«IS Service»* ***i*** *#«IS Service»* – is the number of *«IS Service»* in an ISA
Scale	$1;+\infty$
Architectural Levels	Application Architecture and Technological Architecture
ISA Primitives and attributes	Primitive:: *«IT Integration Service»* - Attribute: *technology*
ISA Qualities	The Portability and Technical Interoperability of an EIS tend to increase with this metric
Support	The technical interoperability of a software architecture increases by providing the same interface in different technologies (Sarkis and Sundarraj 2003, section 3.2.1). In the same way, with this metric the technical interoperability and portability of an EIS is analysed as the average of the Technologies that each application interface provides.
Example(s)	Example ISA A  $NDTIS_{ISA} = \frac{1}{1} = 1$ Example ISA B  $NDTIS_{ISA} = \frac{3}{1} = 3$

Acronym	*NIIE* (or $NIIE_{ISA}$)
Name	*Average Number of (Different) Implementations of an Information Entity*
Computation	The Average Number of (Different) Implementations of an Information Entity is computed by counting, for each *«Information Entity»*, the number of possible implementations in *«Low Level Information Entities»*. $NIIE_{ISA} = \frac{\sum_{i=1}^{\#«Information\ Entity»} NLLIE_i}{\#«Information\ Entity»}$, where: $NLLIE_i$ – is the number of *«Low Level Information Entities»* that are related to the *«Information Entity»$_i$* by the «*implements*» relation *#«Information Entity»* – is the number of *«Information Entities»* in an ISA
Scale	$1;+\infty$
Architectural Levels	Information Architecture
ISA Primitives and attributes	Primitive: *«Low Level Information Entity»* Primitive: *«Information Entity»*
ISA Qualities	The <u>Syntactic Interoperability</u> of an ISA will increase by the decrease of this metric
Support	This metric measures the number of different implementations that exist for each information entity. According to Inmon (2000), for each information entity ("top level") there might be other entities that implementing it ("low level information entity"). The existence of different *«Low Level Information Entities»* points to syntactic incompatibilities for that *«Information Entity»* (e.g., by using different formats or attributes in the implementation of the information entity).

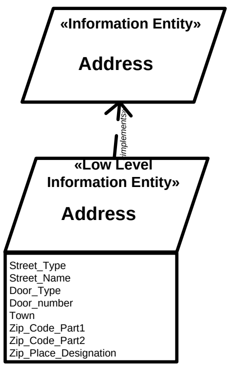
Example(s)

Example ISA A

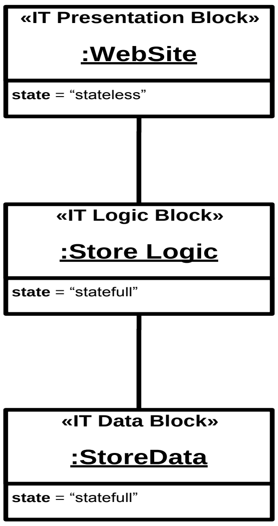
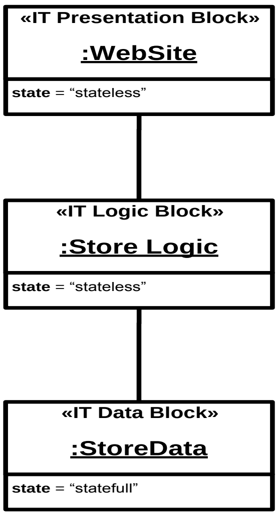


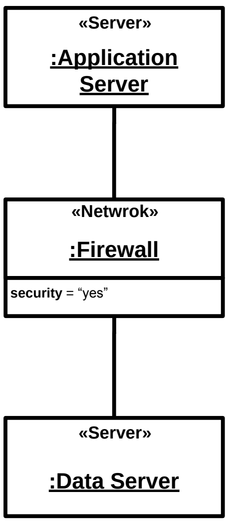
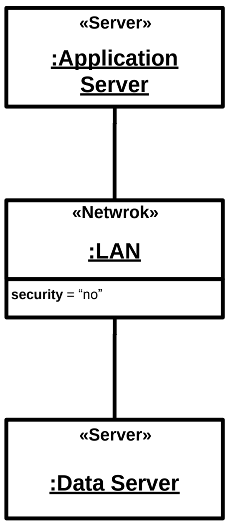
$$NIIE_{ISA} = \frac{3}{1} = 3$$

Example ISA B



$$NIIE_{ISA} = \frac{1}{1} = 1$$

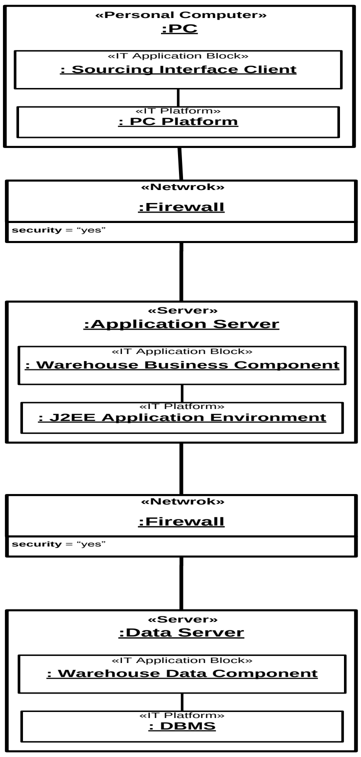
Acronym	*NSITPLB* (or $NSITPLB_{ISA}$)
Name	*Average Number of stateful «IT Presentation Block» and «IT Logic Block»*
Computation	The Average Number of stateful «IT Presentation Block» and «IT Logic Block» is computed counting the number of *«IT Presentation Block»* and *«IT Logic Block»* that its attribute *"state"* value is "*stateful*". $NSITPLB_{ISA} = \frac{\#SITPLB}{\#«IT\ PresentationBlock» + \#«IT\ LogicBlock»}$, 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 »*
Scale	[0; 1]
Architectural Levels	Technological Architecture
ISA Primitives and attributes	Primitives: *«IT Presentation Block», «IT Logic Block», «IT Data Block», «IT Coordination Block»* Attribute: *state*
ISA Qualities	The scalability of an ISA tends to increase with the decrease of this metric.
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) – BEA (2006). The Scalability of an ISA tend 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.
Example(s)	Example ISA A  $NSITPLB_{ISA} = \frac{1}{1+1} = \frac{1}{2}$ Example ISA B  $NSITPLB_{ISA} = \frac{0}{1+1} = 0$

Acronym	*NSC* (or NSC_{ISA})
Name	*Average Number of security components*
Computation	The Average Number of security components is computed counting all the *«IT Blocks»* which attribute "*security*" value is "*YES*". $NSC_{ISA} = \frac{\#SITB}{\#«IT\ Block»}$, where: *#SITB* – is the number of *«IT Blocks»* which attribute "security" value is "YES" *#«IT Block»* – is the number of *« IT Block »*
Scale	[0; 1]
Architectural Levels	Technological Architecture
ISA Primitives and attributes	Primitive: *«IT Block»* Attribute: *security*
ISA Qualities	The security of an ISA tends to increase with this metric increasing.
Support	The ISA security is increased by putting security elements on it, as IDS, firewalls, etc (Rito 2004). This metric considers this fact.
Observations	This simple to compute metric is provides a quick first overview on the potential (miss of) security of an ISA. However, this metric does not considers the role of the security components on the ISA.
Example(s)	Example ISA A  $NSC_{ISA} = \frac{1}{3}$ Example ISA B  $NSC_{ISA} = \frac{0}{3} = 0$

Acronym	*NSCBITAB* (or *$NSCBITAB_{ISA}$*)
Name	*Average Number of security components between «IT Application Blocks»*
Computation	The Average Number of security components between «IT Application Blocks» is computed counting, for each *«IT Application Block»,* the minimum number of *«IT Blocks»,* which attribute "*security*" value is "*YES*", that are between that block and all the other *«IT Application Blocks»*. $$NSCBITAB_{ISA} = \frac{\sum_{i=1}^{\#IT\ Application\ Block} \left[\sum_{j=1}^{\#«IT\ Application\ Block»} Min\ \{\#SITB_{ij}\} \right]}{\left(\#«IT\ Application\ Block»\right)^2}$$, where: *$Min\{\#SITB_{ij}\}$*– is the minimum number of *«IT Blocks»* which attribute "security" value is "YES" that are between *«IT Application Block»$_i$* and *«IT Application Block»$_j$.* *#«IT Application Block»* – is the number of *«IT Application Blocks»*
Scale	$0;+\infty$
Architectural Levels	Technological Architecture
ISA Primitives and attributes	Primitive: *«IT Block»* Attribute: *security*
ISA Qualities	The Security of an ISA tends to increase with this metric.
Support	The ISA security is increased by putting security elements on it, as IDS, firewalls, etc (Rito 2004). 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.

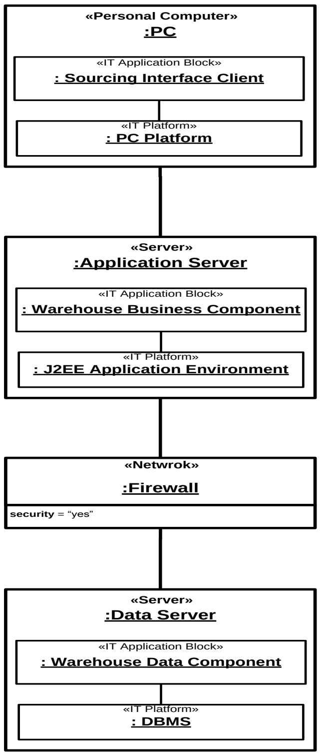
Example(s)

Example ISA A

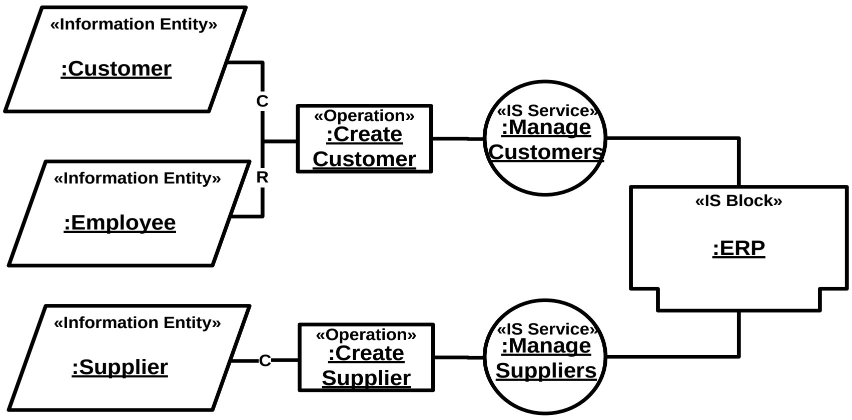


$$NSCBITAB = \frac{(1+2)+(1+1)+(1+2)}{3^2} = \frac{8}{9}$$

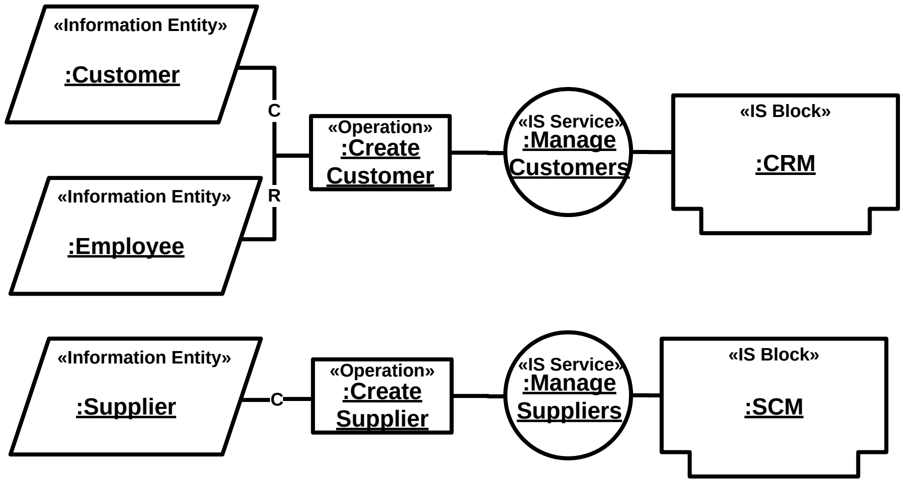
Example ISA B



$$NSCBITAB = \frac{(1+1)+(0+1)+(0+1)}{3^2} = \frac{4}{9}$$

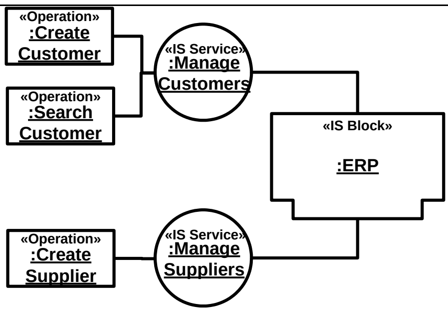
Acronym	*LCOIS* (or *LCOIS* $_{ISA}$)
Name	*Average Lack of COhesion in «IS Blocks».*
Computation	The Average Lack of Cohesion in «IS Blocks» is computed counting the number of sets of information entities that are used by distinct functionalities of the same application (provided by operations in «IS Blocks»). $LCOIS_{ISA} = \frac{\sum_{i=1}^{\#IS\ Block} \#LCOIS_i}{\#«IS\ Block»}$, 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»*
Scale	$[0;+\infty[$
Architectural Levels	Technological Architecture
ISA Primitives and attributes	Primitive: *«IT Block»* Attribute: *security*
ISA Qualities	The <u>security</u> of an ISA tends to increase with this metric.
Support	This metric measures 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.
Example(s)	Example ISA A  $LCOIS_{ISA} = \frac{1+1}{1} = 2$

Example ISA B

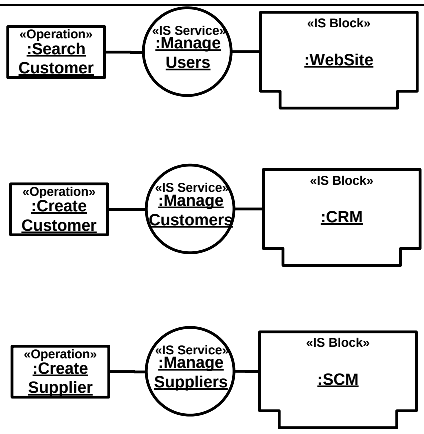


$$LCOIS_{ISA} = \frac{1+1}{2} = 1$$

Acronym	*NOIS* (or *NOIS $_{ISA}$*)
Name	*Average Number of Operations in «IS Blocks»*
Computation	The Average Number of Operations in «IS Blocks» is computed counting the number of operations on each «IS Block» divided be the number of «IS Blocks» $NOIS_{ISA} = \frac{\sum_{i=1}^{\#IS\ Block} \#«operation»_{«ISBlock»i}}{\#«IS\ Block»}$, where: *#«operation»$_{«IS\ Block»i}$* – is the number of operations on *«IS Block» i.* *#«IS Block»* – is the number of *«IS Block»*
Scale	$0;+\infty$
Architectural Levels	Application Architecture
ISA Primitives and attributes	Primitive: *«IS Block»* ; *«operation»*
ISA Qualities	The <u>modificability</u> of an ISA tends to be reduced with the increase of this metric
Support	The simplicity of adapt/alter operations in an ISA to new business demands is maximised when the impact of changing each operation is reduced to a certain application block (*«IS Block»*). This metric measures this fact.
Example(s)	Example ISA A Example ISA B



$$NOIS_{ISA} = \frac{3}{1} = 3$$

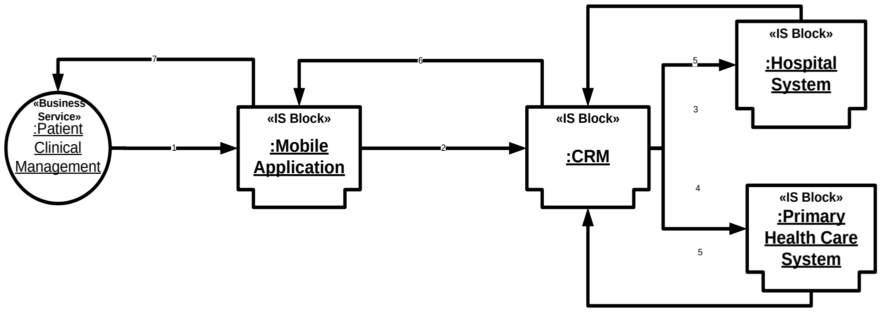


$$NOIS_{ISA} = \frac{1+1+1}{3} = 1$$

Acronym	*SCC* (or SCC_{ISA})
Name	*Average Service Cyclomatic Complexity*
Computation	The Average Service Cyclomatic Complexity is computed considering the average, number of dependencies between «IS Blocks» subtracted by the number of «IS Blocks» that support the service, for each service. $$SCC_{ISA} = \frac{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} e_i - n_i + 2}{\#«Business\ Service»+\#«IS\ Service»}$$, 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»*
Scale	$1;+\infty$
Architectural Levels	Application Architecture
ISA Primitives and attributes	Primitive: *«IS Block»* ; *«Business Service»*
ISA Qualities	The complexity of an ISA tends to increase with this metric. The modificability of an ISA tends to decrease with the increase of this metric.
Support	Like McCabe (1976), 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 Vasconcelos et. al. (2005) 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.

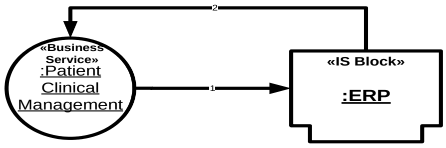
Example(s)

Example ISA A



$$SCC_{ISA} = \frac{8-4+2}{1} = 6$$

Example ISA B

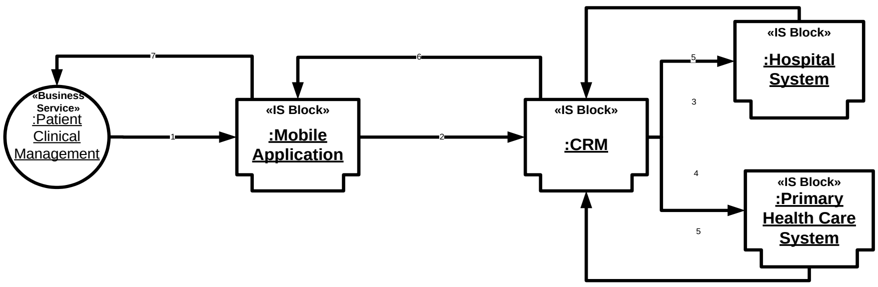


$$SCC_{ISA} = \frac{2-1+2}{1} = 3$$

Acronym	*RS* (or RS_{ISA})
Name	*Average Response for a Service*
Computation	The Average Response for a Service is computed by considering the average of the number of «*IS Blocks*» that might be used to support each *«Service»*: $$RS_{ISA} = \frac{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} \#«IS\ Block»_i}{\#«Business\ Service»+\#«IS\ Service»}$$, 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»*
Scale	$]0;+\infty[$
Architectural Levels	Application Architecture
ISA Primitives and attributes	Primitives: *«IS Block»* ; *«Business Service»*
ISA Qualities	The complexity of an ISA tends to increase with this metric
Support	Similar to the software metric "Response For a Class" – see Chidamber and Kemerer (1995) and Basili (1996) for further details – that computes the number of methods that can potentially be executed in response to a message received. In Vasconcelos et. al. (2005) 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. In recent researches Sousa, Pereira and Marques (2004) suggest that each business process should be supported by the less number of applications as possible – this is also measure by this metric.

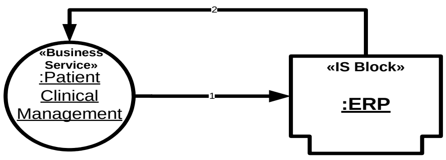
Example(s)

Example ISA A



$$RS_{ISA} = \frac{4}{1} = 4$$

Example ISA B

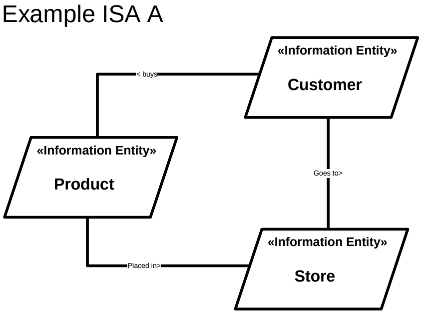


$$RS_{ISA} = \frac{1}{1} = 1$$

Acronym	*NE* (or NE_{ISA})
Name	*Number of Entities*
Computation	The *Number of Entities* (of an ISA) is computed by counting the number of *«Information Entities»*.
Scale	$0;+\infty$
Architectural Levels	Information Architecture
ISA Primitives and attributes	Primitive: *«Information entity»*
ISA Qualities	The *Maintainability* of an ISA tends to decrease with this metric increase.
Support	According to Género, Poels and Piattini (2003), this metric is strongly related to the Maintaining of an ISA.

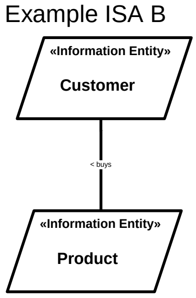
Example(s)

Example ISA A



$NE_{ISA}=3$

Example ISA B

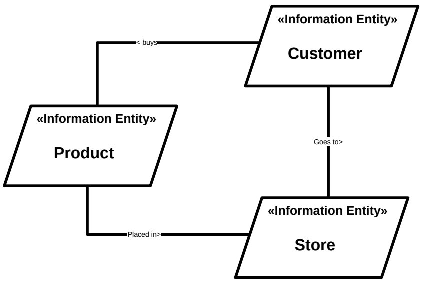


$NE_{ISA}=2$

Acronym	*NR* (or NR_{ISA})
Name	*Number of Relations*
Computation	The *Number of Relations of an Information Architecture* is computed by counting the number of relations between *«Information Entities»*
Scale	$[0;+\infty[$
Architectural Levels	Information Architecture
ISA Primitives and attributes	Primitive: *«Information entity»*
ISA Qualities	The *Maintainability* of an ISA tends to decrease with this metric increase
Support	According to Género, Poels and Piattini (2003), this metric is strongly related to the Maintaining of an ISA.

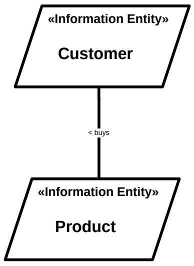
Example(s)

Example ISA A



$NR_{ISA}=3$

Example ISA B



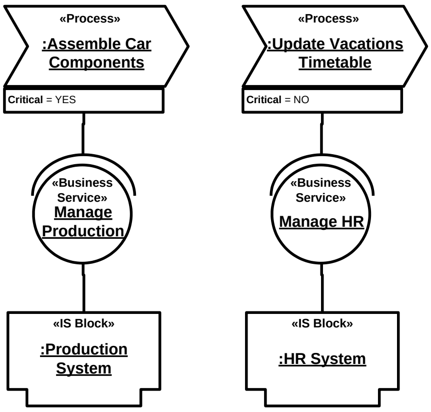
$NR_{ISA}=1$

Acronym	*CPSM* (or $CPSM_{ISA}$)
Name	Critical Process - System *Mismatch*
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 $CPSM_{ISA} = \frac{\#\{Process_C \in ISBlock_{NC}\} + \#\{Process_{NC} \in ISBlock_C\}}{\#«Process»}$, 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
Scale	$[0;1]$
Architectural Levels	Business Architecture and Application Architecture
ISA Primitives and attributes	Primitive: - *«IS Block»* ; - *«Process» - attribute: Critical ={Yes, No}*

ISA Qualities	Business-Application Alignment
Support	As described in Sousa, Pereira and Marques (2004) the critical business processes should be supported by different applications than non-critical business processes.

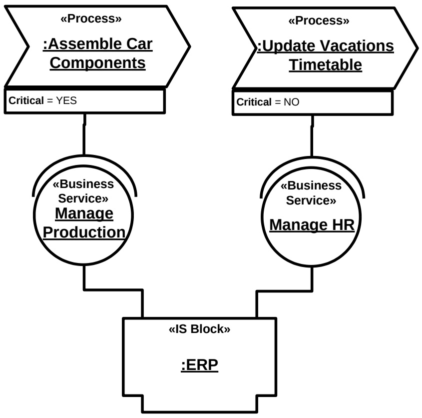
Example(s)

Example ISA A



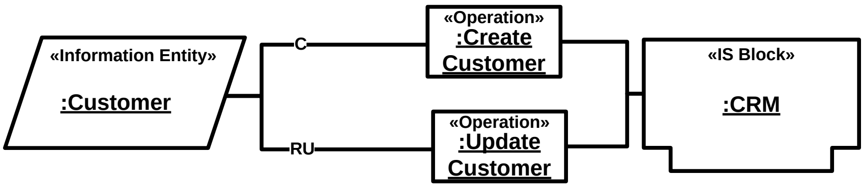
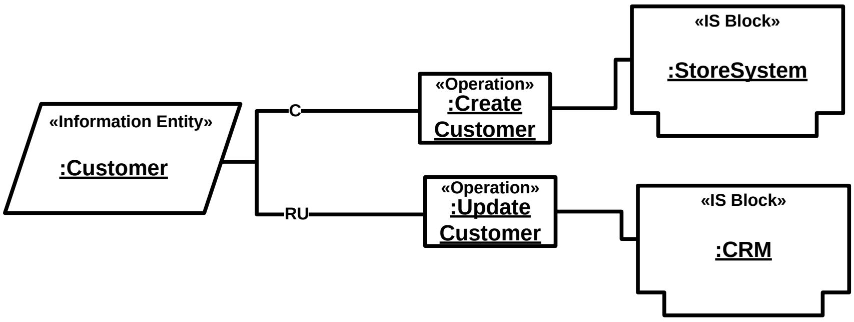
$$CPSM_{ISA} = \frac{0+0}{2} = 0$$

Example ISA B



$$CPSM_{ISA} = \frac{1+1}{2} = 1$$

Acronym	*NAIE* (or $NAIE_{ISA}$)
Name	Average Number of Applications per «Information Entity»
Computation	The Average Number of Applications per «Information Entity» is computed counting the average number of applications («IS Blocks») that through its «operations» support each «information entity». $NAIE_{ISA} = \frac{\sum_{i=1}^{\#InformationEntity} \#ISBlocks \in \exists «operation» \ CUD \ «InformationEntity»_i}{\#«Information\ Entity»}$, 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*»
Scale	$0;+\infty$
Architectural Levels	Information Architecture and Application Architecture
ISA Primitives and attributes	Primitive: - *«IS Block»* - *«Information entity»* - *«operation»*
ISA Qualities	Information Architecture – Application Architecture Alignment
Support	According to Sousa, Pereira e Marques (2004) each information entity should be managed by a single application.

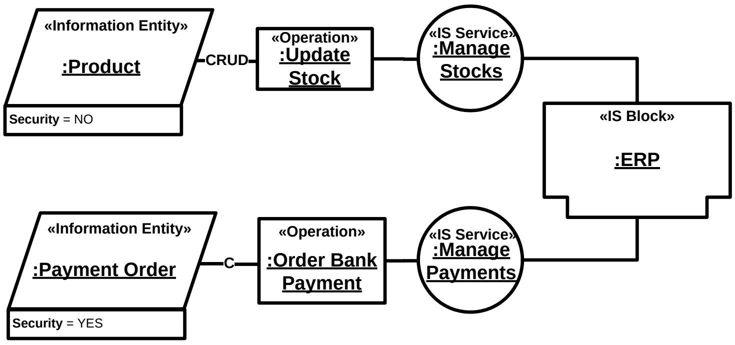
Example(s)	Example ISA A  $NAIE_{ISA} = \frac{1}{1} = 1$ Example ISA B  $NAIE_{ISA} = \frac{2}{1} = 2$

Acronym	*IESSM* (or *IESSM* $_{ISA}$)
Name	*Information Entity - System Security Mismatch*
Computation	The Information Entity - System Security Mismatch 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 the number of information entities with low-level security requirements supported in «IS Blocks» that also support information entities with high level security requirements. $IESSM_{ISA} = \frac{\#\{InformationEntity_{s} \in ISBlock_{NS}\} + \#\{InformationEntity_{NS} \in ISBlock_{S}\}}{\#«InformationEntity»}$, 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*». $\#\{InformationEntity_{NS} \in ISBlock_{S}\}$ – 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.
Scale	$[0;1]$
Architectural Levels	Information Architecture and Application Architecture
ISA Primitives and attributes	Primitive: - *«IS Block»*

	- *«Information entity»;* Attribute: *Security ={Yes, No}* - *«operation»*
ISA Qualities	Information Architecture – Application Architecture Alignment
Support	According to Sousa, Pereira e Marques (2004) applications should manage information entities of the same security level.

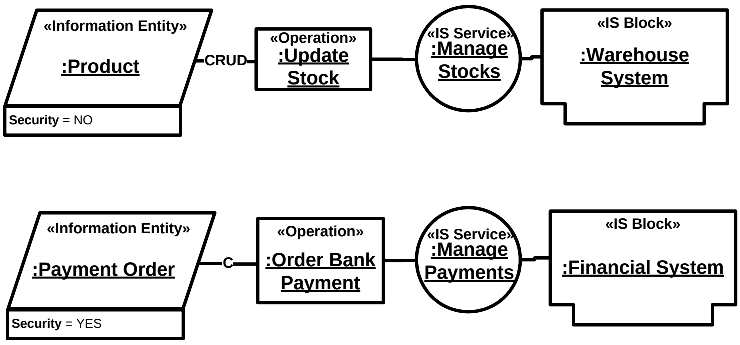
Example(s)

Example ISA A



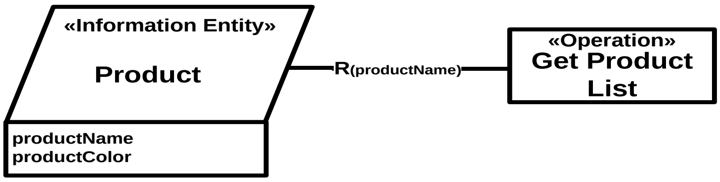
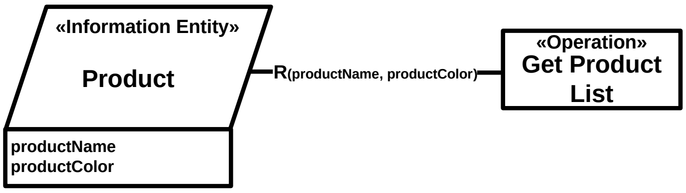
$$IESSM_{ISA} = \frac{1+1}{2} = 1$$

Example ISA B



$$IESSM_{ISA} = \frac{0+0}{2} = 0$$

Acronym	*NUIEA* (or $NUIEA_{ISA}$)
Name	*Average Number of Unused Information Entity Attributes*
Computation	The Average Number of Unused Information Entity Attributes is computed counting the number of attributes in information entities that are not used in any Read (R) «operation». $$NUIEA_{ISA} = \frac{\sum_{i=1}^{\#«Information\ Entity»} \sum_{j=1}^{\#attribute \in InformationEntity_i} attribute_j\ NOT\ \ R(\forall «operation»)}{\sum_{i=1}^{\#«Information\ Entity»} \#\, attribute \in InformationEntity_i}$$, where:

	$\sum_{i=1}^{\#«Information\ Entity»} \sum_{j=1}^{\#attribute \in InformationEntity_i} attribute_j\ NOT\ \ R(\forall «operation»)$ - is the total number of attributes of «*Information Entity*» *i* that are not read (*R*) by any *«operation»* $\sum_{i=1}^{\#«Information\ Entity»} \#attribute \in InformationEntity_i$ - is the total number of attributes in all the *«information entities»* of the ISA *#«Information Entity»* – is the number *«information entities»*
Scale	$[0;1]$
Architectural Levels	Information Architecture and Application Architecture
ISA Primitives and attributes	Primitive: - *«Information entity»* - *«operation»*
ISA Qualities	Information Architecture – Application Architecture Alignment
Support	According to Sousa, Pereira e Marques (2004) all the information entities' attributes should be read for, at least, a business process.
Example(s)	Example ISA A  $NUIEA_{ISA} = \frac{1}{2}$ Example ISA B  $NUIEA_{ISA} = \frac{0}{2} = 0$

4. An ISA evaluation case study

In this section we briefly describe a real case study where some of the previously proposed metrics are used in the process of building, analysing and improving an ISA in a Portuguese e-government project[i]. We start by describing the global project goals, in section 4.1, and then (in section 4.2) we focus on presenting how the CEO modelling framework and metrics supported the ISA definition for a project phase. In section 4.3 we present a short discussion on the case study results.

4.1 The enterprise life cycle project

In order to improve government services for enterprises UMIC - Knowledge Society Agency (a Portuguese governmental organisation) set out the Enterprise Life Cycle Project. This project's major goals are:

- Implement an "Electronic Enterprise Folder", dematerialising and providing enterprise information that currently is disperse and sometimes incoherent in different government organisations, through the Internet.
- Reengineer, improve and accelerate the Firm Start-up Process – that by the time of the project definition took, in average, between 27 and 65 days.

- Reposition and reorganise the government departments in order to provide a best, cheaper and agile service to enterprises and entrepreneurs.
- Implement the Enterprise Portal, the preferred channel for government services to Enterprises.

In the meantime of this project definition and kick-off, another initiative was implemented: The "On the Spot Firm". This initiative makes possible for entrepreneurs to create a company in just one office (one-stop office) in a single day (currently the average time is 1 hour and 14 minutes). Thus "The Enterprise Life Cycle Project" was split into two major phases; a first phase that pretends to implement the "Enterprise Portal" and to make the "On the Spot Firm offline process" also available online (in the Enterprise Portal). The second phase is expected to achieve the other project goals (as implementing the "Electronic Enterprise Folder", Repositioning and reorganising the government departments and reengineering the Firm start-up Process). In this article we will focus only on the first phase of the project (since the second one is still starting).

4.2 Selecting the "right" ISA

The "on the spot" firm start-up process major difference to the traditional company creation process is that the members of the future company may only choose the company name from a set of pre-approved firm names and a set of pre-approved association packs. Currently the entrepreneurs can create an "on the spot" company only on physical desks. This project first phase will also make available this process on the Internet – the process is described on **Figure 4.**

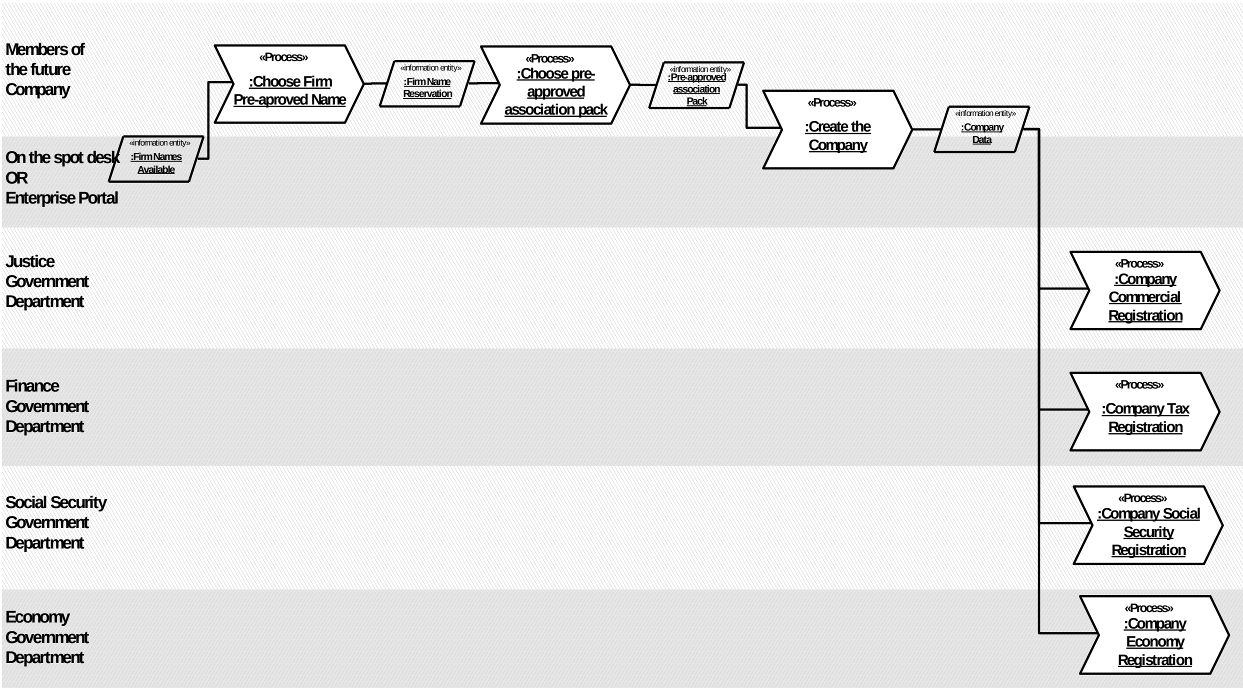


Figure 4. On the spot firm start-up process

Considering that an application that supports the "offline" on the spot company start-up process already exists, two major options were analysed before implementing the company start-up process online. The first option ("Architecture **A**") considered was to used the "On the spot firm" application to support the online creation of a company in the Enterprise Portal and use this application to directly ("point-to-point") integrate with the other government departments' information systems – see **Figure 5**.

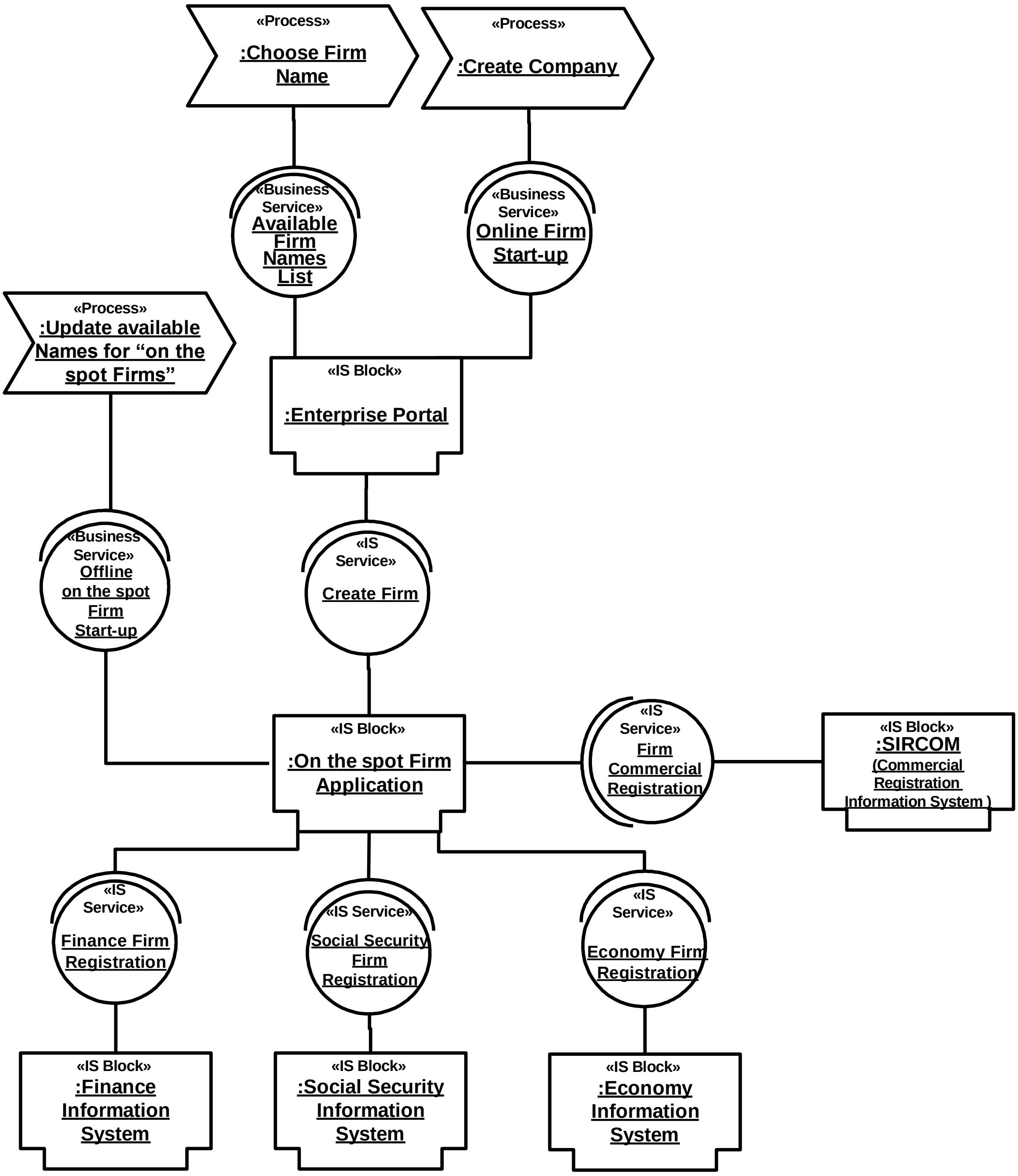


Figure 5. ISA "A" for the online company creation process

Another option considered (ISA "**B**") was to implement an integration/interoperability layer that would be globally responsible for the "company creation online business process", integrating with the different applications (namely the "on the spot" application, for getting the available pre-approved firm names) – see **Figure 6**

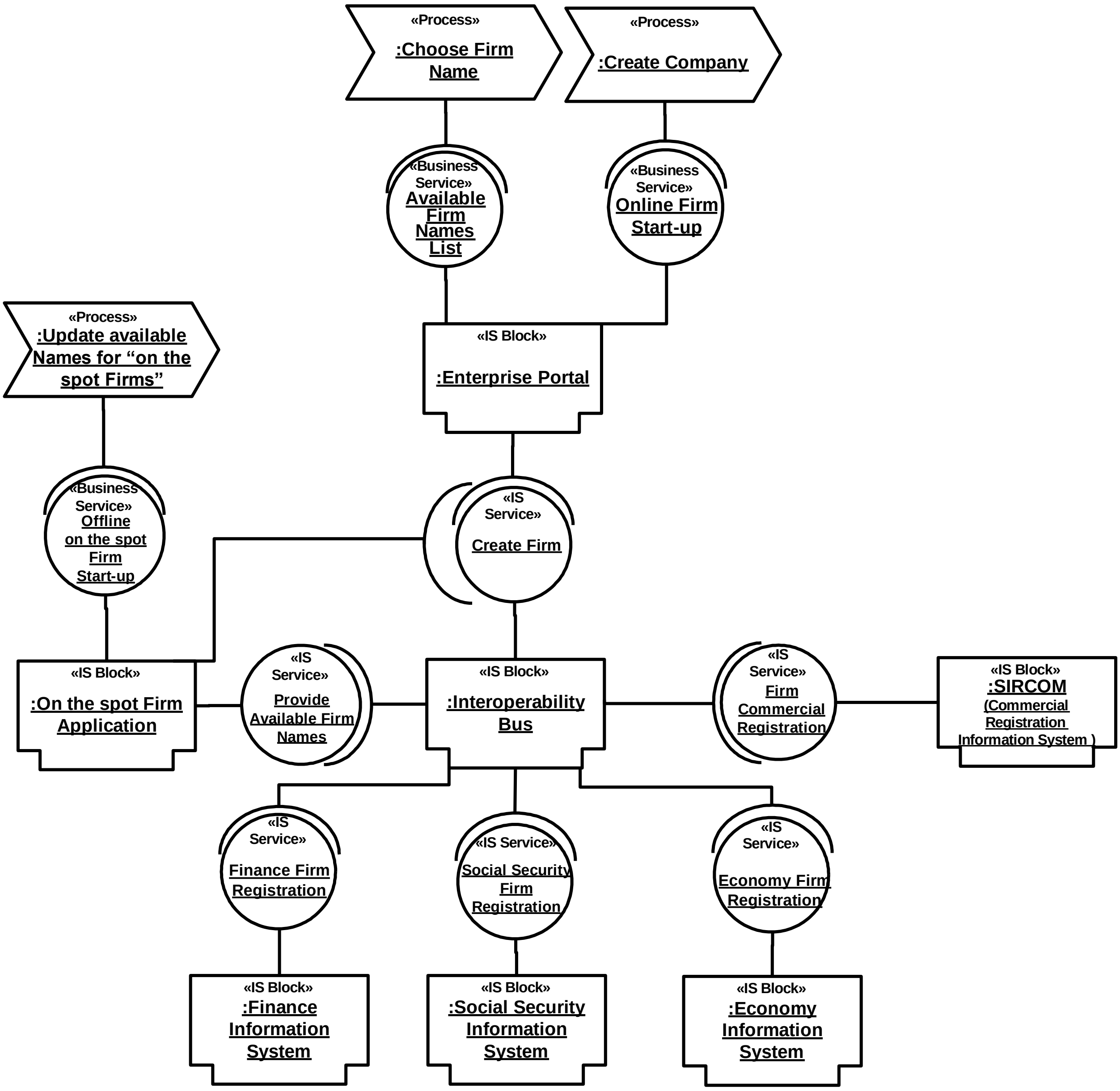


Figure 6. ISA "B" for the online company creation process

Both architectures were analysed using some of the previously metrics.

For the Information Entity - System Security Mismatch metric applied to architecture A (**Figure 7**) (considering the presented simplified architecture, with only two information entities) we have:

$$IESSM_A = \frac{\#\{InformationEntity_s \in ISBlock_{NS}\} + \#\{InformationEntity_{NS} \in ISBlock_S\}}{\#«InformationEntity»} = \frac{1+1}{2} = 1$$

And the Information Entity - System Security Mismatch metric, for (simplified) architecture B (**Figure 8**), value is:

$$IESSM_B = \frac{\#\{InformationEntity_s \in ISBlock_{NS}\} + \#\{InformationEntity_{NS} \in ISBlock_S\}}{\#«InformationEntity»} = \frac{0+0}{2} = 0$$

For the *Average Number of Applications per «Information Entity»* metric, for ISA A we have:

$$NAIE_A = \frac{\sum_{i=1}^{\#InformationEntity} \#\left|ISBlocks \in \exists«operation»\ CUD\ «InformationEntity»_i\right|}{\#«Information\ Entity»} = \frac{2+1}{2} = \frac{3}{2}$$

And for ISA B the *Average Number of Applications per «Information Entity»* metric is:

$$NAIE_A = \frac{\sum_{i=1}^{\#InformationEntity} \#\left|ISBlocks \in \exists«operation»\ CUD\ «InformationEntity»_i\right|}{\#«Information\ Entity»} = \frac{1+1}{2} = 1$$

These metrics point that ISA B has a better align between its application and information architectures than ISA A.

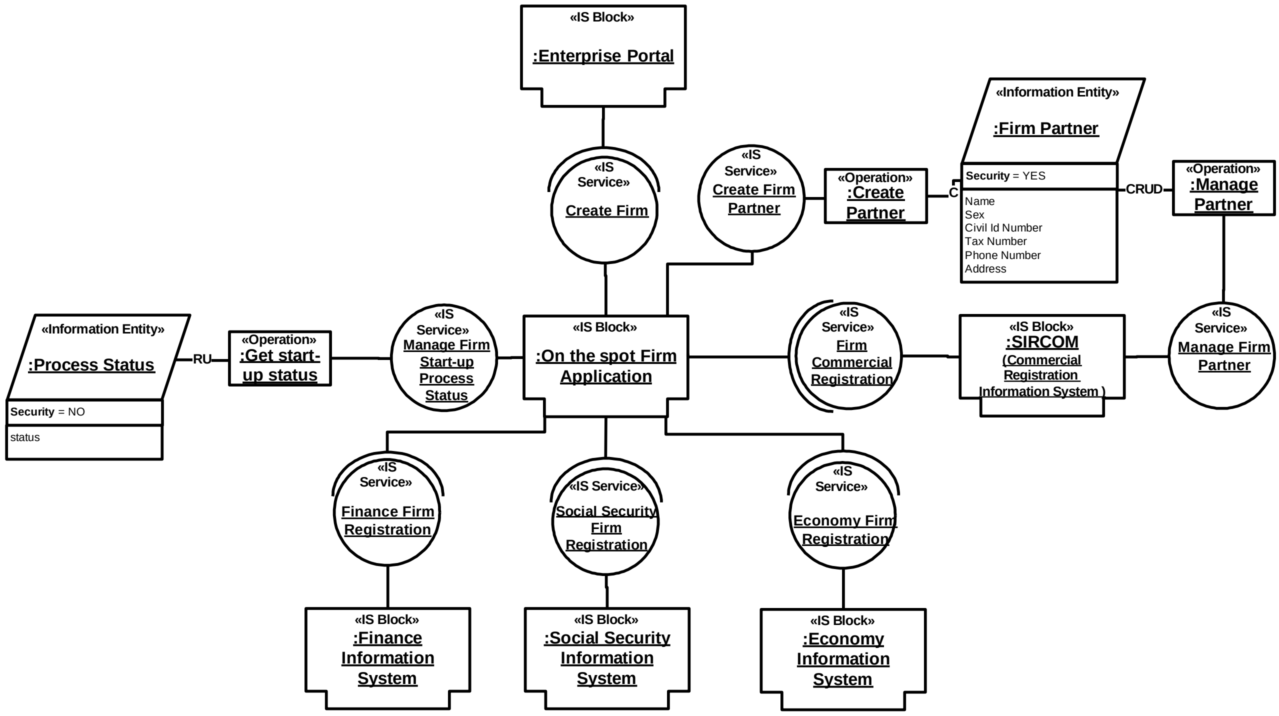


Figure 7. ISA A – Application-information architectures (partial view)

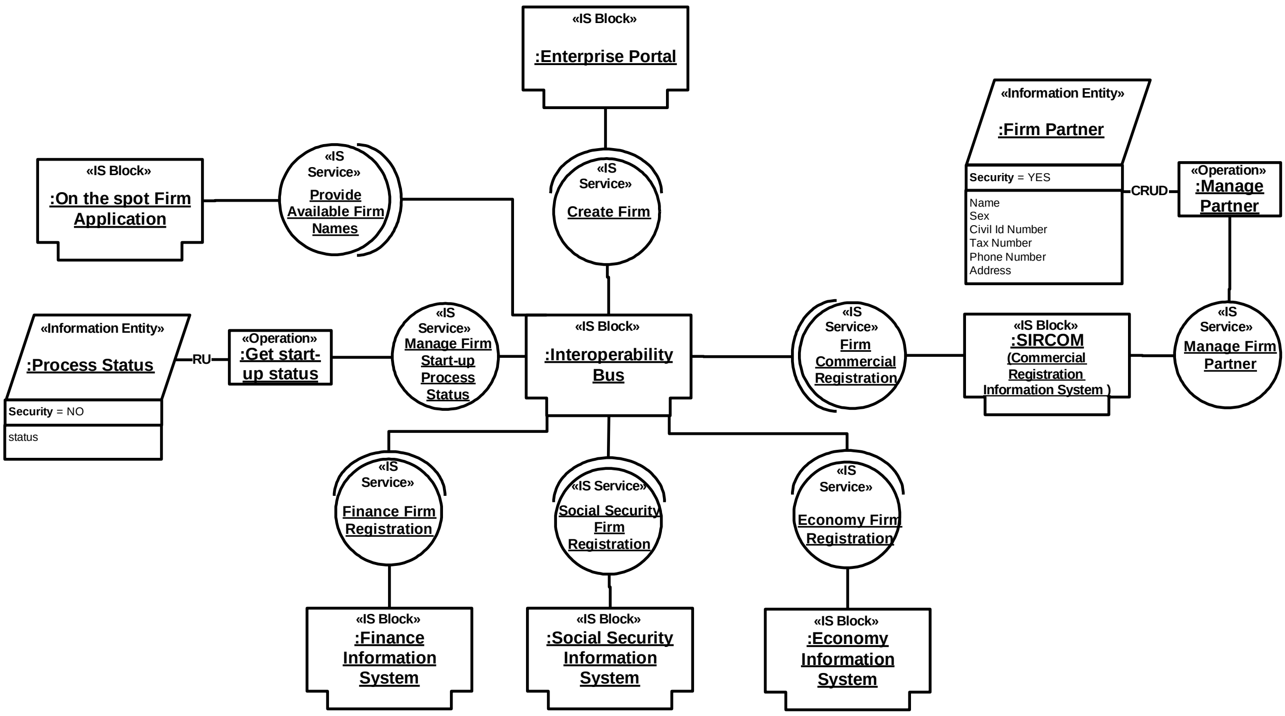


Figure 8. ISA B – Application-information architectures (partial view)

In terms of complexity, two metrics were used: The *Average Response for a Service* metric and the *Average Service Cyclomatic Complexity* metric.

The Average Response for a Service metric (considering only the three business services described in **Figure 9** to **Figure 14**, for simplicity) for ISA A is:

$$RS_A = \frac{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} \#«IS\ Block»_i}{\#«Business\ Service»+\#«IS\ Service»} = \frac{2+6+5}{3} = \frac{13}{3}$$

And for ISA B:

$$RS_B = \frac{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} \#«IS\ Block»_i}{\#«Business\ Service»+\#«IS\ Service»} = \frac{3+6+6}{3} = \frac{15}{3} = 5$$

For the *Average Service Cyclomatic Complexity* metric for ISA A, we have:

$$SCC_A = \frac{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} e_i - n_i + 2}{\#«Business\ Service»+\#«IS\ Service»} = \frac{(4-2+2)+(8-6+2)+(6-5+2)}{3} = \frac{11}{3}$$

And the *Average Service Cyclomatic Complexity* metric for ISA B is:

$$SCC_B = \frac{\sum_{i=1}^{\#«Business\ Service»+\#«IS\ Service»} e_i - n_i + 2}{\#«Business\ Service»+\#«IS\ Service»} = \frac{(6-3+2)+(8-6+2)+(8-6+2)}{3} = \frac{13}{3}$$

Thus, these metrics indicate that ISA B is slightly more complex than ISA A.

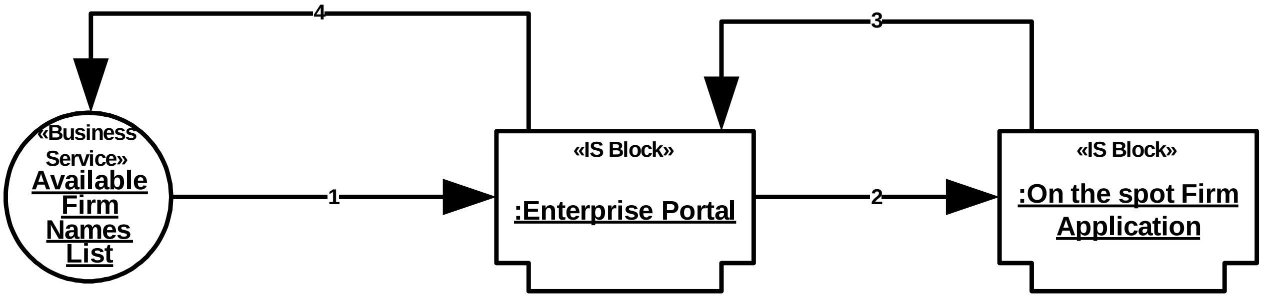


Figure 9. Available firm names list collaboration diagram (ISA A)

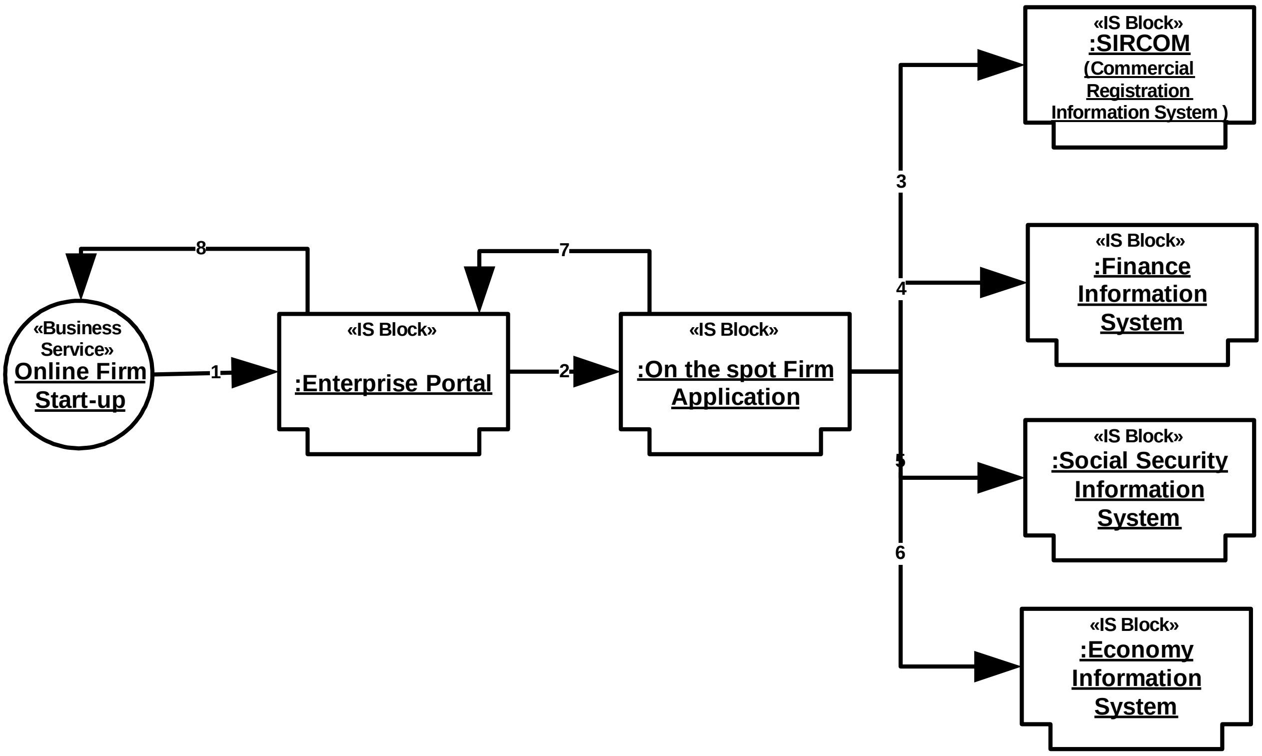


Figure 10. Online firm start-up collaboration diagram (ISA A)

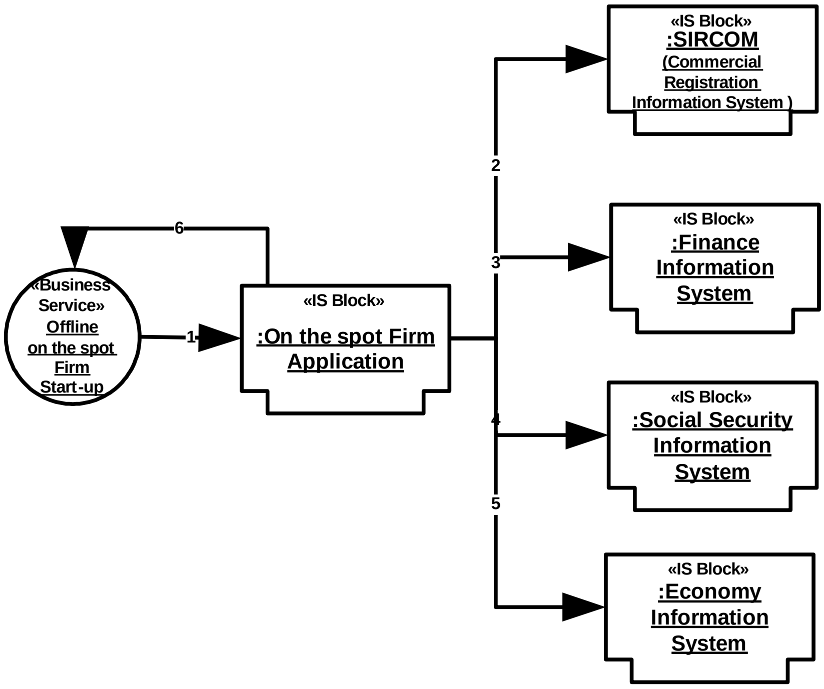


Figure 11. Offline on the spot firm start-up collaboration diagram (ISA A)

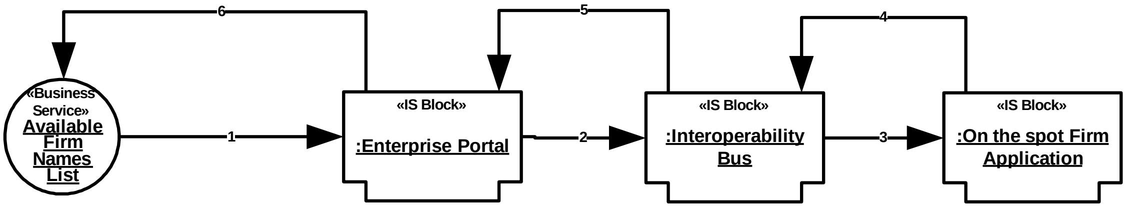


Figure 12. Available firm names list collaboration diagram (ISA B)

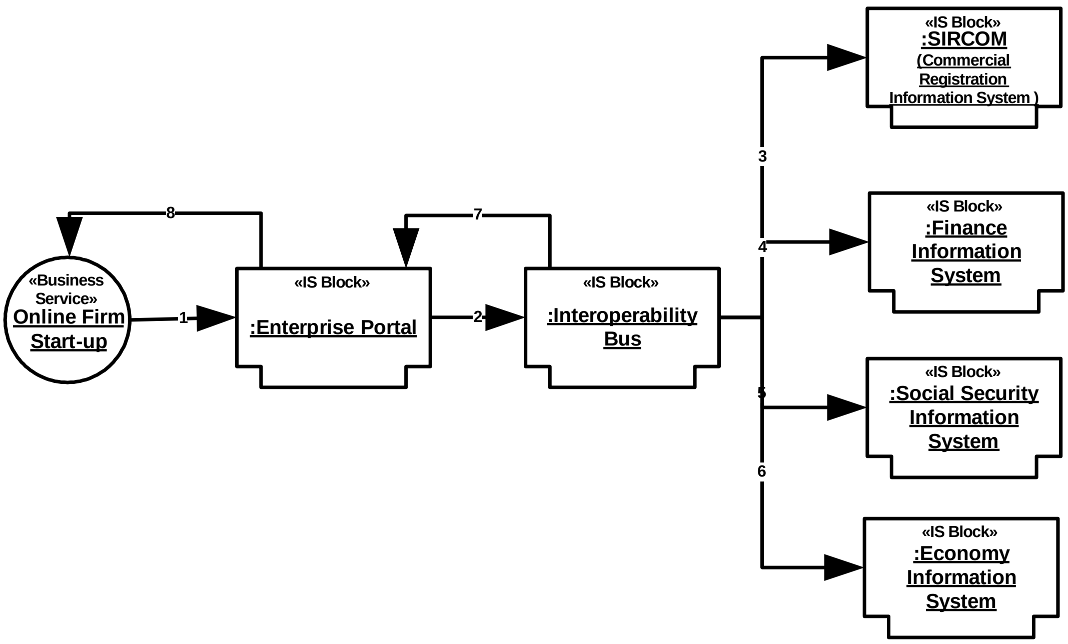


Figure 13. Online firm start-up collaboration diagram (ISA B)

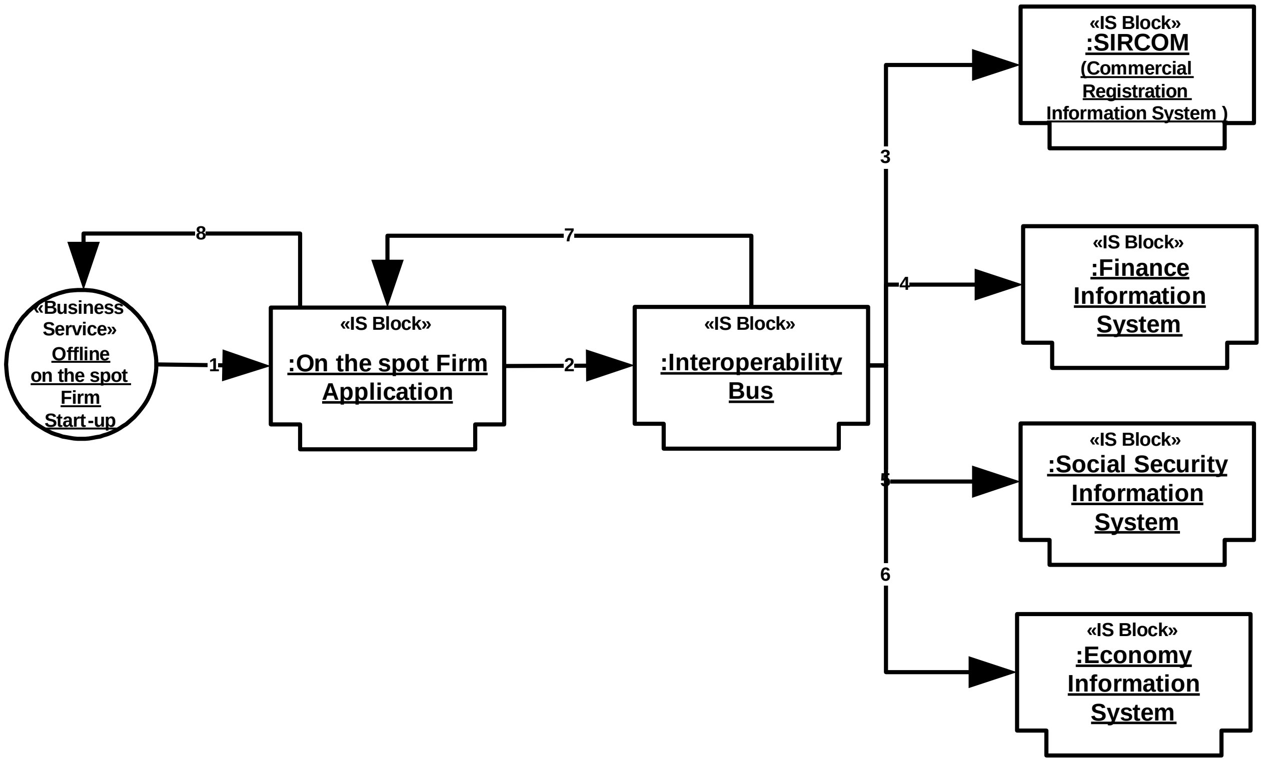


Figure 14. Offline on the spot firm start-up collaboration diagram (ISA B)

In order to evaluate the security of both ISA, from a technological perspective, the technological architecture of ISA A and ISA B were modelled (**Figure 15** and **Figure 16**) and two metrics were applied.

The Average Number of security components metric values for ISA A and ISA B are:

$$NSC_A = \frac{\#SITB}{\#\text{«}IT\ Block\text{»}} = \frac{6}{59} = 10{,}17\%$$

$$NSC_B = \frac{\#SITB}{\#\text{«}IT\ Block\text{»}} = \frac{11}{102} = 10{,}78\%$$

And The Average Number of security components between «IT Application Blocks» metric values for ISA A and ISA B are:

$$NSCBITAB_A = \frac{\sum_{i=1}^{\#IT\ Application\ Block} \left[\sum_{j=1}^{\#\text{«}IT\ Application\ Block\text{»}} Min\ \{\#SITB_{ij}\} \right]}{\left(\#\text{«}IT\ Application\ Block\text{»}\right)^2} = \frac{19+8+8+8+8+8+8+8+8+14+14+14+14+43}{14^2} = 0{,}93$$

$$NSCBITAB_B = \frac{\sum_{i=1}^{\#IT\ Application\ Block} \left[\sum_{j=1}^{\#\text{«}IT\ Application\ Block\text{»}} Min\ \{\#SITB_{ij}\} \right]}{\left(\#\text{«}IT\ Application\ Block\text{»}\right)^2} = \frac{64+53+53+53+53+53+53+53+53+69+69+69+69+90+63+63+63+63+63+63+79+79+79+79}{24^2} = 2{,}68$$

These metrics point that ISA B has a higher security level than ISA A.

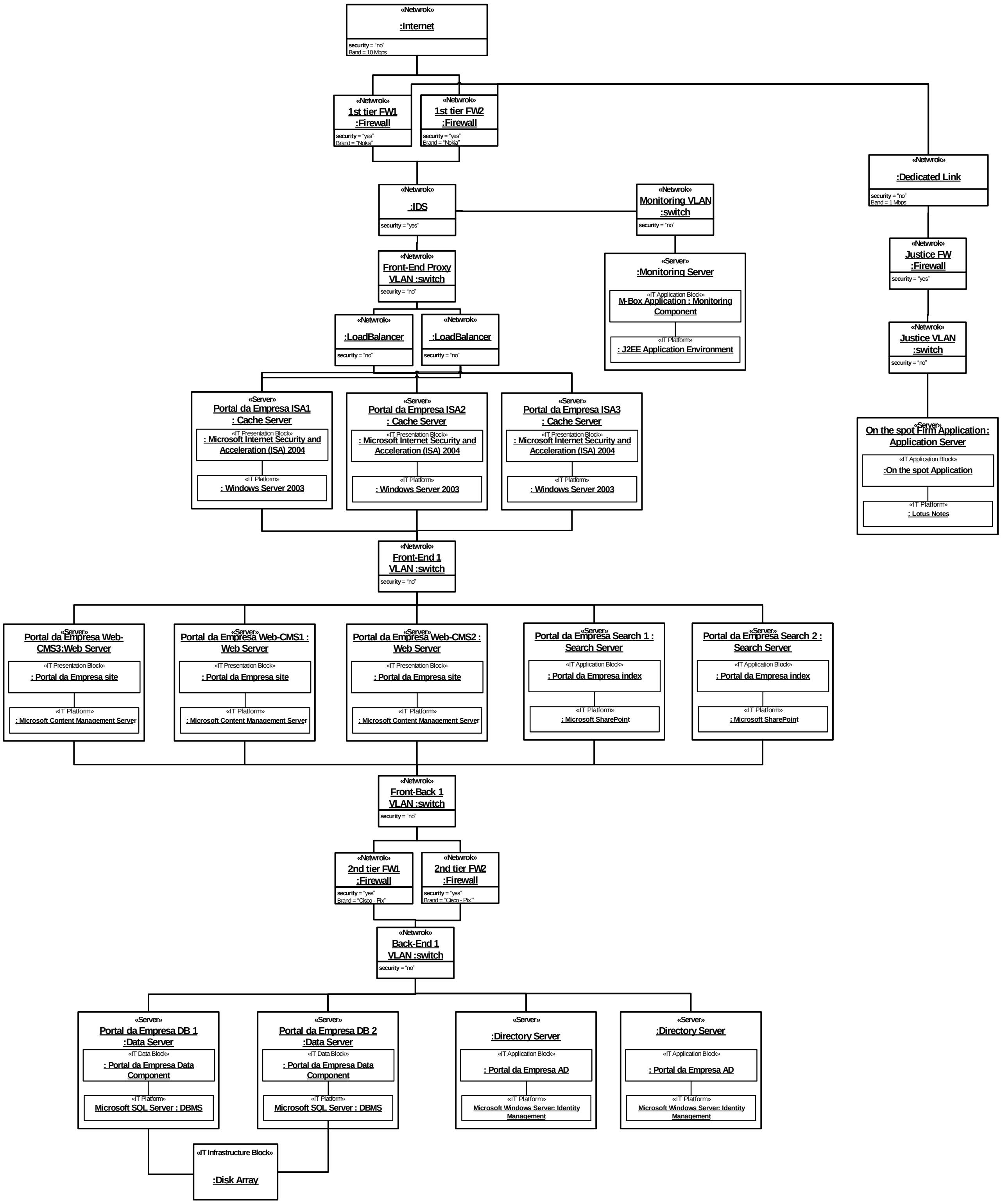


Figure 15. Technological architecture A

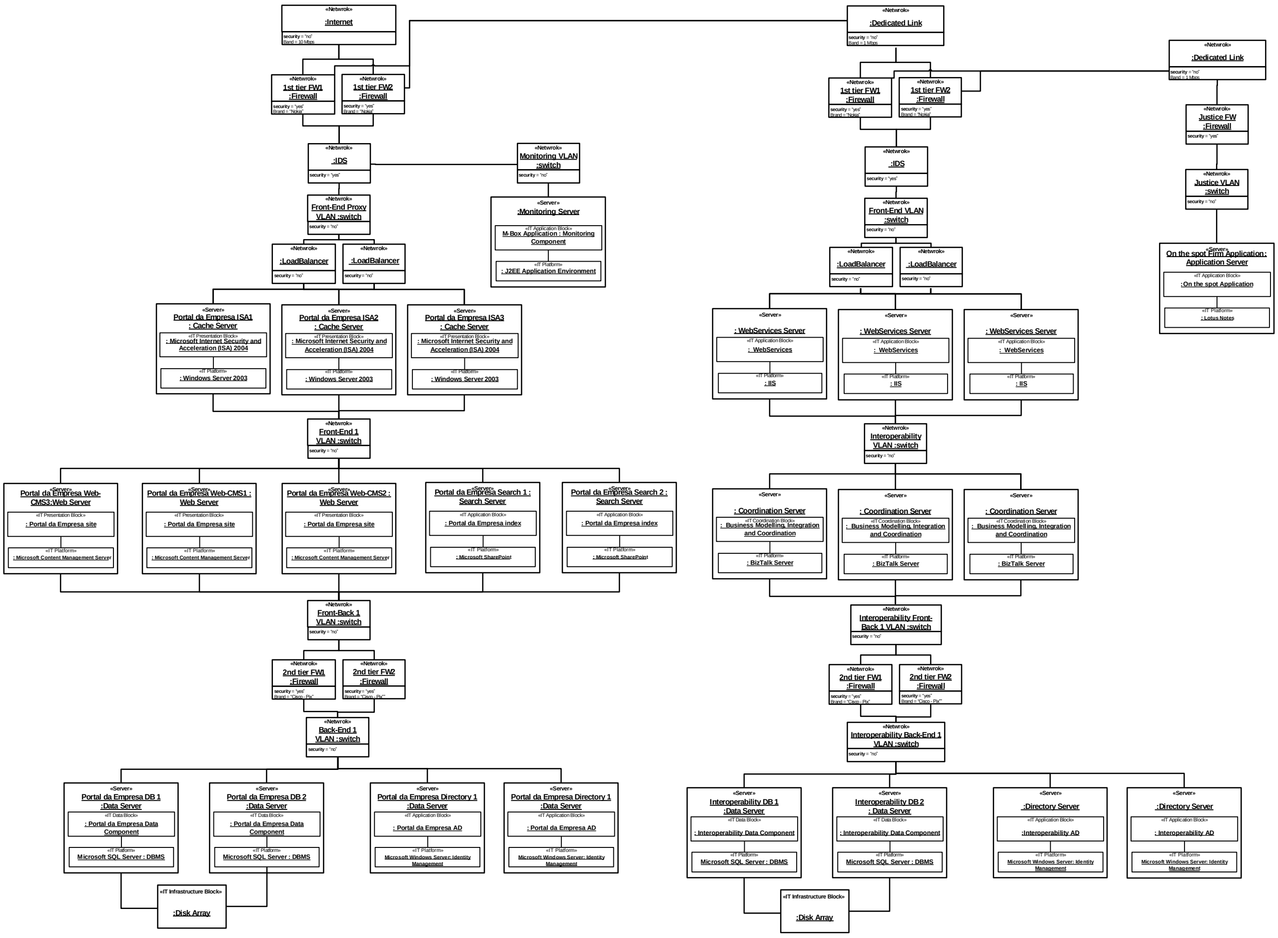


Figure 16. Technological architecture B

4.3 Discussion

The metrics used in the previous section point that:

- ISA **A** has a worst alignment between its application and information architecture than ISA **B;**
- ISA **A** is less complex than ISA **B**;
- ISA **A** is less secure than ISA **B**.

Considering that this phase of the project had a high level of pressure on its implementation timeframe, the complexity quality was considered of highest importance. Thus, since the Interoperability layer was not available, the ISA adopted for this first phase was ISA **A**. However, the project is now going to the second phase with several business and technological enhancements – like online creation of companies with names formulated online by the citizen (instead of picking up from a set of pre-approved ones), Electronic Folder on all the information available on the enterprise (after creation), among others. For this next phase the team is going to implement an ISA similar to ISA **B**, previously described.

5. Conclusions and future work

In this paper the authors proposed a set of ISA evaluation metrics, namely: Average Number of Possible Operating Systems, Average Number of Distinct Technologies for IS Services, Average Number of (Different) Implementations of an Information Entity, Average Number of stateful «IT Presentation Block» and «IT Logic Block», Average Number of security components, Average Number of security components between «IT Application Blocks», Average Lack of COhesion in «IS Blocks», Average Number of Operations in «IS Blocks», Average Service Cyclomatic Complexity, Average Response for a Service, Number of Entities, Number of Relations, Critical Process - System Mismatch, Average Number of Applications per «Information Entity», Information Entity - System Security Mismatch and Average Number of Unused Information Entity Attributes. Some of these metrics were applied in an e-government ISA evaluation project and revealed to be useful on the process of selecting the most appropriate ISA for a set of desired qualities. With these metrics, as described in the case study in this paper, the architect has a set of indicators on the impact of each of his or her decision during the process of building an ISA. However the authors recognised that much more testing on the metrics should be developed in order to assess its merit and significance. Currently, in other projects these metrics are being applied and improved. The implementation of a tool for automatically evaluate ISA according to a set of qualities is also a planned future work.

Acknowledgements

The case study research presented in this paper was possible thanks to the support of UMIC - Knowledge Society Agency.

References

Basili, V. R., L. C. Briand and W. L. Melo, *A validation of object-oriented design metrics as quality indicators*,"IEEE Trans. Software Eng. 22, 1996.

Bass, C., P. Clements, and R. Kazman, *Software Architecture in Practice*, Addison-Wesley, 1998

BEA, *Scaling EJB Applications*, available in http://edocs.bea.com/wle/tuning/tsejb.htm, access in May 2006.

Carnegie, *How do You Define Software Architecture?*, Software Engineering Institute, Carnegie Mellon University, December 2000. http://www.sei.cmu.edu/architecture/definitions.html.

Chidamber S., and C. Kemerer *A metrics suite for object-oriented design*. IEEE Trans. on Software Engineering, 20(6):476–493, 1995.

Clements,P., R. Kazman, and M.Klein, *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley Professional , ISBN 02017048, 2002

Cook, Melissa A., *Building Enterprise Information Architectures: Reengineering Information Systems*, Prentice Hall PTR, ISBN 0 13 440256-1, 1996

DeBoever, L., *Enterprise Architecture Boot Camp and Best Practices: A Workshop*, Meta Group, 1997.

Department of Defense, Department of Defense *Joint Technical Architecture*, July 2002.

Eriksson, Hans-Erik, and Magnus Penker, *Business Modeling with UML: Business Patterns at Work*, John Wiley and Sons, ISBN 0-471-29551-5, 2000.

Federal Enterprise Architecture Framework, version 1.1., September 1999

Garlan, D. et al., *Architectural Mismatch (Why It's Hard to Build Systems Out of Existing Parts)*, Proceedings 17th International Conference on Software Engineering, Seatle, WA, April 23-30 1995, pp.170-185

Genero, M., G. Poels, and M. Piattini, Defining and Validating Metrics for Assessing the Maintainability of Entity-Relationship Diagrams, Working Paper, Faculteit Economie En Bedrijfskunde, 2003.

Gillies, A. C., *Software Quality : Theory and Management*, Chapman and Hall, 1992.

IEEE Architecture Working Group, *Recommended Practice for Architecture Description – Draft IEEE standard P1471/D4.1*, IEEE, December 1998

Inmon, W., *Using The Generic Data Model* , White Paper, 2000.

Khaddaj, Souheil and G Horgan, *The Evaluation of Software Quality Factors in Very Large Information Systems,* The Electronic Journal of Information System Evaluation, Vol.7, Issue 1, 2004.

Maes, Rik, Daan Rijsenbrij, Onno Truijens, and Hans Goedvolk, *Redefining Business – IT Alignment Through a Unified Framework*, White Paper, May 2000. http://www.cs.vu.nl/~daan/

Marshall, Chris, *Enterprise Modeling with UML: Designing Successful Software through Business Analysis*, Addison Wesley Longman, ISBN 0 201-43313-3, 2000

McCabe, T. J.,`*A Complexity Measure*, IEEE Trans. Software Eng. 2, 1976.

OMG, Unified Modeling Language Specification, v 2.0, 2004, http://www.omg.org

Open Group, *The Open Group Architectural Framework (TOGAF) – Version 8.1*, The Open Group, December 2003

Rito, A., *Atributos de Qualidade*, Presentation on the Software Architecture Course, Instituto Superior Técnico, Technical University of Lisbon, 2004 (in Portuguese).

Roulo, M., *Java's three types of portability*, available in http://www.javaworld.com/javaworld/jw-05-1997/jw-05-portability_p.html, 1997 (access in May 2006).

Sarkis, J. and R. Sundarraj 2003, *Evaluating Componentised Enterprise Information Technologies: A Multiattribute Modeling Approach*, Information Systems Frontiers 5:3, 303–319, Kluwer Academic Publishers, 2003.

Sassoon, Jacques, *Urbanisation des systèmes d'information*, Editions Hermes, 1998 (in French)

Sousa, P., C. Pereira and J. Marques, *Enterprise Architecture Alignment Heuristics*, Journal 4, Microsoft Architects Journal, October 2004.

Spewak, Steven, and Steven Hill, *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology*, Wiley-QED, ISBN 0-471-599859, 1992

The Business Reference Model - A Foundation for Government-wide Improvement, 2002.

Tissot, Florence, and Wes Crump, *An Integrated Enterprise Modeling Environment*, P. Bernus, K. Mertins, G. Schmidt (Eds.), Handbook on Architectures of Information Systems, Springer, pp.59-79, ISBN 3-540-64453-9, 1998.

Vasconcelos, A., A. Caetano, J. Neves, P. Sinogas, R. Mendes, e J. Tribolet, *A Framework for Modeling Strategy, Business Processes and Information Systems*, 5th International Enterprise Distributed Object Computing Conference EDOC, Seatle, USA, 2001.

Vasconcelos, A., M. Silva, A. Fernandes, and J. Tribolet, *An Information System Architectural Framework for Enterprise Application Integration*, Proceedings of 37th Annual Hawaii International Conference On System Sciences (HICCS37), Hawaii, USA, January 2004b.

Vasconcelos, A., P. Sousa, and J. Tribolet, *Information System Architectures*, Proceedings of Business Excellence 2003, International Conference on Performance Measures, Benchmarking and Best Practices in New Economy, Guimarães, Portugal, June 2003.

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

Vasconcelos, A., R. Mendes, and J. Tribolet, *Using Organisational Modeling to Evaluate Health Care IS/IT Projects*, Proceedings of 37th Annual Hawaii International Conference On System Sciences (HICCS37), Hawaii, USA, January 2004a.

Vernadat, François, *Enterprise Modeling and Integration*, London, Chapman and Hall, 1996.

W3C, World Wide Web Consortium, Web Services, 2002, http://www.w3.org/2002/ws/

Zachman, John, *A Framework for Information System Architecture*, IBM system journal Vol.26 Nº 3, 1987, p.276 – 292.

Zijden, Stefan, Hans Goedvolk, and Daan Rijsenbrij, *Architecture: Enabling Business and IT Alignment in Information System Development*, 2000. http://www.cs.vu.nl/~daan/

[i] Some facts presented in this case study were changed for security and confidentiality reasons.