Formalizing Architecture Principles using Object-Role Modelling

G.J.N.M. (Guido) Chorus , Y.H.C. (Yves) Janse, C.J.P. (Chris) Nellen,
S.J.B.A. (Stijn) Hoppenbrouwers and H.A. (Erik) Proper

Institute for Computing and Information Sciences, Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, EU
{S.Hoppenbrouwers, E.Proper}@cs.ru.nl

Abstract. This technical report is the result of two experiments conducted as part of an ongoing research effort to formalize architecture principles. The experiment involves a first, and modest, evaluation of the use of ORM and ORC as a means to formalize and ground architecture principles. The experiments involve the evaluation of the use of ORM and ORC to formalize the example principles provided by the TOGAF (The Open Group Architecture Framework) and principles taken from industrial practice.

1 Introduction

This technical report is concerned with an experiment on the precise formulation and underpinning of architecture principles. It can readily be observed how large organizations increasingly make use of enterprise architectures to direct the development of the enterprise as a whole and IT development in particular [Lo05]. These developments are fuelled even more by government regulations such as the Clinger-Cohan Act in the USA [USA96], which requires government bodies to provide an IT architecture based on a set of architecture principles. A more specific way of expressing the role of enterprise is to state: "*Architecture serves the purpose of constraining design space"* [xAF06]. In most (enterprise) architecture approaches, this constraining is done by means of so-called architecture principles [IEE00, TOG04].

According to the TOGAF architecture framework [TOG04], "*Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission*". Such principles typically address the concerns of the key stakeholders within an organization. Some example principles, taken from [TOG04], are:

1 *Data is an asset that has value to the Enterprise and is managed accordingly.*
2 *Data is accessible for users to perform their functions.*
3 *Applications are independent of specific technology choices and therefore can operate on a variety of technology platforms.*

When considering principles in an enterprise architecture context, three perspectives on principles can be discerned:

1 *Principles as emergent laws*. These are essentially properties of (classes of) a system that can be observed and validated. Examples are the laws of nature, law of requisite variety, laws of social behaviour, etc.
2 *Principles as imposed laws.* Like emergent laws, they are properties that can be validated. However, imposed laws also require mechanisms to enforce them. Imposed laws typically address concerns of stakeholders. Some of these concerns may be raised by emergent laws having a negative

impact on the system / enterprise being designed. Examples are: societal laws, policies and regulations within organizations, etc.

3 *Principles as guidelines*. Desired properties that are so concrete that they offer guidelines to make operational behaviour fit imposed laws. For example: "Always use your car's cruise control" is an advisable property to abide by that provides guidance in obeying the law concerning maximum speeds on roads.

The principles we will mainly focus on in this report are *principles as imposed laws* and *principles as guidelines*. These are the kinds of principles that typically require enforcing and validation by stakeholders. In a development project these principles can best be regarded as general requirements that will apply to a class of systems.

While several sources attribute a pivotal role to principles, a precise definition of the concept of principles as well as the mechanisms and procedures needed to turn them into an effective means still lacks. Both IEEE [IEE00] and TOGAF [TOG04] position principles as a means to guide the design and evolution of enterprises/systems, while xAF [xAF06] goes even further by *equating* architecture to a set of (situation specific) principles. Nevertheless, no clear definition of principles and associated mechanisms and procedures are given. Such definitions are, however, direly needed if enterprise architecture is to become a steering mechanism as it is intended to be. Providing the ultimate answer to this question is also not the aim of this report, even though it is part of our ongoing research [BHPW06,OP07] in which we are indeed progressing towards such a definition.

In this report, however, we focus the evaluation of an approach to making the formulation of principles more precise. The conducted evaluation is a first and most evaluation, in that it has been performed by students. The aim of this modest evaluation is also to ascertain if further, more rigid, evaluation is worth while. In [BHPW06] a method for formalizing architecture principles was proposed using ORM and ORC: "*It is argued that when using ORM and ORC for formal modelling of architecture principles, the underlying logical principles of the techniques may lead to better insight into the rational structure of the principles.*" To this end, van Bommel et al. [BHPW06] formalized two exploratory example principles of The Open Group's Architecture Framework [TOG04]. In this report, we proceed to explore the other 18 principles presented in this framework to further test the value of formalizing architecture principles. Apart from the formalization of TOGAF principles, the practical applicability of the method has been evaluated by applying the method in an interactive joint modelling session with an enterprise architect. Reflections on of the formalization of the TOGAF principles and the results of the joint modelling session can be found at the end of this report.

2 First experiment: Formalization of Principles

In the first experiment, a group of three students:

G.J.N.M. (Guido) Chorus , Y.H.C. (Yves) Janse, C.J.P. (Chris) Nellen

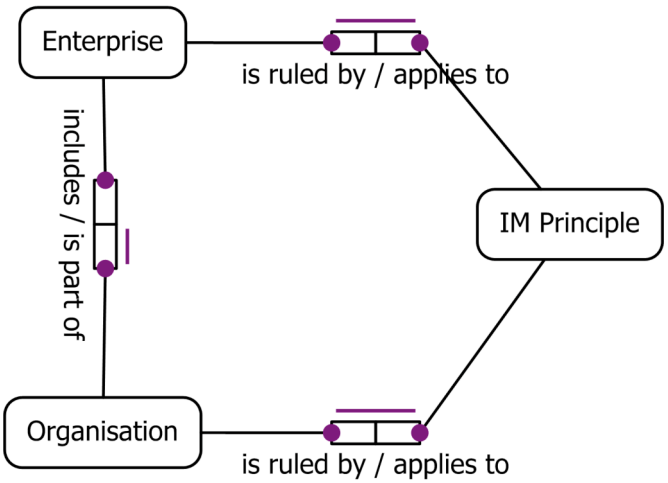
formalized the TOGAF architecture in line with the approach taken in [BHPW06], with the slight modification of using the ORM2 notation rather than the traditional ORM notation.

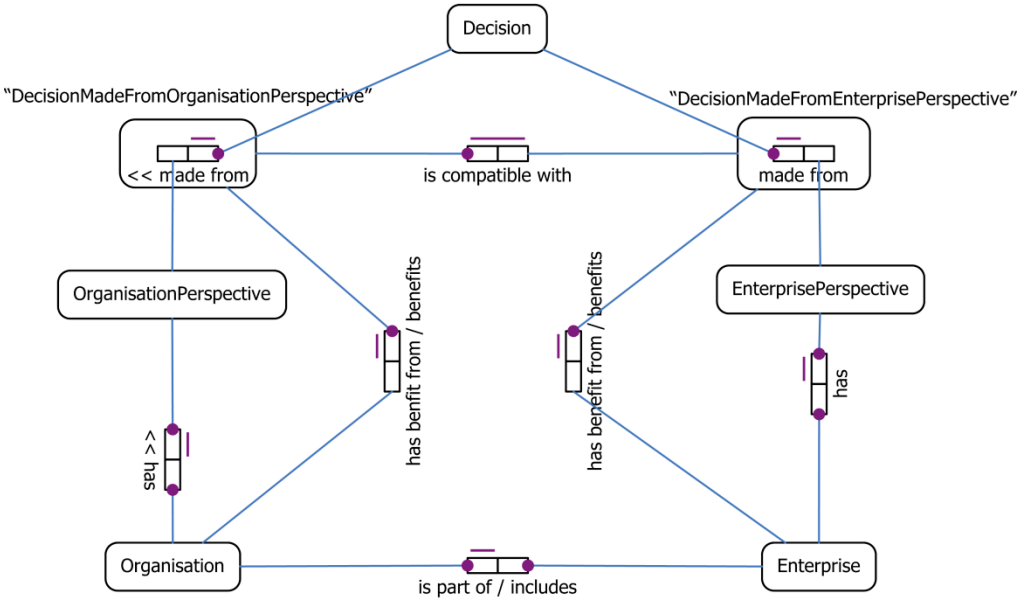
It is important to identify the difference in definitions of principle that TOGAF 8.1 discusses. TOGAF distinguishes three levels of principles. The top-most level is 'enterprise principles'. The 'IT principles' elaborates on this level. Finally 'architecture principles' are even more specific, a subset of IT principles that relate to the actual architecture. TOGAF defines twenty of such 'architecture principles'. These principles must therefore be a subset of the TOGAF IT principles. Strangely enough though, principles such as 'Maximize Benefit to the Enterprise' adhere more to general top-level enterprise principles instead of IT specific principles. Because of this discrepancy, this report uses the enterprise level definition whenever architecture principles are mentioned.

In the formalization of the 18 architecture principles, the following template was used:

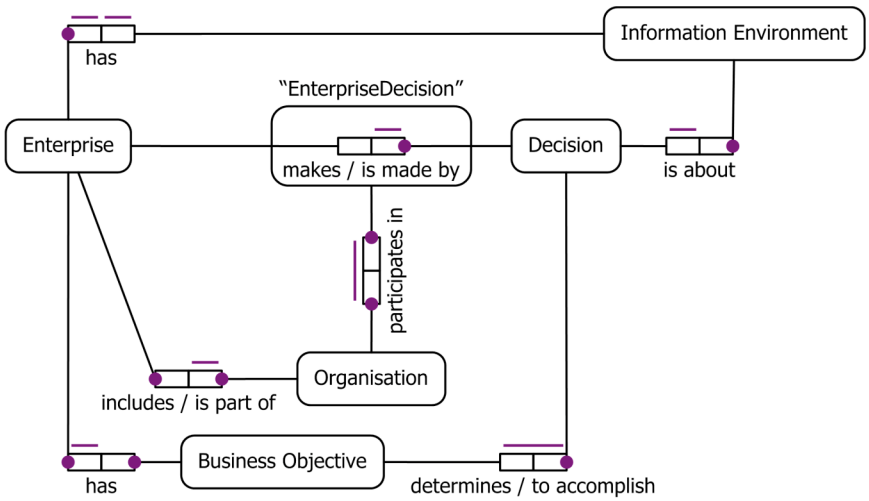
Principle #	**The Name Of The Principle**
Statement	The original text of the statement as defined in the TOGAF document. This is simply for reference; it shows what the fundamental rule is of the principle, in natural language. This should be unambiguous according to TOGAF, but they do not really succeed at that because of the inherent weaknesses of natural language.
Rationale	The original text of the rationale as defined in the TOGAF document. This should show the reason behind the principle in the form of business benefits which are obtained by adhering to this principle. This is the rationale in the words of the architect, not the rationale of the person(s) who give(s) meaning to the architecture.
Assumptions	Here the assumptions made while interpreting the statement and rationale will be included in an ordered numerical list for easy (cross) reference. This is the rationale of those who give meaning to the architecture by interpreting it and extracting the meaning from the information.
ORM Schema	This is where the graphical ORM schema is inserted.
Verbalization	This is where we include the relevant verbalization of the fact types and entity types. This shows in a semi-formalized natural language what the structure of the schema is meant to represent by means of NORMA generated sentences.
Issues	Issues that need to be solved in making the formalization of the principle specific enough to make it imposable.

The resulting filled in templates are listed on the following pages.

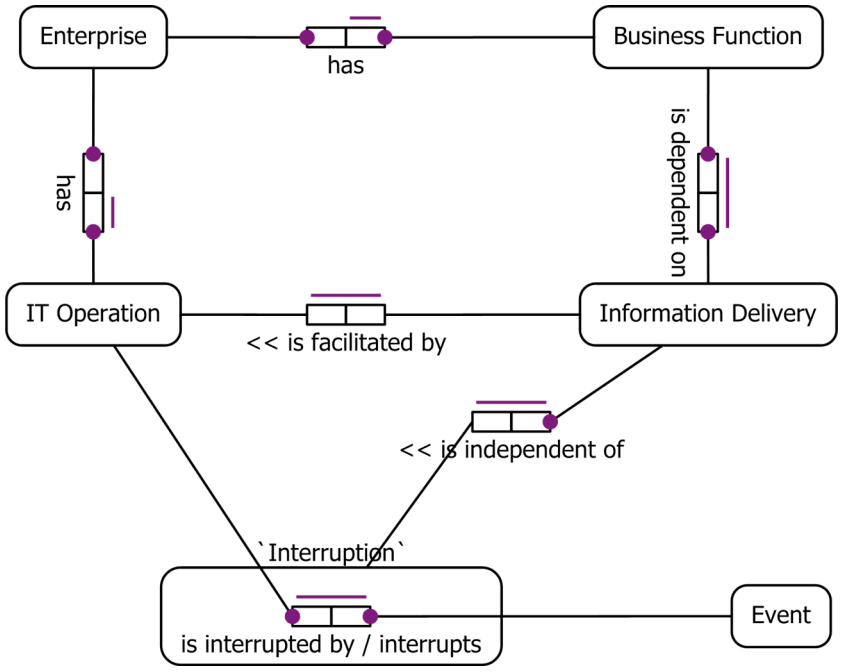
1	**Primacy of Principles**
Statement	These principles of information management apply to all organizations within the enterprise.
Rationale	The only way we can provide a consistent and measurable level of quality information to decision makers is if all organizations abide by the principles.
Assumptions	A1: Organizations are part of an enterprise. A2: An enterprise consists of more than one organization. A3: These principles: all defined principles for the current enterprise (in this case the nineteen principles that follow). A4: The principles apply to both the enterprise and its organizations.
ORM Schema	
Verbalization	Enterprise includes Organisation. **Each** Enterprise includes **some** Organisation. **Each** Organisation is part of **exactly one** Enterprise. **It is possible that the same** Enterprise includes **more than one** Organisation. Enterprise is ruled by IM Principle. **It is possible that more than one** Enterprise is ruled by **the same** IM Principle **and that more than one** IM Principle applies to **the same** Enterprise. **Each** Enterprise, IM Principle **combination occurs at most once in the population of** Enterprise is ruled by IM Principle. **Each** Enterprise is ruled by **some** IM Principle. **Each** IM Principle applies to **some** Enterprise. Organisation is ruled by IM Principle. **It is possible that more than one** Organisation is ruled by **the same** IM Principle **and that more than one** IM Principle applies to **the same** Organisation. **Each** Organisation, IM Principle **combination occurs at most once in the population of** Organisation is ruled by IM Principle. **Each** Organisation is ruled by **some** IM Principle. **Each** IM Principle applies to **some** Organisation. **Each** IM Principle **that** applies to **some** Enterprise **which** includes **some** Organisation, applies to **that** Organisation **as well.**
Issues	What does it mean for a principle to “apply”?

2	**Maximize Benefit to the Enterprise**
Statement	Information management decisions are made to provide maximum benefit to the Enterprise as a whole.
Rationale	This principle embodies "Service above self." Decisions made from an Enterprise-wide perspective have greater long term value than decisions made from any particular organizational perspective. Maximum return on investment requires information management decisions to adhere to Enterprise-wide drivers and priorities. No minority group will detract from the benefit of the whole. However, this principle will not preclude any minority group from getting its job done.
Assumptions	A1: Decisions can be made from the perspective of an enterprise (high level) or from the perspective of one of the organizations within the enterprise. A2: Decisions made from an organization perspective always must be compatible with decisions made from the enterprise perspective. A3: Every decision has benefit to the enterprise, independent of its origin.
ORM Schema	
Verbalization	Organisation is part of Enterprise. **Each** Organisation is part of **exactly one** Enterprise. **Each** Enterprise includes **some** Organisation. **It is possible that the same** Enterprise includes **more than one** Organisation. Organisation has OrganisationPerspective. **For each** OrganisationPerspective, **exactly one** Organisation has **that** OrganisationPerspective. **Each** Organisation has **some** OrganisationPerspective. **It is possible that the same** Organisation has **more than one** OrganisationPerspective. Organisation has benefit from DecisionMadeFromOrganisationPerspective. **Each** DecisionMadeFromOrganisationPerspective benefits **exactly one** Organisation. **It is possible that the same** Organisation has benefit from **more than one** DecisionMade-

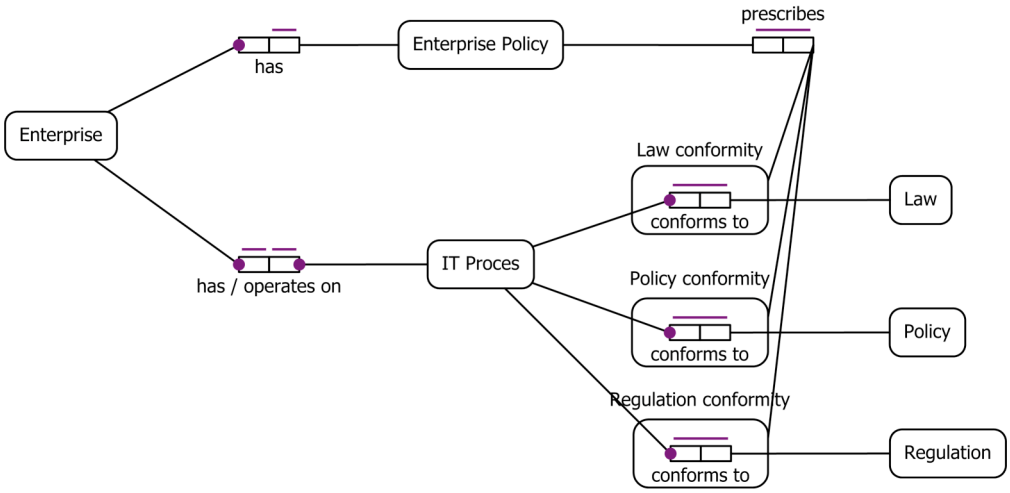
Verbalization	Organisation is part of Enterprise. **Each** Organisation is part of **exactly one** Enterprise. **Each** Enterprise includes **some** Organisation. **It is possible that the same** Enterprise includes **more than one** Organisation. Organisation has OrganisationPerspective. **For each** OrganisationPerspective, **exactly one** Organisation has **that** OrganisationPerspective. **Each** Organisation has **some** OrganisationPerspective. **It is possible that the same** Organisation has **more than one** OrganisationPerspective. Organisation has benefit from DecisionMadeFromOrganisationPerspective. **Each** DecisionMadeFromOrganisationPerspective benefits **exactly one** Organisation. **It is possible that the same** Organisation has benefit from **more than one** DecisionMadeFromOrganisationPerspective. Decision made from OrganisationPerspective. **Each** Decision made from **exactly one** OrganisationPerspective. **It is possible that more than one** Decision made from **the same** OrganisationPerspective. DecisionMadeFromOrganisationPerspective is compatible with DecisionMadeFromEnterprisePerspective. **It is possible that more than one** DecisionMadeFromOrganisationPerspective is compatible with **the same** DecisionMadeFromEnterprisePerspective **and that the same** DecisionMadeFromOrganisationPerspective is compatible with **more than one** DecisionMadeFromEnterprisePerspective. **Each** DecisionMadeFromOrganisationPerspective, DecisionMadeFromEnterprisePerspective **combination occurs at most once in the population of** DecisionMadeFromOrganisationPerspective is compatible with DecisionMadeFromEnterprisePerspective. **Each** DecisionMadeFromOrganisationPerspective is compatible with **some** DecisionMadeFromEnterprisePerspective. Enterprise has EnterprisePerspective. **Each** Enterprise has **some** EnterprisePerspective. **For each** EnterprisePerspective, **exactly one** Enterprise has **that** EnterprisePerspective. **It is possible that the same** Enterprise has **more than one** EnterprisePerspective. Enterprise has benefit from DecisionMadeFromEnterprisePerspective. **Each** DecisionMadeFromEnterprisePerspective benefits **exactly one** Enterprise. **It is possible that the same** Enterprise has benefit from **more than one** DecisionMadeFromEnterprisePerspective. Decision made from EnterprisePerspective. **Each** Decision made from **exactly one** EnterprisePerspective. **It is possible that more than one** Decision made from **the same** EnterprisePerspective.
Issues	How is "maximum benefit" defined, and how is the benefit to be measured at all?

3	**Information Management is Everybody's Business**
Statement	All organizations in the Enterprise participate in information management decisions needed to accomplish business objectives.
Rationale	Information users are the key stakeholders, or customers, in the application of technology to address a business need. In order to ensure information management is aligned with the business, all organizations in the Enterprise must be involved in all aspects of the information environment. The business experts from across the Enterprise and the technical staff responsible for developing and sustaining the information environment need to come together as a team to jointly define the goals and objectives of information technology.
Assumptions	A1: The concept Decision is about Information Management Decisions.
ORM Schema	
Verbalization	Enterprise has Business Objective. **For each** Business Objective, **exactly one** Enterprise has **that** Business Objective. **Each** Enterprise has **some** Business Objective. **It is possible that the same** Enterprise has **more than one** Business Objective. Enterprise has Information Environment. **Each** Enterprise has **exactly one** Information Environment. **For each** Information Environment, **at most one** Enterprise has **that** Information Environment. Enterprise includes Organisation. **Each** Enterprise includes **some** Organisation. **Each** Organisation is part of **exactly one** Enterprise. **It is possible that the same** Enterprise includes **more than one** Organisation. Business Objective determines Decision. **It is possible that more than one** Business Objective determines **the same** Decision **and that more than one** Decision to accomplish **the same** Business Objective. **Each** Decision, Business Objective **combination occurs at most once in the population of** Business Objective determines Decision. **Each** Decision to accomplish **some** Business Objective.

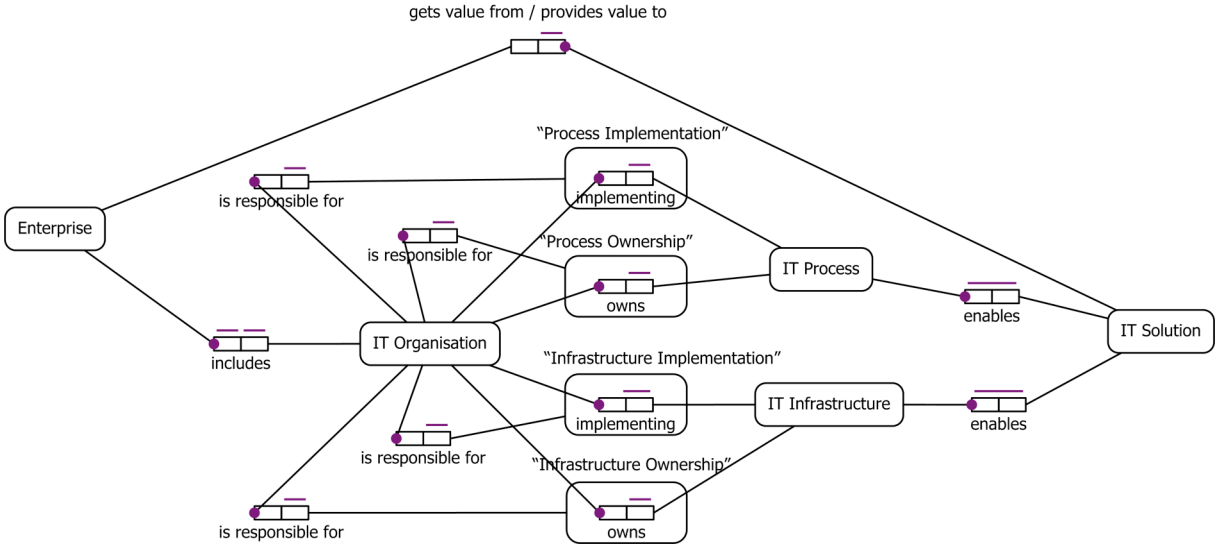
	Enterprise makes Decision. **Each** Decision is made by **exactly one** Enterprise. **It is possible that the same** Enterprise makes **more than one** Decision. Organisation participates in EnterpriseDecision. **It is possible that more than one** Organisation participates in **the same** EnterpriseDecision **and that the same** Organisation participates in **more than one** EnterpriseDecision. **Each** EnterpriseDecision, Organisation **combination occurs at most once in the population of** Organisation participates in EnterpriseDecision. **Each** Organisation participates in **some** EnterpriseDecision. **For each** EnterpriseDecision, **some** Organisation participates in **that** EnterpriseDecision. Decision is about Information Environment. **Each** Decision is about **at most one** Information Environment. **For each** Information Environment, **some** Decision is about **that** Information Environment. **It is possible that more than one** Decision is about **the same** Information Environment. **If an** Organisation **which** is part of **some** Enterprise **which** makes **some** Decision **then that** Organisation participates in **some** EnterpriseDecision involving **that** Enterprise
Issue	What does it mean to "participate"?

4	**Business Continuity**
Statement	Enterprise operations are maintained in spite of system interruptions.
Rationale	As system operations become more pervasive, we become more dependent on them, therefore, we must consider the reliability of such systems throughout their design and use. Business premises throughout the Enterprise must be provided the capability to continue their business functions regardless of external events. Hardware failure, natural disasters, and data corruption should not be allowed to disrupt or stop Enterprise activities. The Enterprise business functions must be capable of operating on alternative information delivery mechanisms.
Assumptions	A1: Business Functions need Information, and implicitly their delivery. A2: system interoperations are mechanisms for Information Delivery. A3: The Business functions should be dependent on other delivery mechanisms then system operations. A4: An event is something that causes an interruption like a natural disaster or bug.
ORM Schema	
Verbalization	Enterprise has Business Function. **Each** Enterprise has **some** Business Function. **For each** Business Function, **exactly one** Enterprise has **that** Business Function. **It is possible that the same** Enterprise has **more than one** Business Function. Enterprise has IT Operation. **Each** Enterprise has **some** IT Operation. **For each** IT Operation, **exactly one** Enterprise has **that** IT Operation. **It is possible that the same** Enterprise has **more than one** IT Operation. Business Function is dependent on Information Delivery. **It is possible that more than one** Business Function is dependent on **the same** Information Delivery

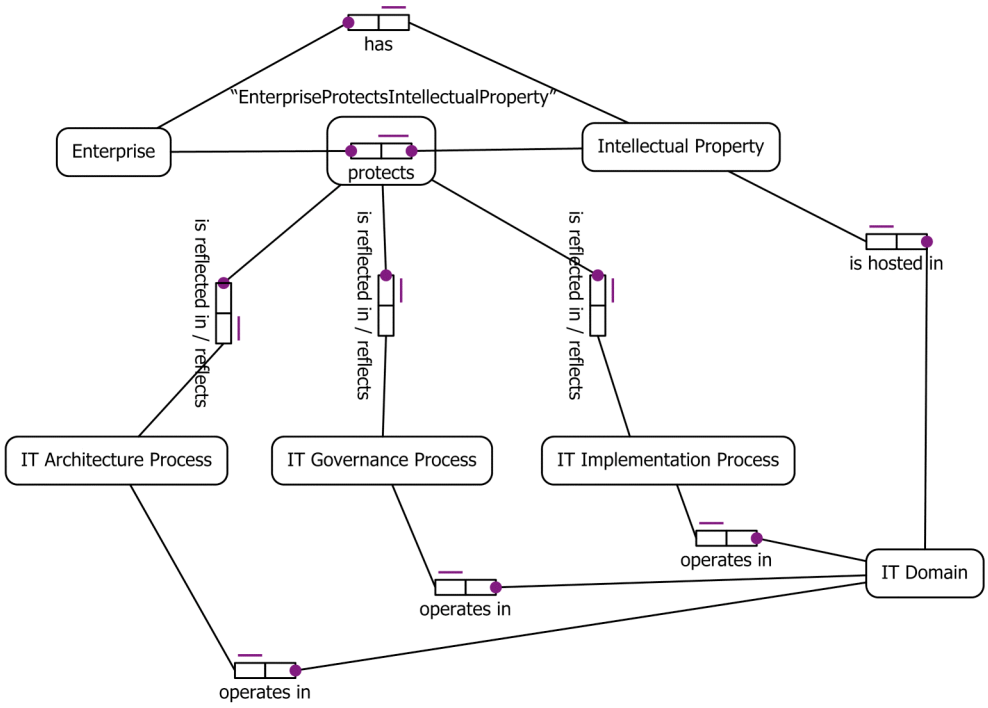
	and that the same Business Function is dependent on **more than one** Information Delivery. **Each** Business Function, Information Delivery **combination occurs at most once in the population of** Business Function is dependent on Information Delivery. **Each** Business Function is dependent on **some** Information Delivery. **For each** Information Delivery, **some** Business Function is dependent on **that** Information Delivery. Information Delivery facilitated by IT Operation. **It is possible that the same** Information Delivery facilitated by **more than one** IT Operation **and that more than one** Information Delivery facilitated by **the same** IT Operation. **Each** Information Delivery, IT Operation **combination occurs at most once in the population of** Information Delivery facilitated by IT Operation. Information Delivery is independent of Interruption. **It is possible that the same** Information Delivery is independent of **more than one** Interruption **and that more than one** Information Delivery is independent of **the same** Interruption. **Each** Interruption, Information Delivery **combination occurs at most once in the population of** Information Delivery is independent of Interruption. **Each** Information Delivery is independent of **some** Interruption. IT Operation is interrupted by Event. **It is possible that more than one** IT Operation is interrupted by **the same** Event **and that more than one** Event interrupts **the same** IT Operation. **Each** Event, IT Operation **combination occurs at most once in the population of** IT Operation is interrupted by Event. **Each** IT Operation is interrupted by **some** Event. **Each** Event interrupts **some** IT Operation.
Issues	How to determine if an event is an interruption?

6	Compliance with Law
Statement	Enterprise information management processes comply with all relevant laws, policies, and regulations.
Rationale	Enterprise policy is to abide by laws, policies, and regulations. This will not preclude business process improvements that lead to changes in policies and regulations.
Assumptions	A1: An enterprise has several policies for different purposes, like for example security-policies and customer-relationship policies.
ORM Schema	
Verbalization	Enterprise has Enterprise Policy. **Each** Enterprise has **some** Enterprise Policy. **For each** Enterprise Policy, **at most one** Enterprise has **that** Enterprise Policy. **It is possible that the same** Enterprise has **more than one** Enterprise Policy. Enterprise has IT Process. **Each** Enterprise has **exactly one** IT Process. **Each** IT Process operates in **exactly one** Enterprise. Enterprise Policy prescribes ITProcessConformsToLaw. **It is possible that more than one** Enterprise Policy prescribes **the same** ITProcessConform-sToLaw **and that the same** Enterprise Policy prescribes **more than one** ITProcessConformsToLaw. **Each** ITProcessConformsToLaw, Enterprise Policy **combination occurs at most once in the population of** Enterprise Policy prescribes ITProcessConformsToLaw. Enterprise Policy prescribes ITProcessConformsToRegulation. **It is possible that more than one** Enterprise Policy prescribes **the same** ITProcessConform-sToRegulation **and that the same** Enterprise Policy prescribes **more than one** ITProcessConformsToRegula-tion. **Each** Enterprise Policy, ITProcessConformsToRegulation **combination occurs at most once in the population of** Enterprise Policy prescribes ITProcessConformsToRegulation. Enterprise Policy prescribes ITProcessConformsToPolicy. **It is possible that more than one** Enterprise Policy prescribes **the same** ITProcessConform-

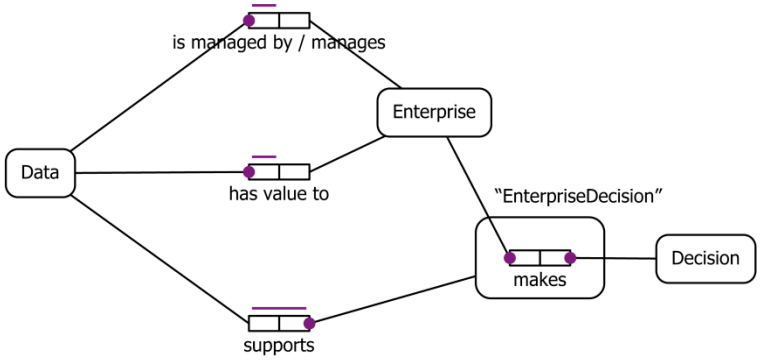
	sToPolicy **and that the same** Enterprise Policy prescribes **more than one** ITProcessConformsToPolicy. **Each** ITProcessConformsToPolicy, Enterprise Policy **combination occurs at most once in the population of** Enterprise Policy prescribes ITProcessConformsToPolicy. IT Process conforms to Law. **It is possible that more than one** IT Process conforms to **the same** Law **and that the same** IT Process conforms to **more than one** Law. **Each** IT Process, Law **combination occurs at most once in the population of** IT Process conforms to Law. **Each** IT Process conforms to **some** Law. IT Process conforms to Regulation. **It is possible that more than one** IT Process conforms to **the same** Regulation **and that the same** IT Process conforms to **more than one** Regulation. **Each** IT Process, Regulation **combination occurs at most once in the population of** IT Process conforms to Regulation. **Each** IT Process conforms to **some** Regulation. IT Process conforms to Policy. **It is possible that more than one** IT Process conforms to **the same** Policy **and that the same** IT Process conforms to **more than one** Policy. **Each** IT Process, Policy **combination occurs at most once in the population of** IT Process conforms to Policy. **Each** IT Process conforms to **some** Policy.
Issue	How to determine/evaluate compliance?

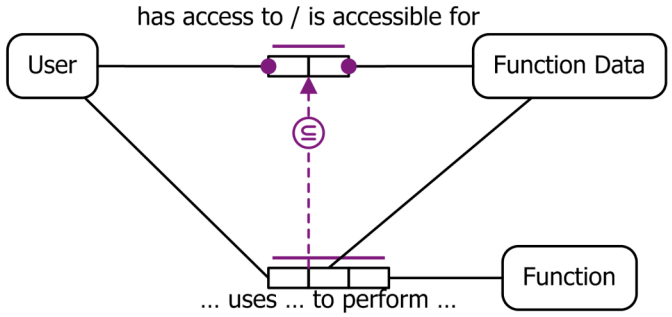
7	IT Responsibility
Statement	The IT organization is responsible for owning and implementing IT processes and infrastructure that enable solutions to meet user defined requirements for functionality, service levels, cost, and delivery timing.
Rationale	Effectively align expectations with capabilities and costs so that all projects are cost effective. Efficient and effective solutions have reasonable costs and clear benefits.
Assumptions	A1: All IT processes enable a solution. A2: All elements of IT infrastructure enable a solution. A3: In this scope we assume solutions are IT solutions.
ORM Schema	
Verbalization	Enterprise includes IT Organisation. **Each** Enterprise includes **exactly one** IT Organisation. **For each** IT Organisation, **at most one** Enterprise includes **that** IT Organisation. Enterprise gets value from IT Solution. **Each** IT Solution provides value to **exactly one** Enterprise. **It is possible that the same** Enterprise gets value from **more than one** IT Solution. Enterprise gets value from IT Solution. **Each** IT Solution provides value to **exactly one** Enterprise. **It is possible that the same** Enterprise gets value from **more than one** IT Solution. IT Organisation is responsible for Process Ownership. **Each** IT Organisation is responsible for **some** Process Ownership. **For each** Process Ownership, **at most one** IT Organisation is responsible for **that** Process Ownership. **It is possible that the same** IT Organisation is responsible for **more than one** Process Ownership. IT Organisation is responsible for Infrastructure Implementation. **Each** IT Organisation is responsible for **some** Infrastructure Implementation. **For each** Infrastructure Implementation, **at most one** IT Organisation is responsible for **that**

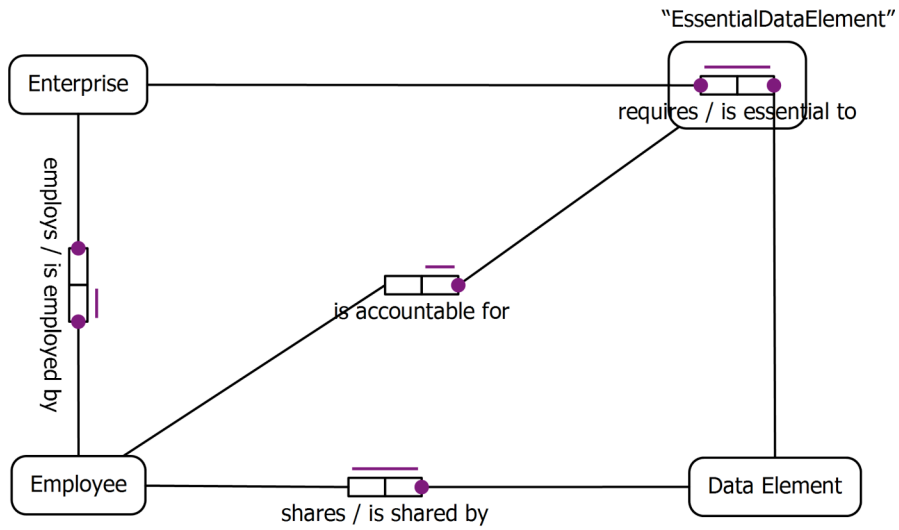
	Infrastructure Implementation. **It is possible that the same** IT Organisation is responsible for **more than one** Infrastructure Implementation. IT Organisation is responsible for Infrastructure Ownership. **Each** IT Organisation is responsible for **some** Infrastructure Ownership. **For each** Infrastructure Ownership, **at most one** IT Organisation is responsible for **that** Infrastructure Ownership. **It is possible that the same** IT Organisation is responsible for **more than one** Infrastructure Ownership. IT Organisation implementing IT Process. **Each** IT Organisation implementing **some** IT Process. **For each** IT Process, **at most one** IT Organisation implementing **that** IT Process. **It is possible that the same** IT Organisation implementing **more than one** IT Process. IT Organisation owns IT Process. **Each** IT Organisation owns **some** IT Process. **For each** IT Process, **at most one** IT Organisation owns **that** IT Process. **It is possible that the same** IT Organisation owns **more than one** IT Process. IT Organisation implementing IT Infrastructure. **Each** IT Organisation implementing **some** IT Infrastructure. **For each** IT Infrastructure, **at most one** IT Organisation implementing **that** IT Infrastructure. **It is possible that the same** IT Organisation implementing **more than one** IT Infrastructure. IT Organisation owns IT Infrastructure. **Each** IT Organisation owns **some** IT Infrastructure. **For each** IT Infrastructure, **at most one** IT Organisation owns **that** IT Infrastructure. **It is possible that the same** IT Organisation owns **more than one** IT Infrastructure. IT Process enables IT Solution. **It is possible that more than one** IT Process enables **the same** IT Solution **and that the same** IT Process enables **more than one** IT Solution. **Each** IT Solution, IT Process **combination occurs at most once in the population of** IT Process enables IT Solution. **Each** IT Process enables **some** IT Solution. IT Infrastructure enables IT Solution. **It is possible that more than one** IT Infrastructure enables **the same** IT Solution **and that the same** IT Infrastructure enables **more than one** IT Solution. **Each** IT Infrastructure, IT Solution **combination occurs at most once in the population of** IT Infrastructure enables IT Solution. **Each** IT Infrastructure enables **some** IT Solution.

8	**Protection of Intellectual Property**
Statement	The enterprise's IP must be protected. This protection must be reflected in the IT Architecture, Implementation, and Governance processes.
Rationale	A major part of an enterprise's Intellectual Property is hosted in the IT domain.
Assumptions	A1: Every enterprise has some Intellectual Property. A2: IP protection is reflected in three processes, being: - IT Architecture process - IT Implementation process - IT Governance process A3: Every enterprise has an IT domain.
ORM Schema	
Verbalization	Enterprise has Intellectual Property. **Each** Enterprise has **some** Intellectual Property. **For each** Intellectual Property, **at most one** Enterprise has **that** Intellectual Property. **It is possible that the same** Enterprise has **more than one** Intellectual Property. Enterprise protects Intellectual Property. **Each** Enterprise protects **some** Intellectual Property. **For each** Intellectual Property, **exactly one** Enterprise protects **that** Intellectual Property. **It is possible that the same** Enterprise protects **more than one** Intellectual Property. EnterpriseProtectsIntellectualProperty is reflected in IT Architecture Process. **Each** EnterpriseProtectsIntellectualProperty is reflected in **some** IT Architecture Process. **Each** IT Architecture Process reflects **at most one** EnterpriseProtectsIntellectualProperty. **It is possible that the same** EnterpriseProtectsIntellectualProperty is reflected in **more than**

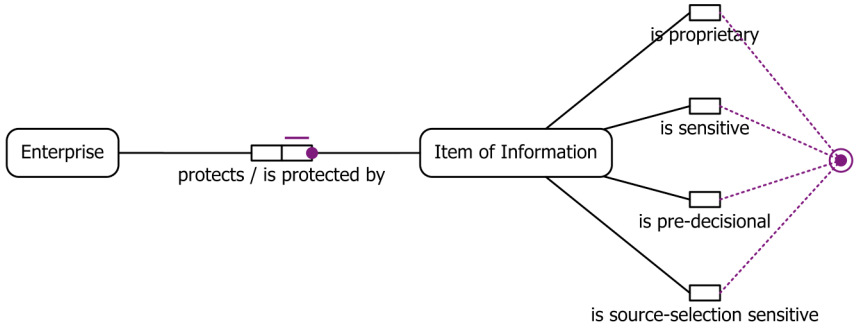
	one IT Architecture Process. EnterpriseProtectsIntellectualProperty is reflected in IT Governance Process. **Each** EnterpriseProtectsIntellectualProperty is reflected in **exactly one** IT Governance Process. **It is possible that the same** IT Governance Process reflects **more than one** EnterpriseProtectsIntellectualProperty. EnterpriseProtectsIntellectualProperty is reflected in IT Implementation Process. **Each** EnterpriseProtectsIntellectualProperty is reflected in **exactly one** IT Implementation Process. **It is possible that the same** IT Implementation Process reflects **more than one** EnterpriseProtectsIntellectualProperty. Intellectual Property is hosted in IT Domain. **Each** Intellectual Property is hosted in **at most one** IT Domain. **For each** IT Domain, **some** Intellectual Property is hosted in **that** IT Domain. **It is possible that more than one** Intellectual Property is hosted in **the same** IT Domain. IT Implementation Process operates in IT Domain. **Each** IT Implementation Process operates in **at most one** IT Domain. **For each** IT Domain, **some** IT Implementation Process operates in **that** IT Domain. **It is possible that more than one** IT Implementation Process operates in **the same** IT Domain. IT Governance Process operates in IT Domain. **Each** IT Governance Process operates in **at most one** IT Domain. **For each** IT Domain, **some** IT Governance Process operates in **that** IT Domain. **It is possible that more than one** IT Governance Process operates in **the same** IT Domain. IT Architecture Process operates in IT Domain. **Each** IT Architecture Process operates in **at most one** IT Domain. **For each** IT Domain, **some** IT Architecture Process operates in **that** IT Domain. **It is possible that more than one** IT Architecture Process operates in **the same** IT Domain.
Issues	How to check whether the processes are “protected”?

9	**Data is an Asset**
Statement	Data is an asset that has value to the Enterprise and is managed accordingly.
Rationale	Data is a valuable corporate resource; it has real, measurable value. In simple terms, the purpose of data is to aid decision-making. Accurate, timely data is critical to accurate, timely decisions. Most corporate assets are carefully managed, and data is no exception. Data is the foundation of our decision making, so we must also carefully manage data to assure that we know where it is, can rely upon its accuracy, and can obtain it when and where we need it.
Assumptions	-
ORM Schema	
Verbalization	Data is managed by Enterprise. **Each** Data is managed by **exactly one** Enterprise. **It is possible that the same** Enterprise manages **more than one** Data. Data has value to Enterprise. **Each** Data has value to **exactly one** Enterprise. **It is possible that more than one** Data has value to **the same** Enterprise. Data supports EnterpriseDecision. **It is possible that more than one** Data supports **the same** EnterpriseDecision **and that the same** Data supports **more than one** EnterpriseDecision. **Each** EnterpriseDecision, Data **combination occurs at most once in the population of** Data supports EnterpriseDecision. **For each** EnterpriseDecision, **some** Data supports **that** EnterpriseDecision. Enterprise makes Decision. **Each** Enterprise makes **some** Decision. **For each** Decision, **exactly one** Enterprise makes **that** Decision. **It is possible that the same** Enterprise makes **more than one** Decision.
Issue	How should a valuable asset be treated?

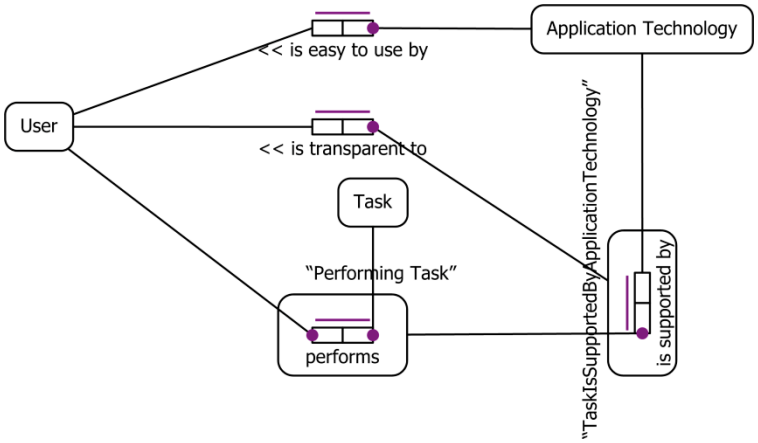
11	Data is Accessible
Statement	Data is accessible for users to perform their functions.
Rationale	Wide access to data leads to efficiency and effectiveness in decision-making, and affords timely response to information requests and service delivery. Using information must be considered from an Enterprise perspective to allow access by a wide variety of users. Staff time is saved and consistency of data is improved.
Assumptions	A1: A user must have at least access to the data it uses to perform functions.
ORM Schema	
Verbalization	User has access to Function Data. **It is possible that more than one** User has access to **the same** Function Data **and that more than one** Function Data is accessible for **the same** User. **Each** User, Function Data **combination occurs at most once in the population of** User has access to Function Data. **Each** User has access to **some** Function Data. **Each** Function Data is accessible for **some** User. User uses Function Data to perform Function. **It is possible that more than one** User uses **the same** Function Data to perform **the same** Function **and that the same** User uses **more than one** Function Data to perform **the same** Function **and that the same** User uses **the same** Function Data to perform **more than one** Function. **Each** User, Function Data, Function **combination occurs at most once in the population of** User uses Function Data to perform Function. User, Function Data, Function **combination occurs at most once in the population of** User uses Function Data to perform Function. **Each** Function data **which** is used by **some** User to perform **some** Function **must also be** accessible for **that** User
Issues	What does it mean for data to be accessible for a user? This may touch on issues such as timeliness, location, media, file format, etc.

12	Data Trustee
Statement	Each data element has a trustee accountable for data quality.
Rationale	One of the benefits of an architected environment is the ability to share data (e.g., text, video, sound, etc.) across the Enterprise. As the degree of data sharing grows and business units rely upon common information, it becomes essential that only the data trustee make decisions about the content of data. Since data can lose its integrity when it is entered multiple times, the data trustee will have sole responsibility for data entry, which eliminates redundant human effort and data storage resources. (Note that a trustee is different from steward - trustee is responsible for accuracy and currency of the data while responsibilities of a steward may be broader and include data standardization and definition tasks.)
Assumptions	A1: Data is shared by employees, as shown in principle 10. A2: There is no essential data element without a data trustee.
ORM Schema	
Verbalization	Enterprise employs Employee. **Each** Enterprise employs **some** Employee. **Each** Employee is employed by **exactly one** Enterprise. **It is possible that the same** Enterprise employs **more than one** Employee. Employee shares Data Element. **It is possible that more than one** Employee shares **the same** Data Element **and that more than one** Data Element is shared by **the same** Employee. **Each** Employee, Data Element **combination occurs at most once in the population of** Employee shares Data Element. **Each** Data Element is shared by **some** Employee. Enterprise requires Data Element. **It is possible that more than one** Enterprise requires **the same** Data Element **and that more than one** Data Element is essential to **the same** Enterprise. **Each** Enterprise, Data Element **combination occurs at most once in the population of** Enterprise requires Data Element. **Each** Enterprise requires **some** Data Element.

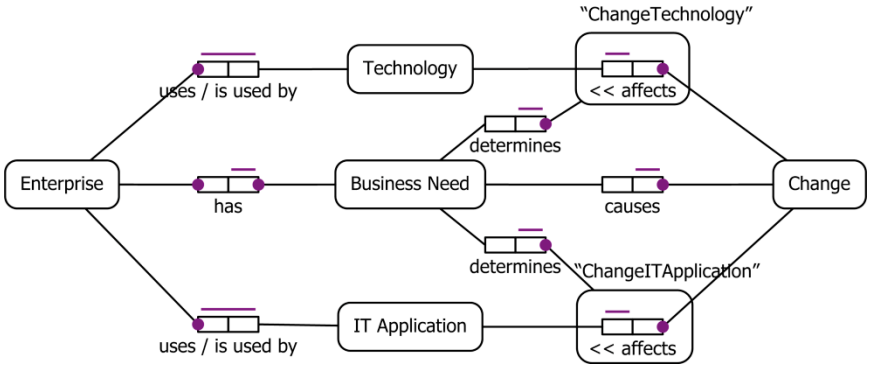
	Each Data Element is essential to **some** Enterprise. EssentialDataElement **is an entity type**. EssentialDataElement **objectifies** "Enterprise requires Data Element". Employee is accountable for EssentialDataElement. **For each** EssentialDataElement, **exactly one** Employee is accountable for **that** Essential-DataElement. **It is possible that the same** Employee is accountable for **more than one** EssentialDataEle-ment.
Issues	When is data shared? How does one define this?

14	Data Security
Statement	Data is protected from unauthorized use and disclosure. In addition to the traditional aspects of national security classification, this includes, but is not limited to, protection of pre-decisional, sensitive, source selection sensitive and proprietary information.
Rationale	Open sharing of information and the release of information via relevant legislation must be balanced against the need to restrict the availability of classified, proprietary, and sensitive information.
Assumptions	A1: Data and Information are used as synonyms, we chose Information.
ORM Schema	
Verbalization	Enterprise protects Item of Information. **Each** Item of Information is protected by **exactly one** Enterprise. **It is possible that the same** Enterprise protects **more than one** Item of Information. **Each** Item of Information is sensitive **or** is proprietary **or** is pre-decisional **or** is source selection sensitive.

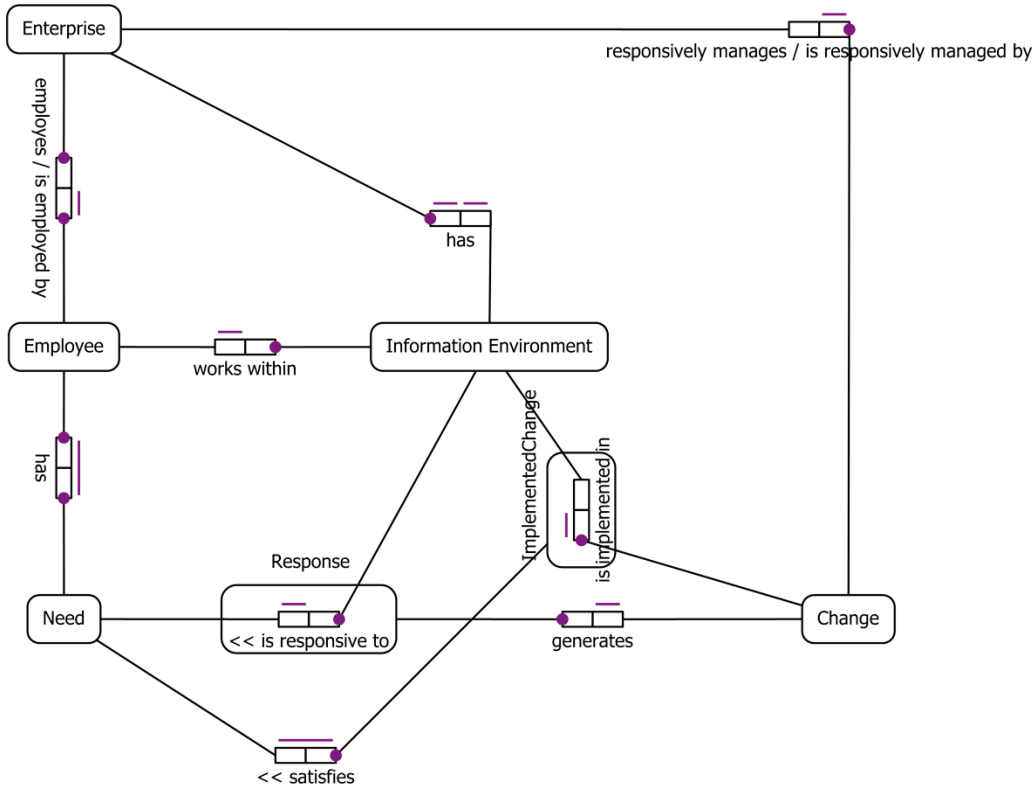
15	**Technology Independence**
Statement	Applications are independent of specific technology choices and therefore can operate on a variety of technology platforms.
Rationale	Independence of applications from the underlying technology allows applications to be developed, upgraded, and operated in the most cost-effective and timely way. Otherwise, technology, which is subject to continual obsolescence and vendor dependence, becomes the driver rather than the user requirements themselves. Realizing that every decision made respect to information technology makes us dependent on that technology, the intent of this principle is to ensure that applications software is not dependent on specific hardware and operating systems software.
Assumptions	-
ORM Schema	
Verbalization	IT Application is independent of Technology. **Each** IT Application is independent of **exactly one** Technology. **It is possible that more than one** IT Application is independent of **the same** Technology. Technology is dependent on IT Platform. **Each** Technology is dependent on **some** IT Platform. **For each** IT Platform, **at most one** Technology is dependent on **that** IT Platform. **It is possible that the same** Technology is dependent on **more than one** IT Platform.
Issues	What does it mean to be able top operate on a variety of platforms?

16	Ease of Use
Statement	Applications are easy to use. The underlying technology is transparent to users, so they can concentrate on tasks at hand.
Rationale	The more a user has to understand the underlying technology the less productive that user is. Ease of use is a positive incentive for use of applications. It encourages users to work within the integrated information environment instead of developing isolated systems to accomplish the task outside of the Enterprise's integrated information environment. Most of the knowledge required to operate one system will be similar to others. Training is kept to a minimum, and the risk of using a system improperly is low.
Assumptions	A1: That 'users can concentrate on the task at hand' is the result of the rule to be defined here, the desired effect. This though does not add any value to the model, because the model is about the rule itself. For this reason, the latter part of the statement is not modelled. A2: The principle is clearly about ease of use. But also stated is that: technology underlying the application is transparent. The word 'is' in this last sentence suggests a fact, but this can be interpreted as a rule: technology underlying the application must be transparent. A3: If application technology transparency is in fact a rule, it can be connected to ease of use in the following way: application technology transparency *helps* in some way the ease of use. It can be seen as some subcategory, not a separate rule. This complies with the statement name: ease of use.
ORM Schema	
Verbalization	Application Technology is easy to use by User. **It is possible that the same** Application Technology is easy to use by **more than one** User **and that more than one** Application Technology is easy to use by **the same** User. **Each** User, Application Technology **combination occurs at most once in the population of** Application Technology is easy to use by User. **Each** Application Technology is easy to use by **some** User. TaskIsSupportedByApplicationTechnology is transparent to User. **It is possible that the same** TaskIsSupportedByApplicationTechnology is transparent to **more than one** User **and that more than one** TaskIsSupportedByApplicationTechnology is transparent to **the same** User.

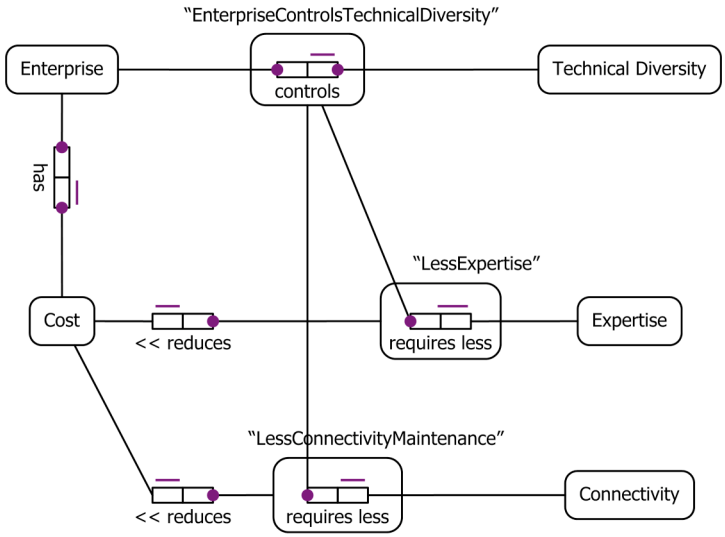
	Each TaskIsSupportedByApplicationTechnology, User **combination occurs at most once in the population of** TaskIsSupportedByApplicationTechnology is transparent to User. **For each** User, **some** TaskIsSupportedByApplicationTechnology is transparent to **that** User. **Each** TaskIsSupportedByApplicationTechnology is transparent to **some** User. Performing Task is supported by Application Technology. **It is possible that more than one** Performing Task is supported by **the same** Application Technology **and that the same** Performing Task is supported by **more than one** Application Technology. **Each** Application Technology, Performing Task **combination occurs at most once in the population of** Performing Task is supported by Application Technology. **Each** Performing Task is supported by **some** Application Technology. User performs Task. **It is possible that more than one** User performs **the same** Task **and that the same** User performs **more than one** Task. **Each** User, Task **combination occurs at most once in the population of** User performs Task. **Each** User performs **some** Task. **For each** Task, **some** User performs **that** Task.
Issues	How to measure ease of use and transparency.

17	Requirements-Based Change
Statement	Only in response to business needs are changes to applications and technology made.
Rationale	This principle will foster an atmosphere where the information environment changes in response to the needs of the business, rather than having the business change in response to information technology changes. This is to ensure that the purpose of the information support -- the transaction of business -- is the basis for any proposed change. Unintended effects on business due to information technology changes will be minimized. A change in technology may provide an opportunity to improve the business process and hence, change business needs.
Assumptions	A1: A change is an actual existence and therefore always affects technology and IT application. A2: IT Application & Technology should not be interpreted as singleton. A3: A change is about an IT change.
ORM Schema	
Verbalization	Enterprise uses Technology. **It is possible that more than one** Enterprise uses **the same** Technology **and that more than one** Technology is used by **the same** Enterprise. **Each** Enterprise, Technology **combination occurs at most once in the population of** Enterprise uses Technology. **Each** Enterprise uses **some** Technology. Enterprise has Business Need. **Each** Enterprise has **some** Business Need. **For each** Business Need, **exactly one** Enterprise has **that** Business Need. **It is possible that the same** Enterprise has **more than one** Business Need. Enterprise uses IT Application. **It is possible that more than one** Enterprise uses **the same** IT Application **and that more than one** IT Application is used by **the same** Enterprise. **Each** Enterprise, IT Application **combination occurs at most once in the population of** Enterprise uses IT Application. **Each** Enterprise uses **some** IT Application. Business Need determines ChangeTechnology. **For each** ChangeTechnology, **exactly one** Business Need determines **that** ChangeTechnology. **It is possible that the same** Business Need determines **more than one** ChangeTechnology. Business Need determines ChangeITApplication.

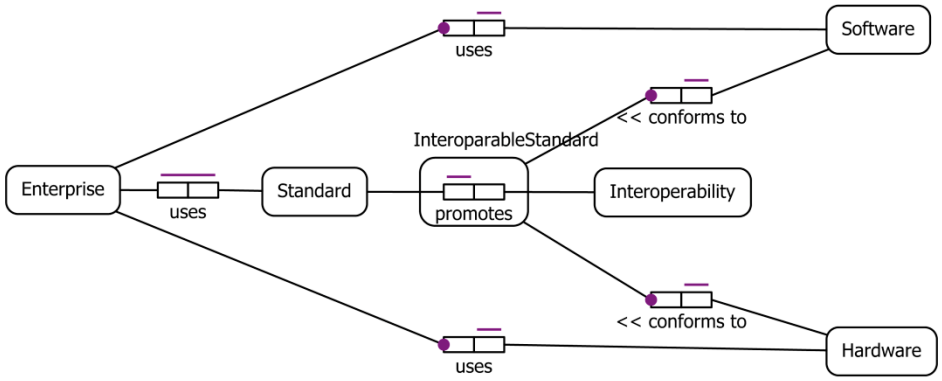
	For each ChangeITApplication, **exactly one** Business Need determines **that** ChangeITApplication. **It is possible that the same** Business Need determines **more than one** ChangeITApplication. Business Need causes Change. **For each** Change, **exactly one** Business Need causes **that** Change. **It is possible that the same** Business Need causes **more than one** Change. Change affects Technology. **For each** Technology, **at most one** Change affects **that** Technology. **Each** Change affects **some** Technology. **It is possible that the same** Change affects **more than one** Technology. Change affects IT Application. **For each** IT Application, **at most one** Change affects **that** IT Application. **Each** Change affects **some** IT Application. **It is possible that the same** Change affects **more than one** IT Application.
Issues	How to determine/predict the impact of changes?

18	Responsive Change Management
Statement	Changes to the Enterprise information environment are implemented in a timely manner.
Rationale	If people are to be expected to work within the Enterprise information environment, that information environment must be responsive to their needs.
Assumptions	-
ORM Schema	
Verbalization	Enterprise responsively manages Change. **Each** Change is responsively managed by **exactly one** Enterprise. **It is possible that the same** Enterprise responsively manages **more than one** Change. Enterprise employs Employee. **Each** Enterprise employs **some** Employee. **Each** Employee is employed by **exactly one** Enterprise. **It is possible that the same** Enterprise employs **more than one** Employee. Enterprise has Information Environment. **Each** Enterprise has **exactly one** Information Environment. **For each** Information Environment, **at most one** Enterprise has **that** Information Environment. Employee works within Information Environment. **Each** Employee works within **at most one** Information Environment. **For each** Information Environment, **some** Employee works within **that** Information Environment.

	It is possible that more than one Employee works within **the same** Information Environment. Employee has Need. **It is possible that more than one** Employee has **the same** Need **and that the same** Employee has **more than one** Need. **Each** Need, Employee **combination occurs at most once in the population of** Employee has Need. **Each** Employee has **some** Need. **For each** Need, **some** Employee has **that** Need. Response generates Change. **Each** Response generates **some** Change. **For each** Change, **at most one** Response generates **that** Change. **It is possible that the same** Response generates **more than one** Change. Change is implemented in Information Environment. **Each** Change is implemented in **exactly one** Information Environment. **It is possible that more than one** Change is implemented in **the same** Information Environment. ImplementedChange satisfies Need. **It is possible that the same** ImplementedChange satisfies **more than one** Need **and that more than one** ImplementedChange satisfies **the same** Need. **Each** ImplementedChange, Need **combination occurs at most once in the population of** ImplementedChange satisfies Need. **Each** ImplementedChange satisfies **some** Need.
Issues	Again. How to gage the impact of changes? How to determine if an implemented change satisfies a need?

19	Control Technical Diversity
Statement	Technological diversity is controlled to minimize the non-trivial cost of maintaining expertise in and connectivity between multiple processing environments.
Rationale	There is a real, non-trivial cost of infrastructure required to support alternative technologies for processing environments. There are further infrastructure costs incurred to keep multiple processor constructs interconnected and maintained. Limiting the number of supported components will simplify maintainability and reduce costs. The business advantages of minimum technical diversity include: standard packaging of components; predictable implementation impact; predictable valuations and returns; redefined testing; utility status; and increased flexibility to accommodate technological advancements. Common technology across the Enterprise brings the benefits of economies of scale to the Enterprise. Technical administration and support costs are better controlled when limited resources can focus on this shared set of technology.
Assumptions	-
ORM Schema	
Verbalization	Enterprise has Cost. **Each** Enterprise has **some** Cost. **For each** Cost, **exactly one** Enterprise has **that** Cost. **It is possible that the same** Enterprise has **more than one** Cost. LessExpertise reduces Cost. **For each** Cost, **at most one** LessExpertise reduces **that** Cost. **Each** LessExpertise reduces **some** Cost. **It is possible that the same** LessExpertise reduces **more than one** Cost. LessConnectivityMaintenance reduces Cost. **For each** Cost, **at most one** LessConnectivityMaintenance reduces **that** Cost. **Each** LessConnectivityMaintenance reduces **some** Cost.

	It is possible that the same LessConnectivityMaintenance reduces **more than one** Cost. Enterprise controls Technical Diversity. **Each** Enterprise controls **some** Technical Diversity. **For each** Technical Diversity, **exactly one** Enterprise controls **that** Technical Diversity. **It is possible that the same** Enterprise controls **more than one** Technical Diversity. EnterpriseControlsTechnicalDiversity requires less Expertise. **Each** EnterpriseControlsTechnicalDiversity requires less **some** Expertise. **For each** Expertise, **at most one** EnterpriseControlsTechnicalDiversity requires less **that** Expertise. **It is possible that the same** EnterpriseControlsTechnicalDiversity requires less **more than one** Expertise. EnterpriseControlsTechnicalDiversity requires less Connectivity. **Each** EnterpriseControlsTechnicalDiversity requires less **some** Connectivity. **For each** Connectivity, **at most one** EnterpriseControlsTechnicalDiversity requires less **that** Connectivity. **It is possible that the same** EnterpriseControlsTechnicalDiversity requires less **more than one** Connectivity.
Issues	How are the different costs to be measured?

20	**Interoperability**
Statement	Software and hardware should conform to defined standards that promote interoperability for data, applications and technology.
Rationale	Standards help ensure consistency, thus improving the ability to manage systems and improve user satisfaction, and protect existing IT investments, thus maximizing return on investment and reducing costs. Standards for interoperability additionally help ensure support from multiple vendors for their products, and facilitate supply chain integration.
Assumptions	-
ORM Schema	
Verbalization	Enterprise uses Software. **Each** Enterprise uses **some** Software. **For each** Software, **at most one** Enterprise uses **that** Software. **It is possible that the same** Enterprise uses **more than one** Software. Enterprise uses Standard. **It is possible that more than one** Enterprise uses **the same** Standard **and that the same** Enterprise uses **more than one** Standard. **Each** Enterprise, Standard **combination occurs at most once in the population of** Enterprise uses Standard. Software conforms to InteroperableStandard. **For each** InteroperableStandard, **some** Software conforms to **that** InteroperableStandard. **Each** Software conforms to **at most one** InteroperableStandard. **It is possible that more than one** Software conforms to **the same** InteroperableStandard. Standard promotes Interoperability. **Each** Standard promotes **at most one** Interoperability. **It is possible that more than one** Standard promotes **the same** Interoperability. Hardware conforms to InteroperableStandard. **For each** InteroperableStandard, **some** Hardware conforms to **that** InteroperableStandard. **Each** Hardware conforms to **at most one** InteroperableStandard. **It is possible that more than one** Hardware conforms to **the same** InteroperableStandard. Enterprise uses Hardware. **Each** Enterprise uses **some** Hardware. **For each** Hardware, **at most one** Enterprise uses **that** Hardware.

	It is possible that the same Enterprise uses **more than one** Hardware.
Issues	How to ascertain if a standard promotes interoperability. How is consequently the conformance to standards to be checked.

3 Second experiment: Practical trial

To evaluate the applicability and value of formalizing architecture principles, the suggested formalization approach has been tried in a more practical setting as well. This trial took the shape of an interactive joint modelling session with an architect (from a large international IT consultancy firm) working in the field. The joint modelling session took about two and a half hours time in which two architecture principles were discussed, preceded by an introduction and discussion of the approach to be evaluated and the relationship to the approach commonly used by the practitioner.
In the joint modelling session two principles were discussed taken from one of the practitioner's clients (a large Anglo-Dutch oil company). This client operates in 145 countries all over the world and consists of 2500 different companies and joint ventures with governments and other companies, so the need for enterprise architecture is obvious.

Discussing the methodologies
In discussing the suggested formalization approach and the approach currently used by the practitioner some interesting questions and remarks arose.

- An issue raised by the practitioner was how to deal with situations when two companies merge, or engage in a joint venture, and there are conflicting principles. We explained that the current approach does not aim to provide mechanisms for the resolution of such conflicts, but that our method can be used to identify these potential conflicts by making the meaning of the principles in natural language explicit by using a formal object-role modelling method.
- A commonly encountered problem with companies the practitioner works with was that there often is *no (or no clear) prioritization of principles*. The unknown ranking of one principle over another becomes even more challenging when these principles overlap or conflict.
- For the client, a deeply rooted issue is the *compliancy* to principles, which is also recognizable in their architecture. The practitioner recollected an incident where the client was forming a cartel, which is illegal and thus non-conformity with their principle to comply with all applicable laws and regulations. The client publicly announced this non-compliance and that they have taken disciplinary measures against those involved and that this incident cost them 250 million euro. The business principles can be felt even on the shop floor in all activities.
- The method for architecture presentation as used by the practitioner provides for so-called sanity-checks. These checks were explained as "The sooner consistency and *sanity checks are made on assumptions*, the better." This supports our own opinion that documenting assumptions as a part of design rationale is essential in the creation of principles.

Conducting the experiment
In the joint modelling session two principles from client organization were discussed. These are defined as follows (translated from Dutch to English):

- *Principle 1: Compliancy*

Rationale: The client organization wants to comply with all law and regulations of the countries they operates in.
Background: Not complying with regulation results not only in economic loss but also loss of reputation and as such loss of market value.

- *Principle 2: Operate Globally, but work and support is regionally (or locally)*
  Rationale: The client organization is governed world-wide, but in daily practice there are a lot of local companies, often organized as a joint venture. This signifies that business operates regionally and locally.

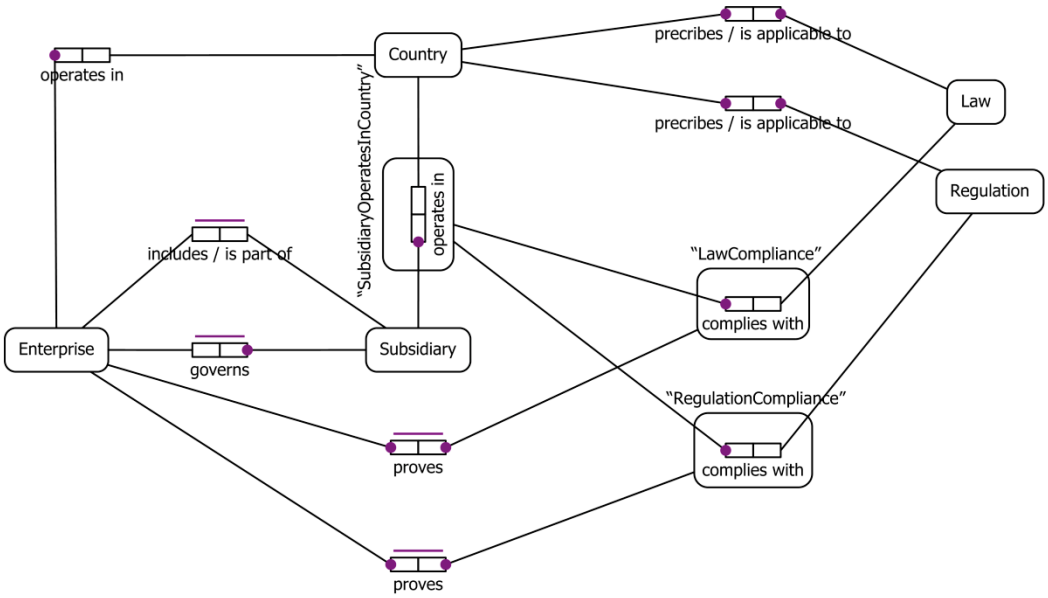
In a global review of these two principles it soon became clear that the second principle is not a true principle. It is merely a statement on the company's organizational structure. This statement may however contain a core that can be modelled into the form of a principle. The formalization approach used may point out anomalies or inconsistencies, but it is not intended to detect such non-principles. As such, principle 2 was not further discussed.

In discussing principle 1 it became clear that the Dutch version of the principle was different from the English one, leading to a misinterpretation of the intended meaning: The word "wil" in Dutch (translated above in 'wants to' was used whereas the English version uses "We comply". This original English statement does not imply a conditional element. Principles must be prescriptive and not optional; therefore a principle with 'wil/wants to' is not recommended.

From discussing principle 1 it was observed that to the client organization it is unmistakably clear that compliancy is the main element guiding its operation and management, together with Health and Safety.

Results of the experiment

The following table contains the result of formalizing Principle 1.

Principle 1	**Compliance**
Statement	Our organization complies with all applicable laws and regulations of the countries in which they operate.
Rationale	Not complying with regulation results not only in economic loss but also loss of reputation and as such loss of market value.
Assumptions	-
ORM Schema	
Verbalization	Enterprise includes Subsidiary. **It is possible that more than one** Enterprise includes **the same** Subsidiary **and that more than one** Subsidiary is part of **the same** Enterprise. **Each** Enterprise, Subsidiary **combination occurs at most once in the population of** Enterprise includes Subsidiary. Enterprise governs Subsidiary. **It is possible that more than one** Enterprise governs **the same** Subsidiary **and that the same** Enterprise governs **more than one** Subsidiary. **Each** Subsidiary, Enterprise **combination occurs at most once in the population of** Enterprise governs Subsidiary. **For each** Subsidiary, **some** Enterprise governs **that** Subsidiary. Enterprise operates in Country. **Each** Enterprise operates in **some** Country. Subsidiary operates in Country. **Each** Subsidiary operates in **some** Country. SubsidiaryOperatesInCountry **is an entity type**. SubsidiaryOperatesInCountry **objectifies** "Subsidiary operates in Country". Country prescribes Law. **Each** Country prescribes **some** Law.

	Each Law is applicable to **some** Country. Country prescribes Regulation. **Each** Country prescribes **some** Regulation. **Each** Regulation is applicable to **some** Country. SubsidiaryOperatesInCountry complies with Regulation. **Each** SubsidiaryOperatesInCountry complies with **some** Regulation. RegulationCompliance **is an entity type**. RegulationCompliance **objectifies** "SubsidiaryOperatesInCountry complies with Regulation". SubsidiaryOperatesInCountry complies with Law. **Each** SubsidiaryOperatesInCountry complies with **some** Law. LawCompliance **is an entity type**. LawCompliance **objectifies** "SubsidiaryOperatesInCountry complies with Law". Enterprise proves LawCompliance. **It is possible that more than one** Enterprise proves **the same** LawCompliance **and that the same** Enterprise proves **more than one** LawCompliance. **Each** LawCompliance, Enterprise **combination occurs at most once in the population of** Enterprise proves LawCompliance. **Each** Enterprise proves **some** LawCompliance. **For each** LawCompliance, **some** Enterprise proves **that** LawCompliance. Enterprise proves RegulationCompliance. **It is possible that more than one** Enterprise proves **the same** RegulationCompliance **and that the same** Enterprise proves **more than one** RegulationCompliance. **Each** RegulationCompliance, Enterprise **combination occurs at most once in the population of** Enterprise proves RegulationCompliance. **Each** Enterprise proves **some** RegulationCompliance. **For each** RegulationCompliance, **some** Enterprise proves **that** RegulationCompliance.
Issues	How to determine compliance?

Besides the direct results in ORM from the formalization of the principle, the practitioner concluded from the discussion that

- "It (i.e. the model) is a good representation of reality and shows the essence of the principle."
- "It (i.e. the model) is useful." The model forces the architect to carefully consider the relationships and cohesion between the underlying concepts. The construction of this model also brings new findings, insights and hidden conflicts to the fore.
- "You should do this (i.e. the process) with the boardroom of the client organization." The above identified insights gained from the modelling process would be highly beneficial.
- "(The model…) Gives insight into existing exceptions. Do not make the model too complex by modelling to allow for every exception. Because then there is the risk of switching abstraction level to rules and guidelines instead of principles.

- "The model shows also the fact that compliancy is something that is the responsibility of the Enterprise."

4 Individual reflections on the methodology

Reflection by Chris Nellen
In [BHPW06] formalizing using ORM is said to have two advantages. Besides achieving formalization of the architecture principle, the quality of the principle can be improved. The analysis of the principle causes reconsideration and thorough understanding of what it actually says. This reflection will discuss my views on the statement above on the basis of my experience with both formalizing TOGAF principles and the joint-modelling-session.

An important thing to notice is that with formalizing TOGAF principles one only has the source document available as a basis for interpreting the described principles. You have to make a lot of assumptions, which you write down, to be able to generate a formalized ORM model. The assumptions are direct proof in that you have found ambiguity in the principle. As such, it is indeed effective for improving the quality or effectiveness of the TOGAF principles. But since you are not able to communicate these assumptions back to a domain expert, let alone validate them, the TOGAF formalizations (i.e. the models) we have created are essentially worthless from a practical point of view.

Although they are worthless in the practical sense, in that it is unknown if they truly reflect the architects intent, the process of creating these formalizations has generated for me some important insights. One of these is determining the scope of what is to be formalized. TOGAF defines three major types of information to explain a principle, being statement, rationale and implications. One can formalize only the statement, this is the way I started formalizing the TOGAF principles. But after a discussion with the other project members I wondered if this adds any value to the model. If nouns from the statement simply change in entities and verbs in to fact types, nothing new is really done. Later on I realized, see the "Issue" column in the principles template, that producing the model also provided valuable insight into the relationships and concepts that needed more precise definitions.

During the rest of the TOGAF formalization process I somewhat struggled with how to determine what to add to the model and what not. There is always some cause or situation that gives rise to the existence of the principle. Next the principle itself which is mostly some kind of a single sentence rule, followed by the implications of this rule. With each formalization, we had to determine what parts of 'rationale' and 'implications' to take into consideration. If you do too much, you end up formalizing the entire domain area of the principle.

Another struggle was the formalization of the concepts of information involved at the level of architecture. Architecture tries to define guiding statements by using vague concepts of information. These concepts must be vague to some level to make sure they are viable for long term use and avoid giving short term solutions. ORM is indeed able to handle things that are not physical, but ideas like 'ease of use' and 'business continuity' are almost impossible to grasp within standard ORM. Determining a population for such concepts is not easy to do. Standard ORM does not really provide the mechanisms to define a method to "measure" the population of such concepts in practical situations.

I found ORM limited in the handling of such singleton type concepts. A second difficulty was the handling of negations: "something or other must not fail if something else is unavailable". A property that must always be present is another difficulty: "A Data definition is understandable". This can me modelled as a binary fact type, but what is the collection of 'understandable'? Should it be modelled as a unary fact type? How then to enforce that it is always the case for every data definition? Bear in mind

that I am not a professional ORM modeller and that some of these questions arose due to my lack of expertise. In the end the ORM model is as good as the modeller's knowledge of ORM.

A thing worthy to notice is the usability of the method in the boardroom. This is also linked with the expertise level of the modeller. A Corporate Information Officer might want to use formalizing principles in the boardroom to enhance quality and reduce ambiguity. The people in the boardroom are mostly bright people and can understand an ORM model. However, the modeller should then avoid the more complex modelling techniques that ORM provides. Question is then if the model is still able to represent the finer constraint of the principles meaning. This is something that also popped up in the case experiment.

From the case experiment an important conclusion is to mention that the model is only intended to clarify what is actually meant with a certain principle. The method does not provide any solution or any way to come up with a solution, in the conflicts that are made visible from the analysis. Neither does the method provide in a solution about ranking the principles.

Finally it is worth to mention that merging al separate models into one big picture could provide in additional insight in conflict between principles. For this however, you need a very strict definition library to make sure that all entities with the same name truly mean the same thing.

In the end I can conclude that ORM is indeed capable to achieve formalization of the architecture principle and at the same time possibly improve the quality of the principle itself. However I have doubts regarding the reliability of the formalization that is achieved. If at this moment two different modellers would formalize the same set of principles, the result would be quite different. The formalization results of both will be unambiguous, but the method of achieving this state is not.

Reflection by Guido Chorus

Formalizing architecture principles with ORM seems to be very useful in creating an understanding and consensus in the meaning of architecture principles. All TOGAF-principles I encountered during our assignment had a deeper meaning then just the sentences itself. Formalizing the principles encouraged me to think about the principle and thereby opening the world behind the principles. The world of a principle contains two kinds of meanings, one that is meant by the architect and the one that is perceived by the reader. Formalizing the principles with ORM brings these two kinds of meanings closer together or even the meanings become the same.

The ambiguity of the principles, which they all implicitly have, is reducible by formalizing the principles. Note that the ambiguity only is reduced for the people who participated in the formalization of the principle. Reading the ORM-model only, without having participated in the formalization process, is still very easy to be miss-interpreted and ambiguous because of concepts used for objects (explained later), the two meanings mentioned earlier are still not completely the same when only reading the ORM-model. This is strange because ORM is a formal method which should eliminate this ambiguity. In my opinion, formalizing the principle is far more important than the product (ORM-model) created with the process.

A problem with the proposed is that much knowledge of ORM is required. I think that architects and boardroom members (the people who should use the method!) do not have the time to learn ORM, thereby reducing the possibilities for the usage of the method. ORM gave us another problem; because many principles contain concepts that are not identifiable as data-elements, it was "strange" to use ORM objects for them, because ORM is not meant for singleton objects. The constraints of ORM are

also not completely applicable to those objects. An example of such an object we used was: Interoperability.

Our way of working is not very strict. We created a template which we filled in during the process. The way of working is a sort of "joint-session" in which we all first read the principle and then 1-by-1 give our notion of the meaning of the principle. Then we tried to formalize the principle by first pointing out the important concepts and meaning we saw. Afterwards the ORM-roles were added. The most important step in the process is adding the constraints. In my opinion, the constraints enabled us to give a more exact (less ambiguous) representation of the meant meaning of the principle. A notable tool when formalizing principles is NORMA. During the creation of the ORM-models, we directly put the model in NORMA. We were able to directly model-check and verbalize constraints which gave us more insight in the ORM-model and background. In addition, it gave us more points to discuss and thereby made the ORM-model more justified and enhanced the consensus creation.

The way of working proved to be very hard when experimenting with the architect. We (Yves, Chris and me) have created some kind of consensus in how to formalize principles. This consensus has probably come into being in the many hours we sat together discussing the principles. The start of the joint-session with the practitioner was therefore very slow, because he did not have that consensus. In addition, the practitioner did not have any knowledge of ORM which also reduced the speed.

The reliability of our way of working is very low because much subjectivity is always present because much cognitive processes are required in the process. In addition the method is not strict (no fixed steps) and therefore the stability and reproducibility are very low. We saw this problem a few times when we tried to create an ORM-model of a principle from scratch which we had already formalized some time earlier. A few times we created a completely different ORM-model. In my opinion, the only stability in the way of working is given by the architect (domain-expert) who should steer the discussion in the joint-session to produce the right consensus (exact match between perceived and meant meaning) and ORM-model among the participants because he/she is the only one with the "right" notion of the meaning behind the principle.

Reflection by Yves Janse

In this reflection, I draw some conclusions on the results of the research project I performed with the other two students, Guido Chorus and Chris Nellen. In addition, I will illustrate some of the strengths and weaknesses of the proposed method of using Object-Role Modelling to formalize architecture principles. I will conclude this reflection with a few points of interest related to this project.

The goal of this project was to discover whether it is a good idea to use Object-Role Modelling (ORM) to formalize architecture principles. My personal answer to this is *yes, with a few comments*. I believe that formalizing architecture principles with ORM is a good and valuable method but it should not be seen as a miracle solution. Principles formalized with ORM are by no means something that should be part of an end-result; i.e. the ORM models are not the final deliverable and the only way the principles are formulated. One of the main important goals of architecture principles is that they should be understandable for everyone, and ORM models are sometimes hard to grasp for those unfamiliar with the method. However, the ORM models prove their worth during architecting as a working document that, when all principles are completed, may be added to an architecture document as an appendix.

Observed strengths of the proposed method are the following:

- The method makes the assumptions made during interpretation of the principles statement explicit. This is valuable because the inherent problem of the natural language description is that

it likely to be ambiguous. The assumptions that are made are documented so they are traceable and verifiable.

- The result of the method as well as the act of performing the method exposes the perceived meaning of the architecture principles. This should match exactly with the intended meaning in the ideal situation.
- This results in the elimination of ambiguity as much as possible, which leads to a set of better architecture principles. It is often argued that principles should be SMART and I believe that this method can help reach that goal.
- The act of performing this method leads to a better understanding of the principle by all people who are involved and participate. This is very important because it increases commitment and support when members of the boardroom are involved in the join-modelling sessions. The method clearly serves as feedback to the architect.
- By making the assumptions and thereby the perceived meaning explicit, one of the results of performing this method is that it uncovers potential conflicting situations. This is a valuable result because it can be the incentive for an organization to adjust or review the prioritization of architecture principles. The discussion that ensues will result in a revised architecture that better suits the organization and its goals, reflecting the order of importance of principles.

Observed weaknesses of the proposed method are:

- ORM is not the easiest and most intuitive method for those unfamiliar with it, even though it is a method that is used widely in many companies. The strength of using ORM for this purpose is in the constraints that make the meaning of the principle explicit and rigid, but the constraints in the graphical language of ORM are the least intuitive element for laymen. We encountered this during the practical experiment; the architect we spoke was not familiar with ORM which posed a slight communication problem. However, this problem was solved quickly by a quick introduction to the basics of ORM notation. I believe that an average boardroom member may need a longer introduction to ORM to be able to participate efficiently.
- It is very hard to capture concepts that have a "Singleton"-like character in ORM diagrams. An example of this is the concept "Enterprise" of which there is only one in the architecture principles we explored. This leads to a strange situation when determining the uniqueness and mandatory role constraints, because the population of the entity type is always the same element. The resulting verbalization is therefore less fluent and comprehensive because of this.
- One of the well-known characteristics of architecture principles is that they can be described in rather verbose language, leading to a broad range of possible interpretations as a result of the inherent ambiguity. Extracting the intended meaning from the statement and rationale is a hard process that is based on assumptions. Even though these assumptions are documented, verified and traceable, they remain assumptions. Putting the assumptions in a formal model does not necessarily bring the assumptions closer to the intended meaning, but they do clarify the perceived meaning in a way that it is possible to discuss.

Considering these strengths and weaknesses, I argue that the method is valuable and will certainly contribute to the quality of the architecture principles when attention is paid to the possible pitfalls and limitations of the method. I do not think that ORM is the only method that can be used to achieve these goals; there will be other conceptual modelling methods that may serve this purpose.

This method should be able to cope with complex constraints, conceptual entities and be relatively intuitive so it can be used as a communication vehicle with the boardroom and architects.

Some additional observations are:

- The TOGAF principles formalized in this report are all about Information Management according to the authors, whereas there is a likeness with the principle of compliancy from the architecture of the client organization we formalized in the practical experiment. The rationale behind the client's principle is much richer and complicated because it is tailor-made for this particular organization. It was very interesting to formalize this principle together with one of the architects because we dynamically created the model together with continuous interaction, refining the model and gradually increasing the match with the intended meaning.
- This method is most useful when it is performed interactively in a joint session-session because then all participants can share their views directly. It is essential that some member of the boardroom of the company is participating in this process.
- When all architecture principles are formalized, it will be an interesting test to try to combine the ORM models into one large model. In the ideal situation, this should pose no problem at all and the models should fit seamlessly. This is however probably never the case because of the hard to determine scope of the individual models. A concept can have a different meaning but the same name in two separate models, for example: "Question" in the context of a question asked from a particular perspective (enterprise or organization) as opposed to a question that is raised at a helpdesk. These are often namespace issues but sometimes design choices result into conflicts that will be unveiled when the model is put together. As in the real organization the architecture principles exist together in coherence, I believe it is a good practice to test this. Unfortunately, due to time-restrictions in this project we have not been able to do this with the TOGAF principles.

5 Conclusions

The prime conclusion that can be drawn from the two modest experiments is that further experimentation is indeed worthwhile since our hypothesis of the potential usefulness of the proposed ORM and ORC approach to formalizing architecture principles has not been falsified. At the same time, some additional careful conclusions can be drawn:

- The formalization approach makes the assumptions made during interpretation of the principles statement explicit. By recording these assumptions, principles and their formalization are likely to become more transferable.
- The models resulting from the formalization process, as well as the actual act of doing the formalization, does expose the perceived meaning (and more particular harboured by different stakeholders) of the architecture principles
- The act of performing the formalization also leads to a better understanding of the principle by all people who are involved in the process.
- Care should be taken about the exposure of stakeholders and domain experts to the ORM notation as it is not the easiest and most intuitive notation to those who are unfamiliar with it. This may require the use of a lightweight version of ORM focusing on the bare essentials: object types, fact types and instances.
- A number of principles required the capture of instances as explicit modelling constructs. Concepts such as "Enterprise" and "Interoperable" have been modelled as object types, but should rather have been treated as instances. What seems to be needed is to allow for instances to be modelled explicitly and furthermore to allow them to participate in fact types.
- By starting with some form of formalization / grounding of architecture principles early on in the modelling process, one may be able to avoid modellers from using unnecessary verbose language in their formulation.

The approach to take in future real-life experiments with architects (in the role of model mediator) and stakeholders (in the role of domain experts) would be to follow the following:

1 Identify the concern / business issue that needs to be addressed by the principles.
2 Prioritize these concerns.
3 Formulate candidate architecture principles, while using the list of concerns as a checklist since each concern should be addressed by at least one principle. At this stage, for each candidate principle should have a name (typically a catchy phrase), a first brief description of its statement and a rationalization.
4 Using the prioritization of the concerns, select which principles to actually enforce and provide the priority of these principles.
5 For each of the selected principles (depth first or breath first):
    a. Provide a domain model of the underlying business concepts (using a lightweight ORM notation). Use the concern(s) which are addressed by the principle as a scoping mechanism.
    b. Clarify any mismatches that may occur with the overall domain model and/or domain models underlying other principles.
    c. Re-formulate the principle's primary statement in terms of ORC rules expressed in terms of the domain model. In doing so, the architect (model mediator) has to use a full-fledged ORM model. In communicating to the domain expert, the light-weight ORM notation may be used.
    d. Express the principle's rationale in terms of the way it addresses the concerns leading to the principle.
    e. In line with the rationale, identify measuring techniques / methods needed to assess the conformance of system designs to the architecture principles.

6 References

[BHPW06] P. van Bommel, S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. Giving meaning to enterprise architectures - architecture principles with ORM and ORC. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM Workshops - OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2006*, Lecture Notes in Computer Science, Montpellier, France, EU, October/November 2006. Springer, Berlin, Germany, EU.

[IEE00] Recommended Practice for Architectural Description of Software Intensive Systems. Technical Report IEEE P1471-2000, The Architecture Working Group of the Software Engineering Committee, Standards Department, IEEE, Piscataway, New Jersey, USA, September 2000.

[Lo05] M.M. Lankhorst and others. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Germany, EU, 2005.

[OP07] M. Op 't Land and H.A. (Erik) Proper. Impact of Principles on Enterprise Engineering. Technical Report ICIS-R07001, Radboud University Nijmegen, January 2007.

[TOG04] The Open Group. *TOGAF - The Open Group Architectural Framework*, 2004.

[USA96] USA Government. Clinger-Cohan; IT Management Reform Act, 1996.

[xAF06] xAF working group. Extensible Architecture Framework version 1.1 (formal edition). Technical report, 2006.