International Journal of Computer Techniques -– Volume 2 Issue 1, 2015

RESEARCH ARTICLE OPEN ACCESS

IT Governance Ontology Building Process : Example of developing Audit Ontology

Chergui Meriyem[1], Sayouti Adil[2], Medromi Hicham[3]
*(ENSEM, Hassan II UniversityLISER-EASCasablanca, Morocco)

--------------------------------*****************--------------------------

Abstract

This article aims at building ontology based on COBIT Framework named: "AuditOntology", to be used in an IT Governance solution, through this implementation, IT Governance Ontology building process will be proposed in order to be used for other frameworks Ontologies. In fact, the ontological meta-modelling is nowadays, a necessity for every discipline to share concepts and terms and to annotate information. It clarifies knowledge structure for better understanding of domain vocabulary and entities relationship. Ontologies are often made for reasoning purposes; there engineering benefits are about stakeholders' communication, systems interoperability and re-usability. Till now, there is no IT Governance ontology with a scientific foundation to be deployed on AI software solutions or conceptual modelling so the main goal of this work is to propose ontology based on COBIT Framework in Web Ontology Language using particular life cycle steps.

Keywords:-**COBIT, Ontology, OWL, semantic web, meta- modelling, IT-Governance, Audit, Information System.**

----------------------------------*****************--------------------------

I. INTRODUCTION

.Information technologies (IT) have more and more impact on companies' revenue, making differences on their evolution function. Information Systems (IS) become a serious investment in front of world market agility and exponential changing; it's also an asset on witch companies rely to achieve business goals **[1]**.But this investment is not 100% a successful one, IT introduction and implementation makes enterprises wasting big budgets without an efficient result in case their deployment project was not correctly measured , randomness has no place on IS overhaul or business applicative development.

IT Governance (ITG) is the vital and unique solution to ensure positive returns [2]**.** Indeed, Information System Experts put at the disposal of companies best practices and frameworks able to control their IT decisions and projects since the launch until the commissioning, dealing with risks and services management. Although, ITG implementation is still a complex task which is not very practical since it depends on companies' context and business particularity, so the challenge is to understand these frameworks structure to be able to control how adaptable they are to the information system and where can we use each of them.

As a result, we need a logical structure modelling of ITG frameworks for an effective integration.

In this sense, Information Technology Governance Institute (ITGI) [3] made a multi-criteria classification with framework functions mapping but it still lacks of theoretical and scientific foundation, despite its efficiency in practical special cases.

As a solution, ontology building in ITG context seems to be an essential step to overcome so as to give a high level of abstraction modelling capable to explain logical structure and relationships between different ITG frameworks components. This in harmony with other disciplines standardized Ontologies development examples: SNEMOD **[4]** in the medical domain, UNSPSC for products and services terminology [5] etc.

In this paper, we try to build method and task ontology based on COBIT framework, using its existing meta-model, documentation and terminology indexation.

The article is presented as following, section 2 presents ontology literature review, section 3 detailed COBIT constitution and its IT Governance contribution section 4 shows different steps of Auditontology building before giving a conclusion and perspectives

I. ONTOLOGIES IMPLEMENTATION

A. Definitions

The Artificial Intelligence (AI) literature gave many definitions of ontology sometimes conflicting one another. The most pertinent and near to our work context is Gruber one defining ontology as "a specification explicit conceptualization”.
In 1997, Borst has defined an ontology as "a formal specification of a shared conceptualization": these two definitions were combined by Studer et al. (1998): "An ontology is a formal and implicit specification of a shared conceptualization ". The construction of a ontology intervenes only after the conceptualization of work has been completed [6].
An ontology with instances of individual classes made in AI a Knowledge Base, the Class is the focus of any ontology, a class can have subclasses to detail the concept of the Superclass the relation between subclasses and Superclass is “subsumption” relation or “is- a” relation. There are other kinds of relations defined in conception context.
Ontology class has also properties to characterized it, these properties are valued and are either data properties/ slots or object properties,

B. Types of Ontologies

There are many types of Ontologies in semantic web literature the most common are namely [7]:

- Generic ontology: describe very general concepts such as space, time, matter, objects, events, actions, etc., which are independent of a problem or a particular area of application.
- Domain ontology: vocabulary linked to a generic domain by specializing the concepts presented in Generic ontology: electronic, automobile…
- Task ontology: vocabulary linked to special task or activity.
- Method ontology: the role played by each concept in the argument is made explicit.
- Application ontology or task and domain ontology: write concepts depending on both of a domain and a particular task, which are often two specializations of the related Ontologies. These concepts often are the roles of domain entities while performing a certain activity, such as replaceable unit or component available.

We can eventually represent the relationship between different types of Ontologies in figure 1 bellow:

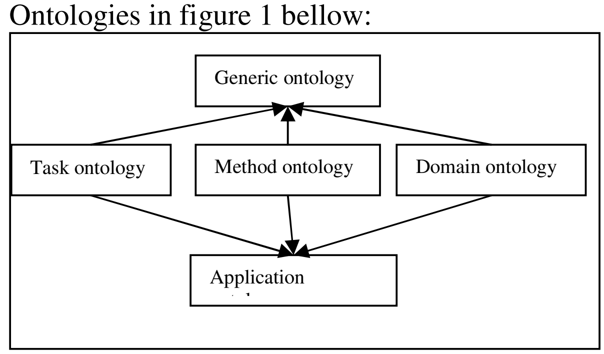

Fig1. Specialization relations between ontologies types

C. Representation Languages and construction tools

There are many languages and formalism used to represent Ontologies, the most common are:

- Ontolingua: for portable Ontologies, defines classes, relations and functions on KIF (knowledge Interchange Format)) formalism [8], able to translate generic Ontologies to Loom, Epikit and KIF.
- Loom: later Power loom is a knowledge representation platform able to make reasoning tools. Based on chaining earlier, the semantic unification and object-oriented technologies to provide a deductive Support.
- OIL: ontology Inference Layer is a language for Ontologies, representation and influence of combining modeling primitives frame languages with formal semantics and descriptive logical reasoning methods. Widely used for the web, based on RDF/ RDFS and XML formalism [9].
- SHOE: Simple HTML Ontology Extensions is an extension of HTML that allows Web page authors to generate annotation of their documents that can be understood by a machine. This language can be used by agents [10].
- OWL: knowledge representation language especially web Ontologies, based on RDF data model, its second version is W3C recommendation, it has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full [11]

In this work, we opted for OWL, since it can be interfaced with all other language and it has a variety of tools to implement ontology with it.

As for tools, there are many, we presented in a previous article the most known [12]. But Protégé remains till now the most popular tool for ontological engineering; we use it as well to implement AuditOntology.

D. Ontologies Conception methodologies

In semantic web literature, there is no unified life cycle, methodologies and techniques to construct Ontologies. But authors review three types of approaches to concept a formal ontology [13]: Bottom up; Top Down; Middle approach [14].

Later, Gomez-Perez [15] proposes a method called METHONTOLOGY, consisting on: 1.listing all the concepts of a domain, 2.conceptualize a set of intermediate relationships between these concepts, 3.implement the model, and 4.evaluate.

There are other propositions such as TOVE [16] but this approach is specialized on evaluating Ontologies, after establishing skills from existing scenarios. Other authors had simply made acceptance criteria [17] or construction difficulties.

In our case, we will use a method close to MEHTONYOLOGY, with IT governance particularities. We will detail it in section 4.

II. COBIT AS IT GOVERNANCE FRAMEWORK

COBIT (Control Objectives for Information and related Technology Business), developed in 1994 (published in 1996) by ISACA (The Information System Audit and Control Association) is an IT governance framework specialized on control objectives of information technology for both Information System Management and business stakeholders.

A. COBIT Contribution

COBIT has many goals expressing top management preoccupations such as:

-Strategic alignment between IT and Business,

-Efficiency by bringing benefits to processes operation,

-Responsible and optimized use of resources with IT,

-IT Risks control in relationship with business.

COBIT originality is the creation of communication links between IS Management and business actors, minimizing strategic distances among stakeholders. It's the reason why COBIT proposes 34 Generic Processes, divided into 4 generic domains, these processes and domains are reviewable according to the

organization specificities. Moreover, COBIT can easily be coupled with other best practices such as CMMI, ITIL, and ISO27001…etc.
COBIT also allows enterprises to compare their processes to others companies, or to evaluate themselves through Generic maturity models.

B. COBIT Components

COBIT defines 4 domains [**18**]:

- **Plan and Organize** (PO): it's the strategic dimension of IT governance.
- **Acquire and Implement** (AI): it's about resources identification, acquisition, development and deployment
- **Deliver and Support** (DS): it's about client services (management of security, data and continuity).
- **Monitor and Evaluate** (ME): it's about control dimension, performance management and audit.

For each 34 IT process COBIT detailed:
Control objectives: it has general view and detailed view, it aims at describing general issues of the IT process such as, IT goals, activities, process goals and metrics, its eventually about results we get while applying the IT process and it helps auditors to defined specific investigation grids.
Management guide: it's about The IT process inputs/outputs, responsibility matrix of key activities and Goals and Metrics matchmaking.
Maturity Model: it's a model inspired from CMMI able to measure in a general way the IT process application level to guide its implementation and its improvement.

C. How to use it?

As generic IT governance framework, COBIT implementation is based on maximum of IT processes deployment remaining within the limits of an appropriate scale project.
Respecting the Balanced scorecard representation, COBIT can be deployed in
-**An operational way** by deploying: the operation management (DS13 process), the physical environment (DS12 process), changes (AI6 Process)
-**A strategic way** by deploying: the three-year plan (PO1), investments (PO5), risk management (PO9), project portfolio (PO10) monitoring of governance (SE4).
-**A Consumer relationship way** by deploying: contractual services level (DS1), (DS10), (DS8).
-**An anticipation way** by deploying: human resource needs (PO7), organization (PO4 and PO8) suppliers management (DS2) technological developments and business needs (PO2 and AI1), architectures changes (PO3).

III. AUDITONTOLOGY BUILDING

AuditOntology is a result of many information sources consolidation. In fact, in the absence of IT Governance semi formal Ontology, we build AuditOntology from:

- 1st COBIT indexing: We choose MAUI indexer, an open source project with GNU license, efficient for text indexing, able to resolve keywords and main thesaurus in many fields **[19]**.
- 2nd COBIT existing meta-models, essentially based on Entity/relationship Methodology, namely: the official COBIT Architecture model [1].
- 3rd COBIT IT processes descriptions for detailed relationships and cardinalities and practical validation.

We opted for METHONTOLOGY as conception methodology and we added successively the previous steps at knowledge acquisition and conception stages. In fact, these 3 actions can be used for every IT Governance Risk Conformity Framework, while building its ontology and this can be considered in ontology building literature as a semi formal modeling or "Ontologization" which will be followed by a translation to a formal language or "Operationalization" **[20]**
AuditOntology is eventually open to enrichment since it's made in OWL and based on COBIT framework which is interfaced with many frameworks and good practices such as ITIL, COSO, Val IT, SOX, PMP…etc.

A. Specification

AuditOntology is formal task ontology for Audit and IT Governance, its main goal is to guide Information Systems users to evaluate Business objectives by themselves through COBIT framework to get: IT processes measures, metrics, responsibilities, activities, resources, IT Goals.

AuditOntology describes vocabulary about IT Governance and Information Systems Audit, in addition to functions and methods done by COBIT components, and this in independent way from enterprise IS and context.

It answers questions such as: witch maturity level did IS achieved? What should we do to improve it? Who's responsible for a given Business Objective or IT Process? Which IT processes can be convenient for a real Business Objective?

In brief, AuditOntology supports all COBIT Components' operations and methods to measure the strategic alignment of an IS to Business matters, to require values and well manage human and material resources.

B. Knowledge acquisition

As said before, in this step, we address first a listing of main vocabulary required from MAUI Indexer applied to COBIT 4.1 framework, the result was as following:

TAB1: MAUI INDEXER APPLIED TO COBIT 4.1

Concepts		Actions
IT ressource	activities	IT ressource
level of maturity	applications	level of maturity
management and control	business	management and control
management guidelines	business goals	management guidelines
management process	business requirements	management process
maturity	COBIT	maturity
procedures	control	procedures
program	control objectives	program
RACI chart	development	RACI chart
requirements	development process	requirements
resources	enterprise	resources
responsibilities	framework	responsibilities
monitoring	goals	monitoring
risks	governance	risks
security	information	security
performance	information criteria	performance
service levels	internal control	service levels
solutions	investment	solutions
standards	tactical plans	standards
strategic		strategic

Second, we studied existing Meta-models and reread IT processes descriptions to give away all questions this ontology can answer in ITG context.

Let's share some of them:

1. Which COBIT Business Objective corresponds to this request?
2. Which IT Goals are linked to this Business Objective?
3. Which IT processes should we deploy?
4. Which controls should we apply to every IT Process?
5. Which metrics are suitable for every control?
6. Which key activities should we execute for an IT process? And who is responsible for these activities?
7. What maturity level did this IT process achieve and for what maturity model?

C. Conceptualization

In this step a Class diagram of the framework should be established, in COBIT case we did the diagram Figure 2:

From previous steps, we were able to define the basic concepts:

IT Domain: COBIT contains 4 domains containing coherent processes.

IT Process: it's the central entity, COBIT has 34 processes that belong to the 4 domains.

Activity: every IT process is divided into many activities, having a management view.

IT Control: Its controls IT process can have, with an operational view for auditors.

Goal: defined by an IT process, measured by a ***Metric*** this represents a key indicator.

Maturity Model: specific declination able to measure every IT process and drive its improvement.

Maturity Level: it's a numeric value from 0 to 5 to measure the process in ***Maturity Model***

Information Criteria: it's information segmentation trough 7 criteria: Effectiveness, Efficiency, Confidentiality, Integrity, Availability, Compliance and Reliability

IT Resource: it concerns, application, infrastructure, information and persons.

IT Governance Focus Area: it's the five control focus area of COSO: Strategic alignment, Value delivery, Risk Management, Resources Management and Performance Measurement.

Result: IT process in output of another IT process.

Role: It's the responsible for a key activity implementation**.**

As for properties there are two types, data properties and object properties.

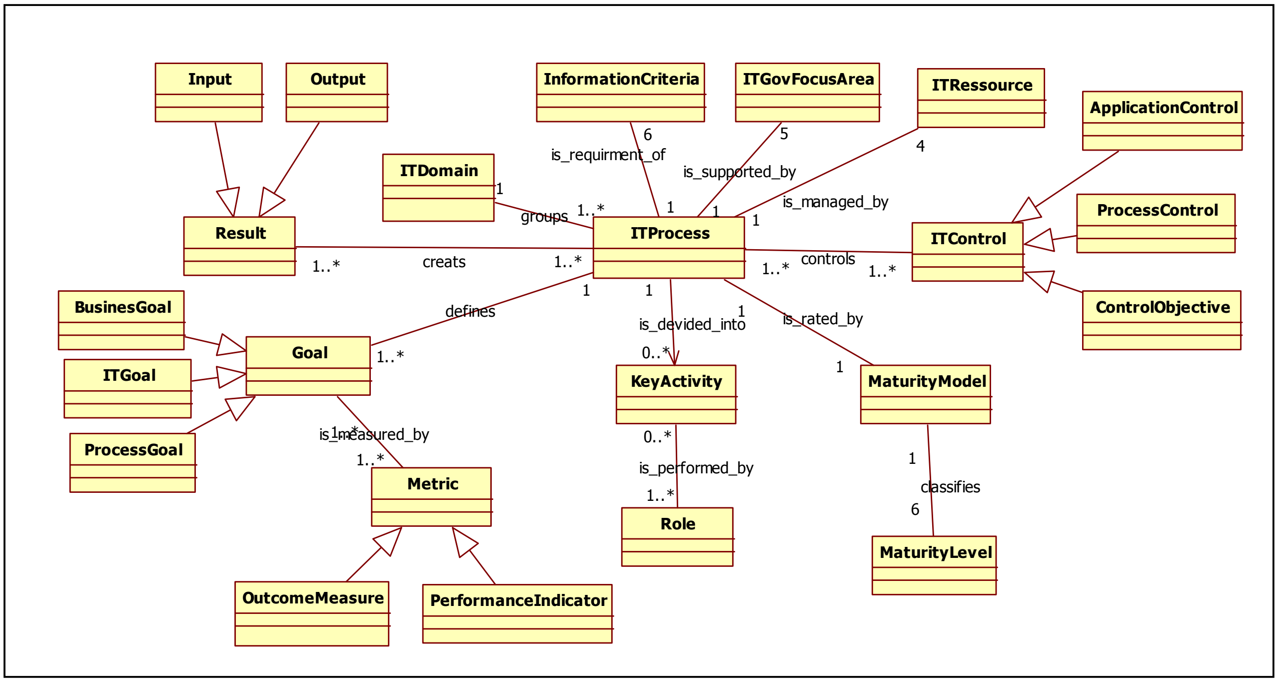

Fig2. The Proposed Class Diagram of COBIT Concepts

The main important object properties are:

Is-a: generalization relation

Is-audited with: relation between IT Control and Control Test.

Is-classified-by: relation between Maturity Model and Maturity Level

Is-controlled-by: relation between IT Process and IT Control

Is-measured-by: relation between Goal and Metric

Is-rated-by: relation between IT Process and Maturity Model

Is-supported-by: relation between IT Process and ITG Focus Area

And there are other object properties relating basic concepts. As for data properties it's Concepts description

Properties: example ITProcess (String processCode, String processDescription).

D. Implementation

To implement AuditOntology, we choose OWL-DL as language and Protégé 4.3 as ontology editor, we pursuit the following steps:

Step1: Active ontology definition and description,

Step2: Classes and subclasses creation,

Step3: Object Properties creation,

Step4: Data Properties creation.

Step5: individuals' creation

We get the following result:

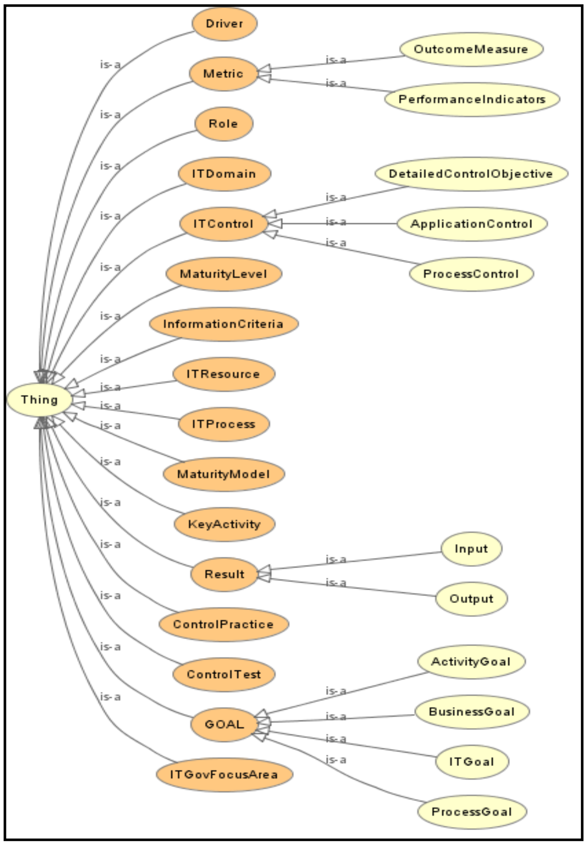

Fig3: OWLViz view of AuditOntology

E. *Ontology Evaluation*

METHONTOLOGY require an evaluation of the proposed Ontology, it's an integrity evaluation done through an inference engine to show how consistent is the ontology. There are many inference engines such as, Jena, Hermit, Pellet and Fact++.

In our case we choose Fact ++ integrated as Plug-in of Protégé to test AuditOntology and through its propositions we get the following inferred classes:

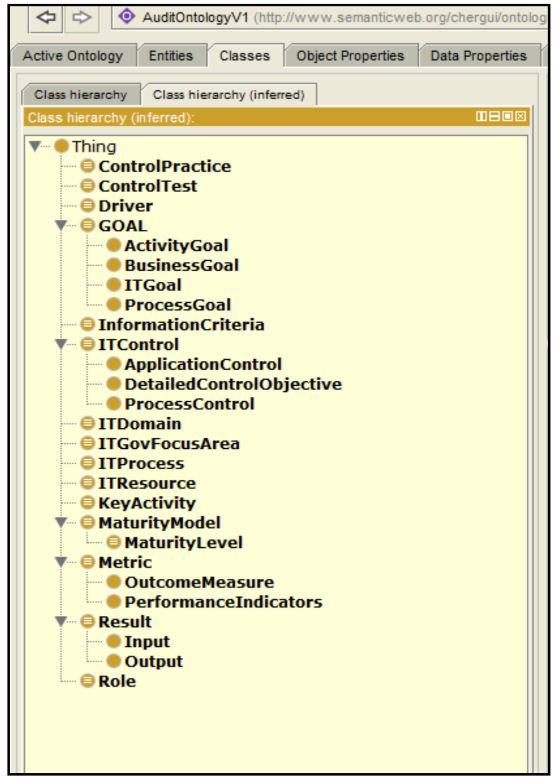

Fig4: AuditOntology inferred classes

II. Conclusion and Perspectives

AuditOntology is task ontology of IT Governance describing Audit Process through COBIT.

It is based on the indexation of the IT Governance Framework, COBIT meta-models federation and IT processes analysis.

AuditOntology is developed with OWL-DL language in Protégé ontological engineering tool, it was evaluated by Fact ++ Inference engine.

As perspective of this ontology we will integrate it in the Knowledge Base of the IT Governance Platform "AUDIT-EAS" **[12],** to make sure of its efficiency and to validate its performance.

REFERENCES

[1] Gerrard, M. (2009). IT Governance, a Flawed Concept: It's Time for Business Change Governance. Gartner Research.

[2] Jacobson, D. D. (2009, January). Revisiting IT governance in the light of institutional theory. In System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on (pp. 1-9). IEEE.

[3] Selig, G. J., & PMP, C. (2008). Implementing IT Governance. Cell, 203, 521-9664.

[4] Stearns, M. Q., Price, C., Spackman, K. A., & Wang, A. Y. (2001). SNOMED clinical terms: overview of the development process and project status. In Proceedings of the AMIA Symposium (p. 662). American Medical Informatics Association.

[5] www.unspsc.org.

[6] Zemmouri, E., Behja, H., Marzak, A., & Trousse, B. (2011). Intégration des connaissances de domaine dans un processus d'ECD multi-vues. Proceedings of 4èmes Journées Francophones sur les Ontologies JFO2011, 73-85.

[7] Charlet, J., Bachimont, B., & Troncy, R. (2004). Ontologies pour le Web sémantique. *Revue Information, Interaction, Intelligence I3*.

[8] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, *5*(2), 199-220.

[9] M. Klein, D. Fensel, F. van Harmelen and I. Horrocks. The Relation between Ontologies and Schema-Languages: Translating OIL-Specifications to XML-Schema.In Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence ECAI-00, Berlin, Germany, August 20-25th 2000.

[10] Luke, S. and J. Heflin. 1997. SHOE 1.0, Proposed Specification. At http://www.cs.umd.edu/projects/plus/SHOE/spec.html.

[11] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(10), 2004.

[12] Chergui, M., Medromi, H., & Sayouti, A. (2014). Inter-organizational Workflow for Intelligent Audit of Information Technologies in terms of Entreprise Business Processes. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *5*(5).

[13] Van Der Vet, P. E., & Mars, N. J. (1998). Bottom-up construction of ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, *10*(4), 513-526.

[14] Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, *11*(02), 93-136.

[15] Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

[16] Gruninger, M., Schlenoff, C., Knutilla, A., & Ray, S. (1997). Using process requirements as the basis for the creation and evaluation of process ontologies for enterprise modeling. *ACM SIGGROUP Bulletin Special Issue on Enterprise Modelling*, *18*(3), 52-55.

[17] Lozano-Tello, A., & Gómez-Pérez, A. (2004). Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, *2*(15), 1-18.

[18] Brand, K., & Boonen, H. (2007). IT Governance based on CobiT® 4.1-A Management Guide. Van Haren.

[19] O. Medelyan. 2009. Human-competitive automatic topic indexing. PhD thesis, department of computer sciences, University of Waikato New Zeland.

[20] Fürst, F., & Trichet, F. (2005). Integrating Domain Ontologies into Knowledge-Based Systems. In *FLAIRS Conference* (pp. 826-827). *and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.