A Governance Requirements Extraction Model for Legal Compliance Validation

Hassan, Waël
School of Engineering and Information Technology
University of Ottawa
Ottawa, Ontario
wael@acm.org

Logrippo, Luigi
Département d'informatique et ingénierie
Université du Québec en Outaouais
Gatineau, Québec
luigi@uqo.ca

***Abstract*—We present a model-based approach to extract governance requirements from the law and enterprise regulations, to formal specifications. This is the first step of an end-to-end implemented methodology for validating legal compliance of enterprises to law through logic models. Our UML-based Governance Extraction Model (GEM) is able to extract many legal and enterprise requirements, particularly business process and access-control requirements. Examples from Canadian and USA financial and privacy laws are provided. As a result of our extraction process, logic analyzers were shown to be able to detect compliance faults.**

Keywords: governance; requirement extraction; compliance; privacy law; financial law; UML metamodel; PIPEDA; SOX.

I. INTRODUCTION AND MOTIVATION

We have implemented a process for validating legal compliance of enterprises to law though logic models and logic analyzers. The first step of this process is to refine and extract legal and enterprise requirements. The second step is to represent the requirements in logic based language. The third step is to validate enterprise compliance to requirements using logic analysers. This paper concentrates on the first step.

Governance officers (GOs), who are accountable for ensuring compliance to laws, seek methods and tools for the analysis and validation of regulatory requirements, and this need is motivating research in modeling aspects of regulations at the enterprise-level in compliance management systems [28]. Modeling regulations presents the challenge of requirement extraction (RE), which is concerned with extracting the relevant compliance data from the law and the enterprise [15]. For instance, a typical RE task might be to find management rules required by the law. Several example solutions are listed in [8] [18].

Requirement extraction is a challenging task since legal documents are written in natural language in its full complexity [3], they can contain vague terms, complex dependencies between provisions, and legal lacunae [9]. In particular, *governance laws*, privacy and financial, are complex and can include a large number of internal and external definitional references, dictionary and ontology requirements, obligations, permissions, process definitions, conditional statements, and others. Enterprise governance requirements are also complex, they contain requirements similar to those present in the law with possibly a different vocabulary. Obviously the extraction process cannot be fully automated, but it can be assisted by methods and tools.

A key component of any RE system is its set of extraction classes that are used to match each requirement. Finding useful extraction classes is a difficult, time-consuming task, and several research efforts have focused on learning the extraction rules from training examples [17].

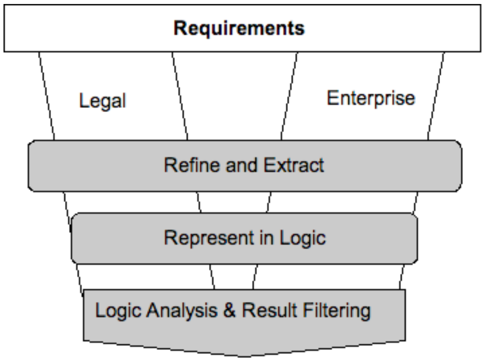

Figure 1. End-to-End Process

Figure 1 shows our end-to-end process of compliance validation. Starting from legal and enterprise requirements a GO extracts requirements using the method described in this paper. The extracted requirements are represented in our logic based language which is passed on to the Alloy logic analyzer [16], which in turn generates diagnostics, such as violation scenarios. This work has been completed and shown to capture compliance results [11]. Note that in our work, we do not regard the laws as programs but rather as governance requirements for enterprise systems [29].

In this paper we describe the very first part of the process: we explain how a UML class model, called GEM for Government Extraction Model, can be used to extract governance requirements from plain legal text.

As shown in Figure 2, the requirements are matched to model components. Eventually, the manual process can be supported using a user interface tool for assistance.

We take examples from Sarbanes-Oxley, a US financial law [27], and its Canadian equivalent, Canadian Instruments 52-111 [13], in addition to examples from the personal information protection and electronic documents Act (Privacy Law) from Canada[25].

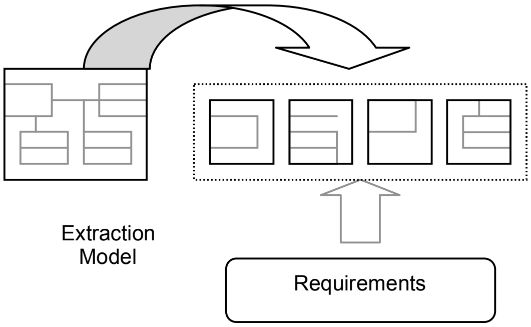

Figure 2. Extraction Process

These examples may be modified or abbreviated for the purpose of facilitating the presentation.

II. CHALLENGES AND RELATED WORK

Laws dictate enterprise requirements. They usually define the 'what' aspect of a requirement, whereas enterprises usually define the 'how'. The task of representing requirements is not straightforward, and it requires subject matter expertise.

Laws declare properties that cannot be always easily translated into logical statements for the purpose of compliance validation. For example, the accountability principle in the Canadian Personal Information Protection and Electronic Documents Act (PIPEDA) states*: an enterprise is accountable for privacy violations, and should attempt to preserve the privacy of its clients*. This declaration may be partially redefined as: *an organisation shall designate an individual or individuals who are accountable for the organisation's compliance process; the compliance process validates privacy of individuals after each transaction*. This can be further implemented in more granular forms.

Some requirement extraction approaches are notation-based, and others are not. The approach presented in this paper is notation-based as are approaches using the notations URN [14], CREE [24], or UML.

URN, with its GRL and UCM components, can describe business processes together with goals as specified in requirements documents. It offers the advantage of traceability from requirements to process implementation. [7] proposes a compliance framework where legislation, policies, and business processes are modeled with URN. Traceability between various models and source documents is provided by using tools such as Telelogic DOORS. Their work, like ours, adopts the business process as a main artifact of access-control and provides a formal representation through Use Case Maps. Our method does not include goal specification, and has no specific mechanisms for traceability, and so URN has an advantage in this respect. However GRL's validation is limited to static validation of structural requirements. In comparison, we have the ability to combine formal logic with a representation of the business process concepts needed for the governance domain. We then offer the possibility of formal analysis and formal simulation of potential scenarios. Further, the GAL translation provides the semantics and provides the formalism needed for consistency and completeness checking.

[24] proposes a method, called CREE, that supports analysis of confidentiality requirements through goal models. Their analysis is complemented by goal modeling through semantic annotation of text from source documents. The annotated text represents concepts, which are used in the creation of goal models and subsequent analysis. Traceability between goal model elements and the text is established through annotations. Their work includes structural analysis based on OCL constrains. When compared to CREE, our method does not include annotations. However, as in CREE we have the ability to do structural analysis and to resolve terminology issues.

When compared to UML, we can say that our design artifacts are object oriented; however, our GEM offers a model view of the enterprise allowing for reasoning directly in terms of enterprise artifacts such as process, role, and activities.

Work that is not notation dependent includes [1][2] [5][6][21][23][26]. Some of these authors take a data-centric approach. Such approaches focus on the extraction process on electing access-control provisions to data elements.

A methodology is presented in [4] for directly extracting access rights and obligations from regulation texts. Similarly, the approach in [19] starts with the text analysis of the law by proposing a tool and methodology for extracting rights and obligations from legal requirements.

An ontological approach is described in [20], it categorizes the requirements by using an ontological model, thus helping to rigorously identify the inconsistencies between the model and the regulations. It captures and models the correlations between the attributes of certification and the accreditation requirements in the regulatory documents. Our approach uses ontologies for representing enterprise specifications.

The OMG Regulatory Compliance Alliance (recently renamed the GRC Round Table) has worked on standard representations of the regulatory documents. Their method aims at providing a dynamic mapping between the regulations and the organizational policies. Our method does not create a mapping: rather, it suggests combining the regulations with the organizational policies.

We conjecture, and we show in this paper, that a requirement extraction model is needed for an effective extraction process, and that this model must include enterprise process concepts.

III. TYPES OF REQUIREMENTS

Governance requirements are often composed of several elementary statements called *provisions*. Each provision can be composed. Similarly, an enterprise may have business requirements represented in several business policies. We refer to three meta types of requirements that dominate governance laws [10][22]; these types are considered to be at the meta level since they can be further specialized.

A. Meta-Types

There are three statement meta-types: Procedural, Declarative, and Ontology. However each type includes some logic operations.

1) Procedural Statements:

Requirements can be procedural statements, usually reducible to an *if-then-else* form. An example is the "Consent Principle-3" of the PIPEDA: *when an individual expresses a withdrawal of consent, the organisation needs to inform the individual of such implications*.

2) Declarative Statements:

Statements declaring facts, or system properties. Such statements cannot be represented directly in our compliance analyser without implementation refinement. A declarative provision can be an invariant, i.e. a property that must remain true. For example, the "Accountability Principle-1" in PIPEDA states *that an organization is responsible for personal information*. This invariant may be further refined into procedural statements found in the law itself. For example, *an organisation shall designate an individual or individuals who are accountable for the organization's compliance; these individuals should be assigned to the privacy audit process*.

3) Ontology Statements:

Legal requirements contain or imply many kinds of ontology definitions but we will only focus on two of these, namely organizational structure and process ontology, in addition to mapping related definitions.

a) Organisational structure ontology:

An ontology requirement can specify a structural element, e.g. *the Approve-credit department is required*. Another requirement can specify that certain people must be attached to it. For example, PIPEDA states that *the privacy officer role may be assumed by one or more individuals… and shall designate an individual or individuals who are accountable for the organization's compliance*.

b) Process ontology:

Process requirements describe obligations related to the sequencing of activities or the existence of processes or their hierarchy. For instance, PIPEDA states *that the audit process should belong to the privacy process*.

c) Ontology definitions:

This type of statement applies to either *process or organisational* ontology elements. Ontology definitions are dictionary definitions, which describe the meaning of a particular activity or equate terms used in the ontology. The enterprise can specify that *approve-credit is also referred to as review-credit department*. Another example in PIPEDA is: *individuals can give consent either by completing and signing a form, using a check-off box, or articulating consent orally when using a particular product or service*. This definition produces an equivalence mapping between ontology elements.

Logic operators in statements:

Legal texts use logic operators. The logic operators can relate to any of the above-mentioned types of requirements, such as enterprise ontology, group assignments or memberships. For example, *there exists a process for data-disposal*.

B. Specialisations

In this section we present subtypes inheriting from the meta types of section 3.1.

1) Access-Right statements (AR):

Governance requirements may define access-rights such as: *a project manager can be given access rights to project financial data*. Another example of a user right assignment might be *user A assumes the Loans process and is assigned to task Receive application, user B is assigned to process Credit-check*. An AR statement can be implemented as a procedure: *if condition then allow access otherwise deny*. Hence we consider AR statements as subtypes of procedural statements.

2) Delegation of Authority Rights (DOR):

An example of delegation of authority right could be the possibility for a role-R_1 to delegate its rights to another role-R_2 which indicates that a user is able to provide another user with access rights. A DOR statement is a specific kind of an access right statement. In this case the access right itself is the subject of access. Such statements also can be translated into procedural statements. Hence a DOR statement is considered as a subtype of a procedural statement.

3) Separation of Concerns (SOC):

Laws may also specify requirements for the separation of concerns. For example: *No data-sharing between marketing and customer service; or no member of the governance board can be a consultant*. SOC statements tend to be declarative. They also can be implemented using procedural statements: *if in conflict of interest group then deny access otherwise allow*.

IV. METHOD

We will now define a model-based extraction method to help extract legal requirements from the law. Our method is dependent on GEM, our UML-based requirement model, which provides a semi-formal representation of entities and their relations. This model helps explain the semantics of our extraction method from governance laws. The model is derived from our analysis of governance provisions taken from privacy and financial laws in Canada and the US. It is presented in Figure 3. The GEM serves as guidance towards:

A. Classification

The parts of the legal text need to be classified based on the principles of Section 3. Using the class model in Figure 3, a GO is able to extract activities, assignments, delegations, definitions, legal entities, processes, roles and separations of concern. Some classes in the model define assignment or sequencing relations such has belongs-to, delegate-to, etc. These relations are defined: *Roles-Activities, Processes-*

Activities, Activities-Activities, Processes-Processes, Legal Entity-Processes, and Roles.

B. *Refinement*

Composed or declarative higher-level classes may need refinement. *Provisions* may be translated to multiple procedural statements. This decomposition will be left to the experience and the knowledge of the GO. One can take the above mentioned accountability statement in PIPEDA as an example of a declarative that needs refinement.

V. Governance Extraction Model (GEM)

In Figure 3, we provide a combined requirements UML model which is our GEM. It shows the classes that we need and their relations. As mentioned, the GEM is not aimed at automating the extraction process; rather it is aimed at providing guidelines for translating requirements into formalised language. It represents our view of how requirements are defined, including relations and semantics. The description in the following sub-sections maps GEM model elements into the categories presented in section 3: ontology, procedurals, declarative, and other.

A. *Ontology Statements*

Ontology statements tend to be statements that combine logical operators and structural requirements.

User: The user class is related to the role class, since users act in roles. Users also assume processes. When this happens, users receive access to all of the process activities.

Activity: One or more activities that are part of a process and can be assigned to roles. An *activity* requirement may also suggest a particular sequencing. A procedural *activity* is a specific type of activity that includes a test followed by possible options. Atomic processes are *composed of* activities starting with a single initiating *activity*.

Process: The enterprise process is represented using a directed graph with possible loops and definite termination. In other words, each *process* must have both a starting and ending activity. A process may contain other *processes* or *activities*. A *process* class can define an *AssignedTo* relation. The *Process* class is a super-class and *Atomic Process* and *Composite Process* classes are its subclasses. Processes that are composed of one or more *Processes* are instances of the *Composite Process*. *Atomic Processes* are leaf processes that are composed solely of *activities*.

Department/Role: A hierarchy of roles and possible assignments to activities. A *Department/Role* includes sub-departments. For example, in the law there are references to the financial or the privacy divisions, requiring them to exist. Each Department/Role is usually assigned certain activities, such as signing financial statements.

Definition: For all the listed elements in this subsection namely: User, Activity, Process, Department/Role there may be definition type statements. These present equivalence relations where a class is declared to be equivalent to others.

B. *Procedural Statements*

Can be delegated (Delegation): Laws may specify role delegation rights. This is established in the can be delegated to relationship. This relation implements a right of delegating an activity from one role to another.

Separation of concerns: Processes in separation of concerns cannot be accessed concurrently by the same role or users. The separation of concerns requirements are represented through a relation between processes.

AssignedTo: A subject, such as a role or a group of users can receive an assignment to access an activity or a process. Activity assignments can implement access control rights and limitations. A user can be assigned to an activity or denied an activity.

Next (Activity Sequences): *Activity* sequences are able to implement path selection based on conditions.

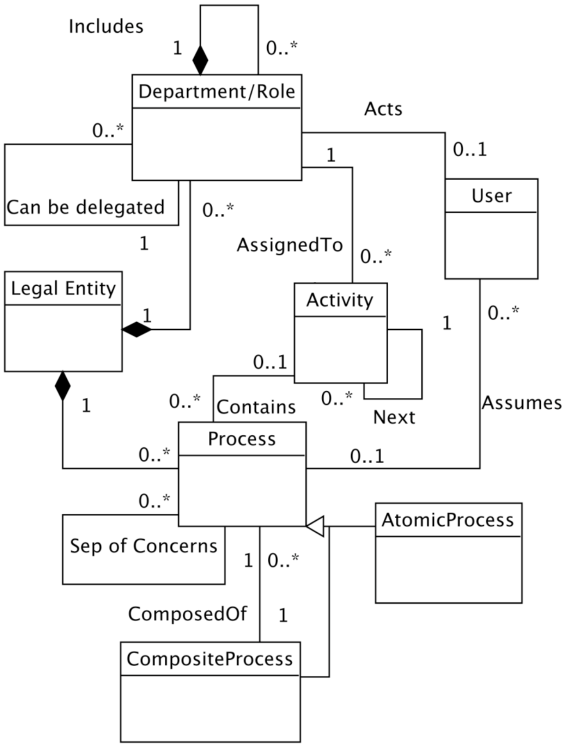

Figure 3. GEM-Governance Extraction Model

C. *Declarative*

A desired state of affairs rather than a process. Often stated as a responsibility: *An entity is responsible for implementing or following a specific process.* Declarative statements can be composite statements. They require an interpretation to fit one of the other types.

D. *Other*

Legal entity: A legally recognised entity. A *legal entity* class is usually helpful if the requirements include multiple parties. This class is here as a stub towards future work concentrating on inter-organisational validation. For reference, the entity class represents a named legal entity. Each legal entity is defined as a composition of departments and processes.

VI. Examples

In this section, we provide several examples taken from SOX, Instruments 52-111, and PIPEDA. Each example presents a text description; we then discuss the classes that we have matched with the text.

A. *SOX Example – USA*

SOX - Section.2 : Audit (3) AUDIT COMMITTEE. *The term ''audit committee'' means a committee (or equivalent body) established by and amongst the board of directors of an issuer for the purpose of overseeing the accounting and financial reporting processes of the issuer and audits of the financial statements of the issuer, and if no such committee exists with respect to an issuer, the entire board of directors of the issuer.*

Classification

Name	*Model match*	*Classification*
Audit committee (AC)	Role	Ontology-Org
Board of directors(BD)	Role	Ontology-Org
Issuer (IR)	Role	Ontology-Org
BD-AC, IR – BD	Includes	Ontology-Org
Audit Fin Statements(AFS)	Activity	Ontology-Process
AC-AFS	AssignedTo	Ontology-Process

According to our method we have classified the governance elements according to their matching to the GEM. The first column shows the name of the extracted element. The second column lists the matching class type in the GEM. The last column shows the classification according to our section 3.

B. *Instruments Example – Canada*

Canada's financial compliance law "Instruments 52-111" defines internal control audit report processes and structures using the following requirements.

Instruments 52-111: *Internal control over financial reporting is a process designed by, or under the supervision of, the issuer's chief executive officer and chief financial officer, or persons performing similar functions, and effected by the issuer's board of directors, management and other personnel, in order to provide reasonable assurance regarding the reliability of financial reporting and the preparation of financial statements for external purposes in accordance with the issuer's GAAP and includes those policies and procedures that: (a) pertain to the maintenance of records that in reasonable detail accurately and fairly reflect the transactions and dispositions of the assets of the issuer, (b) provide reasonable assurance that transactions are recorded as necessary to permit preparation of financial statements in accordance with the issuer's GAAP, and that receipts and expenditures of the issuer are being made only in accordance with authorizations of management and directors of the issuer, and (c) provide reasonable assurance regarding prevention or timely detection of unauthorized acquisition, use or disposition of the issuer's assets that could have a material effect on the annual financial statements or interim financial statements;*

Classification

Name	*Model match*	*Classification*
Internal Control over financial reporting (ICFR)	Process	Ontology-Process
CEO-ICFR, CFO-ICFR	AssignedTo	Ontology-Process
CEO, CFO	Role	Ontology-Org
ICFR-OACC, ICFR-FRP	ComposedOf	Ontology-Process
Overseeing accounting (OACC)	Process	Ontology-Process
Financial reporting process (FRP)	Process	Ontology-Process
CEO-OACC,FRP CFO-OACC, FRP	AssignedTo	Ontology-Process
Record Maintenance(RM), Validate Transactions(VT), Signatures (SI), acquisition(UA),use (US), disposition (DI)	Activity	Ontology-Process
ICFR-RM, VT, SI, UA, US, DI	Contains	Ontology-Process

C. *PIPEDA – Canada*

This example addresses the accountability principle in the Privacy Act.

Accountability Principle: *a designated privacy officer role is responsible and assigned to ensure privacy compliance.* We refine this requirement into several others.

Refinement
Need for a privacy audit process
Assigning a privacy officer to audit process
Create a privacy process assigned to Privacy Officer
The privacy process has sub-processes such as privacy audit, in addition to privacy governance to privacy reporting, and several others

Classification

Name	*Model match*	*Classification*
Privacy Audit (PA)	Process	Ontology-Process
Privacy Officer (PO)	Role	Ontology-Org
Privacy Process (PP)	Process	Ontology-Process
Privacy governance (PG), Privacy Reporting (PR),	Process	Ontology-Process
PP-PG,PR,PA	ComposedOf	Ontology-Process

In the PIPEDA example, we have shown the two stages of the method, namely the refinement by GOs of high-level requirements, in addition to the classification of the refined requirements.

VII. Conclusion

This paper presents a UML-based governance extraction model that can be used to extract and refine governance requirements. The method is part of an implemented legal compliance framework.

The novelty of this paper lies in the classification of legal requirements and in the abstraction of the governance model, in addition to its potential to be translated to a logic-based language for formal validation. Most importantly, this validation process has been shown to produce compliance validation results, as presented in [12] and in forthcoming papers.

We conjecture that the proposed method and GEM can be applied to other laws in the privacy and financial categories, possibly with appropriate adaptations.

Our future work in the RE domain shall be three-pronged: Study of conceptual legal models; Comparative study of other extraction approaches based on URN, CREE, and others; analysis of the completeness of the proposed extraction method.

Acknowledgment

This research was funded in part by the Natural Sciences and Engineering Research Council of Canada.

References

[1] Ashley, P. (2004) A privacy logging and reporting framework. APITSC. University of Queensland.

[2] Balust, J. M., Franch, X. (2001). Building expressive and flexible Process Models Using a UML-Based Approach. LNCS, vol. 2077. Springer-Verlag, London, 152--172.

[3] Bourcier, D. (2004) Legal Knowledge and Information Systems: JURIX 2003. IOS Press, 2004.

[4] Breaux , T., Antón, A. (2008). Analyzing Regulatory Rules for Privacy and Security Requirements. IEEE Trans. Software Eng. 34, 1 (Jan. 2008), 5--20.

[5] Brodie, C. A., Karat, C., Karat, J. (2006) An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. SOUPS '06, ACM. Proc. Series Vol. 149, 8-19.

[6] Fischer-Hubner, S., Otto, A. (2008) From a Formal Privacy Model to its Implementation. 21st NISSC. Arlington, VA, 5–8 October 5-8, 1998.

[7] Ghanavati S., D. Amyot, L. Peyton. (2007) Towards a framework for tracking legal compliance in healthcare. CaiSE'07. 218–232. Springer, 2007.

[8] Giblin, A. Y. Liu, S. Müller, B. Pfitzmann, X. Zhou. (2005) Regulations Expressed as Logical Models (REALM). 18th JURIX 2005. Brussels.. 37-48.

[9] Graham N., Statutory Interpretation: Theory and Practice. Emond Montgomery Publication, 2001.

[10] Hassan, W., Logrippo, L. (2009) Requirements and compliance in legal systems: a logic approach. Requirements Engineering and Law, 2008. RELAW '08. , vol., no., 40-44, 9-9 Sept. 2008.

[11] Hassan, W. (2009) Validating Legal Compliance, Governance analysis method. Doctoral Thesis. University of Ottawa..

[12] Hassan, W., Logrippo, L. (2009) Validating legal compliance, Submitted,.

[13] Multilateral Instrument 52–111- Reporting on internal control over financial reporting. (2005) 28OSCB 1317.

[14] ITU-T: User Requirements Notation (URN) – Language Requirements and Framework. ITU-T Recommendation Z.150. Geneva, Switzerland, February 2003.

[15] Jackson, P., Al-Kofahi, K., Tyrrell, A., Vachher, A. (2003). Information extraction from case law and retrieval of prior cases. Artif. Intell. 150, 1-2 (Nov. 2003), 239-290.

[16] Jackson D. Software Abstractions: Logic, Language, and Analysis. MIT Press. Cambridge, MA. 2006.

[17] Jarke, M. & Kurki-Suonio, R. (1998). Guest Editorial. Special issue on Scenario Management. 1EEE Transactions on Software Engineering, 24(12).

[18] Kerrigan, S., Law, K. H. (2003). Logic-based regulation compliance-assistance. ICAIL '03. ACM, New York, NY, 126-135.

[19] Kiyavitskaya, N., Zeni, N., Breaux, T. D., Antón, A. I., Cordy, J. R., Mich, L., Mylopoulos, J. (2007). Extracting rights and obligations from regulations: toward a tool support tool supported Process. ASE '07. ACM, NY, 429-432.

[20] Lee, S. W., Gandhi, R. A. (2005). Ontology-based Active Requirements Engineering Framework. APSEC. IEEE Computer Society, Washington, DC, 481--490. 2005.

[21] Li, N., Yu, T., Antón, A.I. (2003).A semantics-based approach to privacy languages. CERIAS Technical Report TR 2003–28. Purdue University, Nov. 2003.

[22] Logrippo, L. (2207) Normative Systems: the Meeting Point between Jurisprudence and Information Technology? In: H. Fujita, D. Pisanelli (Eds.): New Trends in Software Methodologies, Tools and Techniques – Proc. of the 6th SoMeT_07. IOS Press, 2007, 343-354.

[23] Mitra, P., Pan, C., Liu, P., Atluri, V. (2006). Privacy-preserving semantic interoperation and access control of heterogeneous databases. ASIACCS '06. ACM Press, New York, NY, 66-77.

[24] Onabajo A. (2009) Analysis of Multilateral Software Confidentiality Requirements. PhD Thesis. 2009.

[25] Personal Information Protection and Electronic Documents Act (PIPEDA). Second Session. Thirty-sixth Parliament. 48–49 Elizabeth II. 1999–2000.

[26] Roessler T, Wenning R. (2007) Challenges for Privacy Policy Languages. W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement..

[27] Sarbanes-Oxley Act of 2002. Second session. One hundred seventh congress of the United States of America. 23 January 2002.

[28] Wintgens L. (2007) Legislation in Context: Essays in Legisprudence. Ashgate Pub. 2007.

[29] Yao-Hua Tan, Leendert W. N. van der Torre. (1995) Why Defeasible Deontic Logic needs a Multi Preference Semantics. ECSQARU 1995: 412-419.