ENTERPRISE ARCHITECTS CONCERN LEGAL REQUIREMENTS FOR THE COMPLIANCE WITH THE LAW

Vytautas Čyras, Reinhard Riedl

Docent, Vilnius University
Faculty of Mathematics and Informatics, Naugarduko 24, 03225 Vilnius, Lithuania
Vytautas.Cyras@mif.vu.lt; http://www.mif.vu.lt/~cyras/

Director, Research and Services, Bern University of Applied Sciences, Bern, Switzerland
Morgartenstrasse 2c, Postfach 305, 300 Bern 22, Switzerland
Reinhard.Riedl@bfh.ch; http://wirtschaft.bfh.ch

Keywords: legal informatics, enterprise architecture, legal machine, transparency engineering, legal visualization

Abstract: This paper addresses enterprise architecture (EA) compliance framework. A methodology to check EA non-compliance with laws and regulations is still a challenge. We think that a compliance methodology should take into account “shared” relevant laws and a requirements engineering framework. We continue discussing the view of enterprise architects on legal informatics which was proposed by Reinhard Riedl. We reflect mainly on two articles: Riedl (2011) and Čaplinskas (2009) who proposes a vision driven approach on requirements elicitation in the context of enterprise engineering.

1. Introduction

The message of this paper is that a requirements engineering framework could contribute to a methodology for compliance in enterprise architecture (EA). Enterprise architects concern legal requirements when building enterprise systems. Different legal issues are concerned in every perspective of EA. Legal informatics integrates the perspectives of different stakeholders. The enterprise architect’s perspective has the task to integrate all the different views on enterprise architectures, in particular, the business view, the ICT view, and the legal view [Riedl 2011]. Fig. 1 shows the key terms within the point of departure. Our reflections on the concepts are further.

Fig. 1: The key concepts tackled in this paper
One of the problems right now with EA (cf. Enterprise Architecture as Strategy [Ross, Weill & Robertson 2006]) is that for reasons of simplicity those practically usable in real life focus on very few aspects of a real world IS, usually issues which are shared throughout the organization. Depending on the operating model of the organization, this is just technology, or in addition data and/or business processes. In the spirit of EA in the sense of Ross, Weill & Robertson it would make sense to define the “shared” relevant laws and integrate them in Fig. 1. “Companies are buffeted by constant changes in regulations, such as Sarbanes-Oxley, Basel II, and HIPAA. As companies become global, they become accountable for increasingly complex reporting requirements. …Companies may not be able to anticipate new regulations, but they can increase the likelihood that needed data is readily available or can easily be accumulated.” [12-13]

Enterprise architects check the architecture for compliance. Potential conflicts have to be identified. The regulations which influence enterprise architectures, perhaps Sarbanes-Oxley Act, can be barely aware [Riedl 2011, 266-267]. The following relationships can be identified:

• Architecture descriptions have to leverage checking the compliance with the law.
• Legal informatics experts are able to contribute to legislation, especially in e-Government.
• Enterprise architects become important partners for legal informatics experts. This is possible in the revision of the law, e.g. in digital identity regulations.
• Contacts with authorities when anticipating ICT perspectives. Regulation of software exchange, e.g. modules in finance informatics.
• Ideas in legal informatics; patterns and anti-patterns.

![Figure 2: The context of legal machines: machine = analogy of human on the Is stage](image)

**Information systems as legal machines.** Enterprise engineering deals with three kinds of systems – business, information, and application systems. An enterprise information system (IS) is a whole formed out of organisational memory and sets of information processing actors (IPA), information flows, and interrelated information processing processes implemented in accordance with the enterprise information processing policies and standards [Čaplinskas 2009, 349-353]. IS can be viewed as a type of legal machines; see [Čyras & Lachmayer 2011]. The complexity of interaction
is a criterion to distinguish the types such as vending machines or intelligent agents. The effects of information systems have legal importance. Decision makers in workflows are comprised of human beings and machines. Data input to a workflow is a legally binding act. The key feature of legal machines is *impositio*. Thus they are related to the Ought world. The actions of legal machines are treated as legal (institutional) facts and posit obligations and rights. Examples are traffic lights, vending machines, information systems such as FinanzOnline in Austria, etc.

**Situative law versus general law. Statutory law, customary law and machine law.** On the Is stage (Fig. 2), customary law and machine law are in the foreground. As an example suppose a zebra crossing and pedestrians which aim to cross the street. The situation is regulated by statutory law (the road rules). However, ordinary people are governed primarily by customary law which superimposes statutory law. And finally this situation is governed by traffic lights – machine law steps in.

The stage – the Is world – is the place where the manifestations of law take place. The façade defines the legal order (the Ought world), while the stage is the venue where order, interests and material facts collide and create facts. In the future research it could be worthwhile to depict the different types of norms other than law which play a role on stage, or to draw a picture, where the forces active on stage are indicated. This can then be used to create an innovative perspective at requirements engineering. In fact, one output of this research should be to develop a new understanding of requirements engineering, which is based on this artificial stage for legal machines.

### 2. The legal perspective in legal informatics and enterprise engineering

Three central perspectives are concerned in [Riedl 2011]:

1) business perspective;
2) ICT (information and communication technologies) perspective;
3) legal perspective.

Other potential perspectives are:

4) internal communication perspective (a direction tool);
5) public relations and marketing perspective;
6) political and economic perspective (perhaps in e-Government).

A conclusion is that the only true purpose of the work of enterprise architects is transparency optimization in an organization.

In our research we would like to focus on the legal perspective though even more perspectives are concerned in the Zachman framework [Sowa & Zachman 1992]. Zachman’s idea to decompose the system into a number of perspectives and focus areas has served as a theoretical basis for the vision driven approach proposed by Čaplinskas (2009). Zachman decomposes each perspective into six focus areas to be answered: what (data)? how (function)? where (network)? who (people)? when (time)? and why (motive)?
Fig. 3: The methodological framework for requirements elicitation, analysis, specification and validation proposed in [Čapliškas 2009, 354]

Five perspectives (views, levels) are shown in Fig. 3:

1) business level requirements (the view of business analyst);
2) user level requirements (the view of stakeholders);
3) IS (information system) requirements (the view of IS analyst);
4) the requirements of IS subsystems (the view of IS engineer);
5) software requirements (the view of software analyst).

The Zachman framework is an architectural one. Čaplinskas notes about the five perspectives:

“To be complete, it should additionally include the requirements of software components (the view of software architect), the implementation requirements (the view of software engineer), the process requirements (the view of process engineer), and the testing requirements (the view of tester). …The first five perspectives differ from corresponding ones provided by the Zachman’s framework because they are designed for different purposes.” [p. 355]

The tight integration of activities in the context of enterprise engineering, strategic planning, information systems engineering, and software engineering can be emphasised: “Traditional, interview-based requirements gathering and elicitation techniques are suited for this aim not enough well and often lead to the violation of the strategic alignment.” [p. 343]

We aim to treat a complex legal machine as an enterprise system. Therefore we stick to a requirements engineering methodology. Each perspective (level) is a model of the system. Legal issues are addressed in every system life-cycle phase. Even the life-cycle *per se* is subject to technical standards. The concepts of a to-be-system-to and requirements are related to law. The requirements document (system specification) is part of the contract with a customer. Every requirement is based on a norm. This norm is present in a technical standard, business rule or other kind of legal source. Every phase of system’s life-cycle concerns the requirements. However the term ‘requirement’ is treated differently during each phase. The flowdown of requirements to lower levels makes them more detailed. The difference in the nature of requirements stems from the difference in the nature of the norms.

3. The task to be solved by a multiphase transformation

**Multiphase transformation**. Friedrich Lachmayer proposed the metaphor of legal informatics as a bridge from law to informatics. The bridge consists of multiple arches. Thus a multiphase transformation emerges and a legal text-to-program approach is identified. The transformation is about building a “bridge” between legal requirements and information system (Fig. 4).

![Fig. 4: The metaphor of multibridge. Multiple transformations lead from norms in legal texts to their machine operationalization [Čyraz & Lachmayer 2012]](image)

The idea of multiphase transformation was first presented in [Čyraz & Lachmayer 2012]\(^1\) where four phases (arches) have been identified:

Phase 1. Microcontenting: from legal text to textual microcontents

---

\(^1\) Submitted to *Takashi, I. & Mori, H.* (eds.) Liber amicorum Guido Tsuno, Chuo University Hachioji Tokyo, Japan.
Phase 2. Visualising: from textual microcontents to visualization

Phase 3. Formalising: from visualization to formalization

Phase 4. Representing: from formalization to representation in a technical language

Legal informatics is about the bridge. However different actors focus on its different elements. Thus the importance of two banks (law and informatics), piers and abutments, or arches can be emphasized differently. Material (‘what’) or procedural (‘how’) can be in the focus.

Taking into account multiple perspectives on information system architectures, multiple bridges or a network of bridges between law and informatics can result. A way from legal texts to legal machine implementation is not as simple as might be supposed by the following example. In order to build a traffic light, take road rules and think about a device to replace policemen. In the case of an information system a long way leads from legal requirements to the specification. Besides the requirements, other legal issues can be concerned. Information systems are in the context and their lifecycle can tackle the whole legal system.

In order to speak about building a multibridge, the problem – task – has to be identified. The multibridge would serve as a means to accomplish this task.

When speaking about enterprise information systems, law and informatics, we concern the IS development task. Other tasks can be examined but are out of scope of this paper:

- **Writing a law** to regulate the domain which involves the legal machine. This is a lawmaking task. An example is to write rules for an eGovernment application.

- **Legal assessment** of the law. Jurists are in the centre. The recommendations to improve both the legal machine and the law serve as the outcome.

In the IS development task engineers are in the centre. Jurists are involved until the requirements specification is written. Then the software process continues with the design, implementation, testing and deployment phases. Goals have to be specified. Events which trigger legal acts are implemented in machine states. However, a key problem in this approach is that successful requirements engineering in practice requires iterative adaptations of the a priori defined requirements as the development of the detailed design and its implementation proceeds. Thus there is the risk that the legal perspective gets lost in the process.

The list above is not exhaustive and other tasks can be identified. High level terms in law are comprised of legislation, execution and adjudication. Other frameworks can consider lower level functions of law. An individual problem can be divided into sub-problems of different types.

### 4. Towards a methodology of the compliance with the law

A good practice for the compliance of enterprise architecture with regulations is still a problem. Riedl (2011) concludes with theme aggregates in order to shape the integration of different recourses compliant with the law:

1. Internal arrangement of transparency
2. Methods for the legal architecture view as part of enterprise architecture
3. Design methods for law-triggered changes in the enterprise architecture

Requirements are in the focus in the Čaplinskas’s vision driven approach. We think that it could provide a framework to shape the methods above. A legal perspective, i.e. the requirements to compliance with the law, could be added. However we can hardly imagine an analyst who can be aware of legal norms in different branches of law. Corporate governance code is not the sole source.
Therefore methodologies are needed to formulate legal requirements and to analyse them for compliance. A way might be simply to check the requirements in each cell in Fig. 3 for compliance. This would be classified as an ex-ante solution [Bonazzi et al. 2009] “to design an artefact aimed at avoiding actions that are not compliant”. However to lower the risk of violating strategic alignment, a holistic approach has to be undertaken. Ex-post solutions “to design an artefact to assess the level of compliance” have to be taken into account, too.

Enterprise information system comprises people, i.e. information processing actors. Their behaviour is also the object of norms. IS legal requirements can be formulated in the way that people observe specific norms from different branches of law.

A holistic view requires a “Governance, Risk and Compliance”\(^2\) approach and involves three loops of management cycles [Bonazzi et al. 2009]. We think that in order to assess the IT GRC in practice, an experiment should be undertaken. An assessment according to theoretical criteria is feasible, for instance, in formal theories but is not enough in management. The reason is that the latter is a more practical rather than theoretical activity. Not all the criteria can be foreseen in advance. Each case can bring specific factors.

The IT compliance framework proposed in [Bonazzi et al. 2009] is worth a special attention. Two dimensions, Legal Dimension and IT Dimension, and two kinds of sources of regulation to comply with, External and Internal, are depicted with a square, see Fig. 5. Distinguished alignments are represented with arrows. The direction points to a defined artefact.

Every concept in Fig. 5 denotes a broad field. For example, corruption is one but not the only violation of law. Different laws regulate distinguished kinds of liabilities such as civil, administrative and criminal. The meaning (Sinn) of law – the Ought realm – can hardly be understood from the sole legal text. In order to analyse for the compliance with the laws, one

---

The design of law-compliant information systems is an area on a lower level of abstraction, though falls within the framework above. Knackstedt et al. (2011) note that early works in this area were lacking in representing legal requirements. Therefore they propose a framework to model the area which is structured in three dimensions: (1) field of law (flow control, reporting, web applications, etc.), (2) modelling level (analysis level, model level, metamodel level), and (3) research goal (explanation- or design-oriented research) [Knackstedt et al. 2011].

An interesting perspective for the actual implementation of such a holistic approach is provided by agile software development [Beck 2001]. Agility in IS solution development was first introduced in the extreme programming (XP) movement. It is nowadays extended to many IS related tasks and applied even in development and design tasks without much IS relationship but with a general multi-disciplinary and multi-stakeholder challenge. The key two ideas of agility is to involve all expertises needed for a good IS solution in the team working on the solution and to develop the solution in short cycles, where at the beginning of each cycle a requirements analysis takes place and at the end of each cycle a working (running) in-between-product is delivered to users. This original concept has been adapted to more conceptual contexts like enterprise architecture management, where no running products can be delivered. In the original setting the key challenge is to design a good basic overall structure, as the spirit of agility contradicts the planning. However, in principle agility proposes an interesting split for a holistic approach. The integration of multidisciplinary perspectives is split into a very rudimentary strategic core concept and a concrete team-work, where requirements analysis and holistic solution development is done iteratively.

Acknowledgements. We are grateful to Friedrich Lachmayer who encouraged this joint work.

5. References


Čyras, Vytautas; Lachmayer, Friedrich, Multiphase transformation in the legal text-to-program approach. Submitted to: Izumo Takashi & Hikaru Mori (eds.) Liber amicorum Guido Tsuno. Chuo University Hachioji Tokyo, Japan (2012)


