A Meta-model for Legal Compliance and Trustworthiness of Information Systems

Fatemeh Zarrabi, Michalis Pavlidis, Haralambos Mouratidis, Shareeful Islam, and David Preston

School of Architecture, Computing and Engineering, University of East London
s.zarrabi@uel.ac.uk, m.pavlidis@ieee.org,
{haris,shareeful,david}@uel.ac.uk

Abstract. Information systems manage and hold a huge amount of important and critical information. For this reason, information systems must be trustworthy and should comply with relevant laws and regulations. Legal issues should be incorporated into the system development process and there should be a systematic and structured assessment of a system's trustworthiness to fulfil relevant legal obligations. This paper presents a novel meta-model, which combines legal and trust related concepts, to enable information systems developers to model and reason about the trustworthiness of a system in terms of its law compliance. A case study is used to demonstrate the applicability and benefits of the proposed meta-model.

Keywords: Hohfeld taxonomy, natural language pattern, legal constraint, trustworthy information systems, trust modelling, control.

1 Introduction

Information systems in the modern world exist in every aspect of human life. Governmental organizations, factories, and hospitals (to name few) deploy such systems to manage huge amount of sensitive and critical information. As such, security of such information systems is of paramount importance. Any security failure of those systems can cause potential losses of money, time, or even life. In such cases liability of information systems is assessed and information system owner should be constituted responsible to replace damages [1]. To this end, information systems should comply with relevant laws. However, information system developers face two main challenges. Firstly, developers need to capture requirements from legal texts, align them with other system requirements and assign them to relevant system components. Secondly, to ensure law compliance, system developers need to place trust and rely on human actors and software components. The trustworthiness of such actors and components to achieve their legal duties needs to be assessed properly during the development of a system. Otherwise, if these actors and components are not trustworthy, they can possibly harm the ability of a system to fulfil legal obligations and be law compliant. However, the current literature fails to support

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 46–60, 2012.

information systems developers with adequate practices and methods to face those challenges. On one hand, although there is some work to support capturing of requirements from legal texts (see section 5 for more information), the literature fails to provide clear evidence of appropriate frameworks and methodologies to support the capture and analysis of system requirements from relevant laws and regulations. On the other hand, there is lack of frameworks to support the analysis of trustworthiness, within the context of law and regulation compliance, at the requirements engineering stage.

This paper presents a first step towards the development of a novel framework that overcomes the above problems by combining concepts from trust engineering and regulatory requirements capturing. In particular, the paper presents a meta-model that enables developers to model legal requirements during information systems development and reason about the trustworthiness of the actors (human or software components) who are assigned and responsible for the fulfilment of those requirements. The proposed meta-model is based on legal concepts such as legal constraints, duties and rights [2] that are assigned to actors, and trust-related concepts, such as experiential, reported, normative, and external trust, and control [3] over the enforcement of duties.

The paper is structured as follows. In section 2 we describe relevant legal and trust related concepts. The meta-model is presented and discussed in section 3, while in section 4 we demonstrate the applicability and benefits of the proposed meta-model with the aid of a case study. In section 5 we discuss related work and section 6 concludes the paper.

2 Legal and Trust Concepts

Laws, regulations and policies related to an information system need to be incorporated into the system development process since they constitute restrictions on the system. Laws and regulations use very high-level language and technical terminologies, which represent legal instructions within obligations and recommendations. Being unfamiliar with technical language of laws has made considerable challenge for system developers to understand laws and indeed to identify the stakeholders and the action of law. To overcome this problem, current work is taking advantage of a legal taxonomy called Hohfeld [4] along with the aid of some natural language patterns. Hohfeld analyses laws by separating them into two main groups of legal relations between individuals. The first group indicates legal respected choice of individuals and is called *Right.* Based on Hohfeld, *Right* is paired with a correlative called *Duty*. Duty is the second category of legal relationship introduced by Hohfeld and indicates which one ought or ought not to do. Therefore, one person's right for an action against another entails the *duty* of the other against the first person in respect of that action and vice versa. Right and Duty, as the main two groups of legal relations, also contain four subgroups that inherit correlative relations from their parent group (Table 1).

Table 1. Hohfeld Legal Rights and correlative Duties

Right	Claim	Power	Immunity	Liberty
Duty	Duty	Liability	Disability	No-Right

Claim is entitlement of a person to have something done from another person and it correlates to duty. For example a contract between employer and employee confers on the employee the right to be paid his wages, which he/she can claim for this right and it correlates to the duty of employer to pay the wages to employee. Liberty is one's freedom from the right or claim of another and it is paired to the correlative of no-right. Suppose that people are free and have the right to smoke in an open environment, therefore no one has a right to prohibit them from smoking. Also, power is one's affirmative control over a given legal relation as against another. For example, a librarian has the power over a student with regard to the use of the library. Normally the student has the right to use the library, but if he is noisy the librarian has the power to take away that right and stop the student from using the library. Immunity is one's freedom from the legal power or control of another. For example, diplomats are supposed to have diplomatic immunity, which means that if they commit a crime in the hosting country, they are immune against arrest and legal prosecution. In other words, the hosting country police are disabled to act [4]. In legal documents, duties and rights of stakeholders are expressed using specific modality notations, for instance 'shall', 'must', 'may' (Table 2).

Table 2. Modality Notations

Duty	Shall, must, shall be, must be, shall prohibit, may not, shall not, must not, is required to
Right	May, may be, shall guarantee the right, has the right to

As seen in Table 2, modality notations are divided in two different categories based on the extent of control they enforce on the action. To identify modality notations, our approach employs a language pattern called Modality Pattern [5].

The phrase "Language Pattern" refers to specific samples related to linguistic typology used in language grammars in order to identify different units in a sentence (Table 3). To identify other elements of law such as subject, verb and object we use basic activity pattern. Whereas our basic activity pattern is following SVO (Subject-Verb-Object) sentence structure where the subjects come first, verb second, and object third [6]. Therefore the process of using these patterns can be used for most of legal languages. The difficulty of this activity to identify these elements from a sentence is related on the complexity of the sentence itself. Sometimes a sentence simply consists of basic activity pattern elements (subject, action, object) or the object itself consists of one or more other sentences. Noun phrases such as "who", "which" and others give more details on the above mentioned elements and are used to identify scope of law or to extract extra requirements from law. Conditional, exceptional and purpose patterns also extract extra requirements of system since they restrict the action of law. They are mandatory requirements of system since they are enforcing some specific circumstances on duties and rights.

Table 3. Natural Language Patterns

Language Pattern	Text's element	Meaning	Identifying phrases
Basic activity pattern	Subject	Who performs the action	-
Basic activity pattern	Object	What the action is performed on	-
Basic activity pattern	Action	What is performed	-
Noun phrase pattern	Target	Action is performed on whom	To, on, of, from, …(It can also be identified from the concept of object or subject)
Purpose pattern	Purpose (goal)	Why the action is performed	To, in order to, …
Condition pattern	Condition	When and in which condition action should be performed	If, when, whenever, …
Exception pattern	Exception	When and in which situations action should not be performed	Except when, except that, is except from, …
Modality pattern	Modality phrases	If the action is required, or recommended to be performed	Must, shall, may, may not, must not, is required to, has a right to, …

In order to perform compliance of modelled laws on the desired system context, we need to map extracted elements from laws to that system context. For this purpose, we adopt the i* modelling language [7]. I* models social relationships between stakeholders of a system environment using the concepts of actor, dependency, goal, plan and resource. In i* actors depend on each other to achieve a goal, carry out a plan or deliver a resource. These concepts are useful in order to represent the relationships between the different stakeholders. However, we extend i* with law related concepts in order to represent the legal relationships, and with trust related concepts in order to reason about trust. The rest of this section provides a brief description of the adopted i* concepts [8].

Actor. Actor is an entity of the domain of interest that possesses strategic goals and can carry out actions that will fulfil those goals. It is a unit that encapsulates intentionality, rationality, and autonomy.

Goal. A goal is a condition that an actor wants to achieve. Of course, this can be done in more than one way so alternatives of achieving a goal can be considered.

Plan. A plan is a procedure that has to be followed in order to accomplish a goal and specifies the way of achieving that.

Resource. A resource is an informational or a physical entity that is needed to accomplish a goal or to carry out a plan.

Dependency. A Dependency is a relationship between two actors. In this relationship one actor is the depender and the other actor is the dependee. The depender is depending on the dependee to satisfy the dependum, which is the object around which the relationship centres. The dependum can be a goal, a plan or a resource.

3 Law and Trust Meta-model

Figure 1 shows the meta-model that combines both law and trust related concepts. The concepts with orange colour are the law related concepts [2] and these concepts are linked with the trust concepts [3] with the grey colour.

Secure Goal. Secure goal is the strategic interest of an actor with respect to security.

Legal Constraint. Legal constraint is the restriction related to legal issues such as when an actor is required and instructed to comply with a law in order to achieve a goal, perform a task or receive a resource.

Duty. Duty is something that one is expected or required to do by legal obligation. It covers Hohfeld's obligations of duty, liability, disability and no-right since they emphasise on actions that are required to be done by actors and are not optional. A Duty *correlates* to a Right of target party of law having a mutual and complementary relationship and based on Hohfeld taxonomy who believes duty of a stakeholder cannot exist without right of another party. A Duty will be satisfied with a goal or secure goal using Mandatory satisfy link. This will enforce the necessity of the fulfilment of the goal or secure goal since they are ordered by law and cannot be refused or ignored.

Right. Right is something that one is allowed or recommended to do or owned by law. It covers Hohfeld rights of claim, power, immunity and liberty since they represent optional decision of actors to perform an action or not. Right of a stakeholder also correlates to Duty of other party based on Hohfeld taxonomy. A Right can be satisfied by goals and secure goals taken by the correlated duty-bearer or it may be satisfied by goals and secure goals which are needed to be taken by the right-holder himself. This is due to the reason that a right may always not be satisfied by a duty.

Resolution. Resolution is the indication of how the uncertainty in the fulfillment of a duty dependency is removed in order to build confidence in that duty dependency. There are two types of resolution, i.e., trust and control, which the developer uses to feel confident about the dependency.

Trust. Trust is the positive expectation of one actor about the behaviour of another actor by whom he might be positively or negatively affected [9]. The actor who trusts another actor is called trustor and the actor that is being trusted is called trustee. Trust can be decomposed into four types. They are:

- **Experiential Trust.** Experiential trust is trust that originates from previous direct experience with the trustee. The depender then is actually depending on himself.
- **Reported Trust.** Reported trust is trust that originates from a third party (the reporter) who reports that the trustee is trustworthy. Therefore, depender depends on the reporter to trust the dependee. As a result, reported trust creates an indirect trust relationship with the reporter. This new indirect trust

relationship is required in order to support the direct trust relationship with the actor that has duties.

- **Normative Trust.** Normative trust is trust that originates from the system environment norms. The depender is then depending on the system norms.
- **External Trust.** External trust is trust that originates from sources outside of the system environment. The depender is the depending on the external source of trust.

Trust Relationship. Trust relationship is defined as a relationship that exists between the trustor and the trustee and resolves a dependency based on trust. There can be direct and indirect trust relationships. Direct trust relationships are the trust relationships with actors that are responsible for fulfilling duties, while indirect trust relationships are trust relationships with actors that exist in order to justify the direct trust relationships or control relationships.

Control. Control is the power of one actor to enforce the fulfilment of a duty by another actor. This eventually means that the actor has the ability to gather information about another actor and also the ability to influence the other actor's present and future. In other words, it is the ability to influence the other actor's goals. When the type of resolution is control then there is a third party who acts as a controller. In particular, when there are legal constraints, actors are required to possess duties in order to satisfy these legal constraints. In this case the dependency on actors to achieve the duties has a control type of resolution that means that a third party acts as a controller. So, there is a new indirect trust relationship with the controller that supports the control relationship.

Entailment. Entailment is a trust assumption and needs to be examined if it is true or not. These assumptions need evidence in order to be justified. For example, if there is an entailment that a certain employee is trusted, is there any evidence that supports this assumption? If the outcome is positive then there is confidence that the actors will fulfil their duties. The system then satisfies the legal constraints and it is trustworthy in terms of law compliance Otherwise, there are possible vulnerabilities that can lead to legal breaches.

The initial step of using the meta-model is to consider the correlative theory of Hohfeld to find all relevant relations of stakeholders. Based on correlative theory of rights, when a stakeholder of a law has a right, the opposite party has the duty against him and vice versa. Our main consideration is on the function of rights and duties. In other words, on what the rights and duties bring to their holders from the requirements point of view. We achieve this goal by correlating a Right with a Duty and satisfying the duty with goals and secure goals. Also a Right can be satisfied by a goal or secure goal since there are cases that rights cannot be satisfied with goals taken by correlative duty-bearer and they need to be satisfied with goals that right-holder takes. We are also representing Duty and Right relationships with the aid of Dependency concept from i*. Therefore right-holder and duty-holder depend on each other or on other actors in order to perform a duty or claim for a right. The reason behind the use of the concept of Dependency is that in a system environment an actor can never work

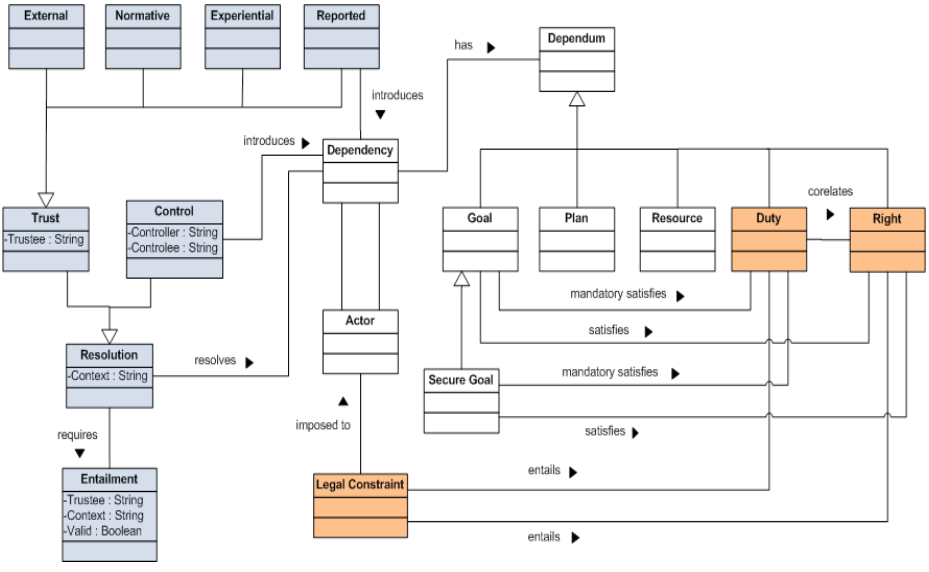

Fig. 1. Law and trust meta-model

without a dependency on other actors and there is always cooperation and relationship between two or more actors, for each actor to satisfy his goals. Also, we cannot ignore the consideration of legal rights in the design of a system since this is important to emphasize on the mandatory of the legal rights. Therefore, the system developer should be precisely instructed on actions that he/she is required to do (duty) and has the option to do (right).

Nevertheless, there is no guarantee that the actors assigned with duties will fulfil them. Further analysis of whether such actors are trusted to fulfil their duties is required in order the developer to be able to make informed and justified decisions during the development process. Verifying that actors are indeed trusted will remove any uncertainty and build confidence that the developed system will be trustworthy and law compliant.

As said before, an actor wants to achieve goals, carry out plans and deliver resources. However, there are legal requirements, which are represented as legal constraints, and are imposed to an actor in order to restrict the achievement of one or more of the actor's goals. To satisfy the legal constraints, the actor needs to achieve duties. At this point the trust-based concepts are introduced in order to be able to reason if actors are trusted to achieve duties, in order to ensure the law compliance of the system. There is an uncertainty whether the actors are willing to achieve duties, so trust and control are used to resolve the dependencies on such actors. The dependencies on actors to achieve duties require a resolution that specifies how the confidence is built. The resolution can be through trust and/or control. Trust and

control reduce the uncertainty and increase the confidence in the actors. Nevertheless, the resolution of dependencies on actors with duties create entailments, which are conditions of trust that need to be validated in order to have confidence that actors will achieve their duties that will ensure the law compliance of the system.

4 Case Study

For the purpose of demonstrating the usefulness of our approach we will use a case study based on Dropbox. Dropbox [10] provides file-hosting services to internet users over the web. The files are stored and shared in cloud storage and particularly in Amazon's S3 storage system. Also, Dropbox collects and stores the files that are being uploaded by the users, information about the device used, its software, and user activity. Dropbox is required to comply with DMCA (Digital Millennium Copyright Act of United States) [11], which establishes a notification-and-takedown system for addressing claims of copyright infringement. Analyzing the system environment of the Dropbox application there is a number of different actors involved. These are:

- Dropbox Engineer. Engineer is responsible for maintenance of Dropbox service and the datacenter.
- User. The internet user of the Dropbox service.
- Amazon S3. The cloud service that Dropbox is using to store files.
- Legal Team. Responsible for the legal issues of Dropbox.
- Dropbox System. It is the technical system of Dropbox service.

Based on the system context of this scenario, we only consider the actors relevant to Dropbox uploading system. Also, size of this paper does not allow the practice of all related laws and duties and rights. There is a legal constraint that Dropbox technical system needs to comply with Digital Millennium Copyright Act (DMCA). Therefore there is number of duties and rights assigned to the Dropbox system in order to comply with this Act which are extracted from the following clauses.

1) DMCA.SEC.103 INTEGRITY OF COPYRIGHT MANAGEMENT INFORMATION

(a) FALSE COPYRIGHT MANAGEMENT INFORMATION. —No person shall knowingly and with the intent to induce, enable, facilitate, or conceal infringement distribute or import for distribution copyright management information that is false.

2) DMCA.SEC.202. INFORMATION STORED ON SERVICE PROVIDERS.

(1).A service provider shall not be liable for monetary relief, or except as provided in subsection (i) For injunctive or other equitable relief, for infringement for the storage at the direction of a user of material that resides on a system or network controlled or operated by or for the service provider, if the service provider

A) Does not have actual knowledge that the material or activity is infringing,

B) Service provider responds expeditiously to remove or disable access to the materials claimed to be infringing upon notification to claimed infringement.

Table 4. Extracted text elements from DMCA. SEC103. (a)

Element	Extracted item	Language pattern
Subject	Person	Basic Activity
Action	Distribute, Import for distribution	Basic Activity
Object	False copyright management information	Basic Activity
Target	Copyright-holder	Using concept of subject and object
Scope on action	Knowingly and with the intent to induce, enable, facilitate, or conceal infringement	Noun Phrase Pattern
Modality Notation	Shall not	Modality Pattern

Table 5. Extracted elements from DCMA. SEC.202. (1)

Elements	Extracted item	Language Pattern
Subject	Legal authority	Using concept of subject and object
Action	Make liable (implicate)	Basic Activity
Scope on action	For monetary relief, or except as provided in subsection for injunctive or other equitable	Noun Phrase Pattern
Target	Service provider	Basic Activity
Object	Storage of materials	Basic Activity
Scope on object	Materials: that resides on a system	Noun Phrase Pattern
Scope on storage	Storage is made at the direction of user	Noun Phrase Pattern
Scope on system or network	Controlled or operated by or for the service provider	Noun Phrase Pattern
Mandatory notation	Shall not	Modality Pattern
Condition	1-Without knowledge of infringing 2-Remove or disable access to materials	Conditional Pattern
Scope on condition2	Upon obtaining such knowledge or awareness	Noun Phrase Pattern
Scope on condition3	In the instance of a notification of claimed infringement as described in paragraph	Noun Phrase Pattern

Table 4 and 5 show the items that are extracted from DMCA. SEC103.(a) and SEC202.(1) using mentioned Language patterns in last column. We have used these extracted elements together with Hohfeld concepts of rights to identify Right and Duty dependencies and their correlatives between subject and target as stakeholders of law, as they are mentioned in Table 6.

Type of dependency is identified based on Modality Notation if it indicates duty or right. Using categorization of right (Table 1) extracted right in Sec103 is a type of claim-right. Therefore the copyright-holder has the right to claim if he believes false copyright management information of him had been distributed. This claim makes a requirement in Dropbox system, for infringement notifications and claims and also to have access to its users shared files to check the validity of the claim. This requirement and a list of other requirements extracted from duties and rights are satisfied through concept of goal and are mentioned in Table 7. Each of mentioned goals should be satisfied by the dependency between a dependee and depender as mentioned in Table 7.

Table 6. Duty & Right dependencies extracted from DCMA. SEC103 & SEC202

DCMA Section	Law's actor	Dropbox actor	Target party	Duty or Right Dependency
Sec. 103	Person	User (Dependee)	Copyright-holder Dropbox system (Depender)	DD1: duty not to distribute or import for distribution false copyright management information
Sec. 103	Copyright-holder	Copyright-holder (Dependee)	Dropbox system (Depender)	RD1: right that false copyright management information of him/her don't be distributed or imported for distribution
Sec. 202	Legal Authority	Legal Authority	Dropbox system (Depender)	DD2: duty not to implicate Dropbox for monetary relief, or except as provided in subsection for injunctive or other equitable relief on storage of materials in mentioned conditions
Sec 202	System provider	Dropbox system (Dependee)	Legal Authority (Depender)	RD2: the right not to be liable for monetary relief, or except as provided in subsection for injunctive or other equitable relief on storage of materials in mentioned conditions

To analyze mentioned clause in Sec202, the first step is again to extract the exact duty from the text using extracted elements in Table 5and correlate it with the right of law's target stakeholder based on Hohfeld taxonomy (Table 6). For example Dropbox system has the right not to be liable for monetary relief of any copyright infringement if it did not have knowledge of infringement or has deleted false materials in case of knowledge. (Table 6. RD2). Since we have a strict condition in this text, duty dependency2 (Table 6. DD2) contains a mandatory goal2, which enforces service provider to remove or disable access in case on knowledge on its infringement of copyright. Also, this goal entails other goals since in order to be able to remove or disable access the service provider should have access to saved materials and also the authority to cancel an account. Also, from the extracted scopes on each of the main elements of the text (Table 5) such as scope on object and storage, we were able to extract other goals such as save the materials, control or operate a service or a goal to notify the service provider. RD2 from Table 6 also entails some other requirements. For example, in order to have knowledge of incident, service provider need to have a notification facility which this requirement is extracted from previous article. Service provider also needs to keep notifications for a period of time to prove his authority to remove materials or cancel user's account. Service provider is able to remove materials or cancel user's account. In order to satisfy this requirement, service provider needs to have access to user's account, which is another requirement. This requirement is not clearly mentioned in the law text but is extracted from molecular analysis of the duty of supervisor.

Since all mentioned requirements are extracted from legal resource, therefore there is a mandatory of their existence in the designed system regardless if they are extracted from rights or duties. In case of right, the requirement and its satisfactory goal, plan and resource should be available in system, but the related actor may

decide to claim and use her/his right or not. Therefore, we are expressing the mandatory of extracted requirements from legal text with the aid of a mandatory satisfy link which can ends to a goal, plan or resource and is followed till the termination of system design.

Table 7. Extracted requirements from Duty & Right dependencies

Duty & Right Dependencies	Depender	Dependee	Goals
RD1	Copyright-holder	Legal	• Notify about infringement
DD2	Legal Authority	Dropbox System	• Check copyright-holder notification • Check system log histories
RD2	Dropbox System	Dropbox Engineer	• Disable user access • Remove materials • Access to materials • Disable account
RD2	Dropbox System	Net Engineer	• Control a service or network • Save system log histories for period of time • Keep copyright-holder notifications for a period of time

There is an uncertainty though if actors are trusted to fulfill their duties. If they are not trusted then there should be some form of control on them in order the developer to feel confident that the duties will be fulfilled. In our case study actors have a number of goals that are required to be fulfilled as part of the fulfillment of their duties. Therefore, the developer needs to resolve the dependencies on the actors with duties. The resolutions of the duty dependency on the Dropbox Engineer are shown in table 8.

Table 8. Resolutions

Duty Dependencies	Resolutions
Dropbox Engineer disables user access	1. Normative Trust 2. Legal Team controls the Dropbox Engineer
Dropbox Engineer removes materials	3. Normative Trust 4. Legal Team controls the Dropbox Engineer
Dropbox Engineer accesses materials	5. Normative Trust 6. Legal Team controls the Dropbox Engineer
Dropbox Engineer disables account	7. Normative Trust 8. Legal Team controls the Dropbox Engineer

The resolutions of duties that are based on control though create new duty dependencies on the Legal Team to control the Dropbox Engineer to fulfil various duties. These new duty dependencies need to be resolved again in order the developer to feel confident that the Legal Team will fulfil its duties. The new resolutions are shown in table 9.

Table 9. New resolutions

Duty Dependencies	Resolutions
Legal Team controls Dropbox engineer to disable user access	9. Normative Trust
Legal Team controls Dropbox engineer to remove material	10. Normative Trust
Legal Team controls Dropbox engineer to access material	11. Normative Trust
Legal Team controls Dropbox engineer to disable user account	12. Normative Trust

The identified resolutions create entailments, which are conditions of trust that are required to be valid in order the analysis that has been carried so far to be based on correct trust assumptions. The required entailments and the resolutions from which they originate are shown in table 10.

Table 10. Entailments

Resolution	Entailments
1	System norm is trusted for Dropbox engineer to disable user access
2	Legal Team is trusted to control Dropbox Engineer to disable user access
3	System norm is trusted for Dropbox Engineer to remove material
4	Legal Team is trusted to control Dropbox Engineer to remove material
5	System Norm is trusted for Dropbox Engineer to access material
6	Legal Team is trusted to control Dropbox Engineer to access material
7	System Norm is trusted for Dropbox Engineer to disable user account
8	Legal Team is trusted to control Dropbox engineer to disable user account
9, 10, 11, 12,	System norms are trusted for Legal Team to control Dropbox Engineer for the respective goals

The above entailments were validated with evidence that was collected from the case study. Dropbox is a small size company where all employees are located in the same office. Also, the Legal Team and the Dropbox Engineer have close collaboration and this enables the Legal Team to control the Dropbox Engineer. Therefore, since the entailments are valid then there is confidence in the fulfilment of duty dependences. Thus, the analysis of legal issues was based on trust assumptions that are valid and as a result the Dropbox System can be trusted to be law compliant.

5 Related Work

Number of researches have analysed laws and regulations in order to extract right and obligations of law's stakeholders. Breaux et al. [12] has used natural language patterns in order to elicit rights and obligations from laws and regulation. He has used HIPAA law (Health Insurance Portability and Accountability Act) to extract security requirements of health systems. Later he extends the work by to analyse access control rules to it. Islam in [5] also has used natural language patterns with Hohfeld legal taxonomy to extract security requirements from laws and combine them with the ISO/IEC policies and has integrated extracted constraints into UMLsec for secure detail design of a system. May at [13] have extracted privacy requirements from legal

text using access control techniques. Dorimont et al. [14] have modelled regulations using GORE software modelling methodology (Goal Oriented Requirement Engineering). The special GORE approach that he has used is KAOS which starts modelling from goal and refine goals in an incremental process through leading to relevant tasks and involved actors. Siena et al. at [15] has only focused on Hohfeld legal taxonomy to extract security and privacy requirements from laws and regulations. Mead et al. [16] has introduced a method called SQUARE (Security Quality Requirement), which elicits and documents security requirements. Mellado et al. [17] has presented a security requirement engineering process based on Common Criteria and ISO/IEC 270001, which can be used as a constant method to develop system based on these two policy references.

Among mentioned works, most have tried to elicit security requirements from laws. Some have only concentrated on special laws such as HIPAA and a case study based on that law. To demonstrate a framework, we need to analyse more number of most applicable laws in order to have a more valid framework that works in different cases. Some of mentioned works only elicit security requirements without considering laws. The advantage of this work is to align legal requirements with other requirements of a system, first to answer to enforcement of compliance with extending current state of the art of software development methodologies with legal concepts; second to elicit more requirements of system considering laws and molecular analysis of rights and duties. Among the above mentioned works some also have introduced legal concepts during development of software system, but the advantage of this work compared to theirs is using of an actor-goal oriented software methodology which has the capacity to map legal relationship of stakeholders to its own language using concept of dependencies between actors of system. The reason of this usage is that we believe an only goal-oriented or actor-oriented methodology lack the contiguous concepts of laws and software development and consequently makes the process difficult.

There are a number of approaches that consider trust issues, including trust modelling and the formation of a common vocabulary during the software development stage. In [18] the proposed method makes use of the Goal Requirement Language (GRL) and Use Case Map (UCM) which both of them belong to the User Requirement Notation (URN). Specifically, trust is captured as a soft goal because of the uncertainty of whether it has been satisfied or not and because of its fuzzy nature. Further analysis of trust as a soft goal eventually leads to well-defined tasks. Yu and Liu [19] address the issues of trust at the requirements level of the system development process. They consider trust as a non-functional requirement, where trust is a combination of all or some quality attributes of a system under development and they demonstrate their approach by describing the behaviour of a system in the case of attack and examine defences that are needed from trust perspective. Secure Tropos [20] extends Tropos methodology with the concepts of trust, delegation, provisioning and ownership in order to allow the developer to capture trust relationships at a social and individual level. Bimrah [21] extends the Secure Tropos [22] methodology with the concepts of request, action, trust relationship, trusting intention, reputative knowledge, recommendation and consequence in order to model trust. The developer is guided through a series of models in order to analyse and reason about trust relationships.

However, the above-mentioned approaches, and in particular, [18] treat the system as a black box without looking into the trust relationships inside the system, thus concentrating on the trust relationships between user and the system. On the other hand, in the cases such as [19], [20], and [21] where trust relationships are modelled they are not justified or they are limited to the direct trust relationships omitting the indirect ones that can become a serious vulnerability to the trustworthiness of the system. We believe our work contributes in this direction by providing a meta-model that supports the capture and reasoning of the direct and indirect trust relationships inside the socio-technical system and identifying the gaps in the chain of trust relationships. With the use of the proposed meta-model, there is the advantage that these trust relationships become explicit and the developer can reason about them in order to develop confidence in them. Otherwise, if any trust relationship is left unidentified it could become a potential vulnerability to the functionality and proper operation of the final system. The main contribution though of the meta-model presented in this paper is that not only allows the elicitation of requirements from laws and regulations but more importantly it allows the trust analysis of the actors that are related with the law compliance of the system.

6 Conclusion

In this paper we have presented a meta-model that combines legal and trust related concepts. It enables the developer to model the legal issues that introduce legal requirements to the system design and constitutes the developed system law compliant. Also, the meta-model enables the assessment of trustworthiness of the actors that are assigned and responsible to fulfil legal duties. The incorporation and analysis of the legal issues is explicitly carried out in order to show its importance and the trustworthiness of the actors involved in the fulfilment of the legal obligation is assessed in a structured and coherent way.

In addition, the applicability and benefits of the meta-model were demonstrated by using a scenario from the popular Dropbox service. Legal requirements were identified and the trustworthiness of the actors responsible for legal duties was assessed in a systematic way in order to ensure that the system is trustworthy in terms of law compliance.

Acknowledgments. The second author would like to acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC) and British Telecom (BT).

References

1. Ryan, J.D.: Two Views on Security Software Liability: Let the legal System Decide. In: Mead, R.N., McGraw, G. (eds.) IEEE Security & Privacy, pp. 70–72. IEEE Computer Society Press (2003)
2. Zarrabi, F., Mouratidis, H., Islam, S.: Extracting Security Requirements from Relevant Laws and Regulations. In: Proceedings of the International Conference on Research Challenges in Information Science (2012)

3. Pavlidis, M., Mouratidis, H., Islam, K.P.: Dealing with Trust and Control: A Meta-Model for Trustworthy Information Systems Development. In: Proceedings of the International Conference on Research Challenges in Information Science (2012)
4. Hohfeld, W.N.: Fundamental Legal Conceptions as Applied in Judicial Reasoning. Yale Law Journal 23(1) (1913)
5. Islam, S., Mouratidis, H., Jürjens, J.: A Framework to Support Alignment of Secure Software Engineering with Legal Regulations. Journal of Software and Systems Modeling (SoSyM), Theme Section on Non-Functional System Properties in Domain-Specific Modeling Languages (NFPinDSML) 10(3), 369–394 (2011)
6. The Cambridge Encyclopaedia of Language. Cambridge University Press (1997) ISBN 0-521-55967-7
7. Yu, E.: Towards Modelling and Reasoning Support for Early-Requirements Engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, pp. 226–235 (1997)
8. Yu, E., Liu, L., Mylopoulos, J.: A Social Ontology for Integrating Security and Software Engineering. In: Mouratidis, H., Giorgini, P. (eds.) Integrating Security and Software Engineering: Advances and Future Visions, pp. 70–105. Idea Group Publishing, London (2007)
9. Mollering, G.: The Trust/Control Duality: An Integrative Perspective on Positive Expectations of Others. International Sociology 20(3), 283–305 (2005)
10. Dropbox, `http://www.dropbox.com`
11. House of Representatives: Conference Report: Digital Millennium Copyright Act. Report 105-796 (October 1998)
12. Breaux, T.D., Antón, A.I.: Analyzing Regulator Rules for privacy and Security Requirements. IEEE Transactions on Software Engineering 34(1) (January-February 2008)
13. May, M.J., Gunter, C.A., Lee, I.: Privacy APIs: Access Control Techniques to Analyze and Verify Legal Privacy Policies. In: Proc. of the 19th Computer Security Foundations Workshop (July 2006)
14. Darimont, R., Lemoine, M.: Goal-oriented Analysis of Regulations. Regulations Modeling and their Validation and Verification (2006)
15. Siena, A., Mylopoulos, J., Perini, A., Susi, A.: From Laws to Requirements. In: 1st International Workshop on Requirements Engineering and Law (2008)
16. Mead, N.R.: Identifying Security Requirements Using the Security Quality Requirements Engineering (SQUARE) Method. In: Integrating Security and Software Engineering, pp. 44–69. Idea Publishing Group (2006)
17. Mellado, D., Medina, E., Piattini, M.: A common criterion based security requirements engineering process for the development of secure information system. Computer Standards & Interfaces 29, 244–253 (2007)
18. Pourshahid, A., Tran, T.: Modelling Trust in E-Commerce: An Approach Based on User Requirement. In: Proceedings of the 9th International Conference on Electronic Commerce, USA, pp. 413–422 (2007)
19. Yu, E., Liu, L.: Modelling Trust for System Design Using the *i** Strategic Actors Framework. In: Falcone, R., Singh, M., Tan, Y.-H. (eds.) AA-WS 2000. LNCS (LNAI), vol. 2246, pp. 175–194. Springer, Heidelberg (2001)
20. Giorgini, P., Massaci, F., Mylopoulos, J., Zanone, N.: Requirements Engineering for Trust Management. International Journal of Information Security 5(4), 257–274 (2004)
21. Bimrah, K.K.: A Framework for Modelling Trust during Information Systems Development. PhD Thesis, University of East London (2009)
22. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-Oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering 17(2), 285–309 (2007)