From e-business to e-laws and e-judgments: 4,000 years of experience

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

Abstract— **Rapid e-transactions are possible today in many areas of application, which creates a need for rapid resolution of conflictual situations potentially deriving from the performance of these transactions. This will lead to the development of e-laws, e-regulations, e-judgments, and e-enforcement, to be quickly and automatically executed when conflictual situations occur. Examples of possible application of these ideas are found in cloud computing, privacy, security, e-business.It is shown that some principles for the implementation of these ideas can be found in the history of law, starting from very ancient legal systems that looked like sets of logic axioms or computer programs, reflecting the will of the legislator to tightly control the judicial authorities. The role of ontologies for creating complex legal systems, useful to formalize e-laws, is discussed. Principles of consistency and completeness of legal systems are briefly presented. The relevance of Artificial Intelligence methods for e-judgments is briefly evaluated. The principles presented in this work have potential for application in future automated cyberlaws contexts.**

Keywords – cyberlaws; electronic commerce; electronic laws; electronic judgments; electronic courts; electronic enforcement; legal ontologies; completeness of law; consistency of law

I. INTRODUCTION

Rapid e-transactions are possible today in many area of e-business, but there are no mechanisms to quickly address conflictual situations that may derive from them. We conjecture that the need for rapid decision of litigation in contexts of e-transactions will lead to e-laws and e-regulations, to be used by automatic e-courts, leading to e-judgments and e-enforcement, and we present principles that can be used for the development of these concepts.

In Section II we present several examples of situations where these concepts could be useful, in the areas of cloud computing, privacy, security, e-business. The rest of the paper deals with concepts that can be used for the formulation of precise e-laws, e-regulations and e-judgments. In Section III, we leap back thousands of years to show that some structuring ideas that could be used for the formulations of e-laws were known in ancient civilizations. In Section IV we show how ontologies can be used to precisely structure legal systems. In Section V we deal with the problem of completeness and consistency of legal systems. Section VI briefly addresses the use of artificial intelligence methods to arrive at e-judgments. Section VII discusses enforcement and e-penalties.

II. MOTIVATING EXAMPLES

Following are some examples showing the practical usefulness of the concept of e-judgment in the e-business and cyberlaw context. Several other examples can be generated with some imagination. A consequence of this is the usefulness of the concepts of e-laws or e-regulations, which in this paper will be taken to be laws or regulations that can be automatically inferred from, leading to e-judgments. E-courts will be automated mechanisms capable of performing these inferences.

Example 1: Service-Level Agreements (SLAs) for telecom or cloud computing. Subject A leases a line or contracts a cloud computing agreement with operator B. They agree that entity C will arbitrate any disagreements, and they deposit with C an electronic, formalized SLA, specifying certain elements of *QoS* (Quality of Service), as well as penalties for non-compliance. Later A has reason to believe that the promised QoS is not being delivered, and advises B, who disagrees. A then contacts C, who performs some tests or consults existing logs and agrees with A, therefore it sends B an order to pay A a penalty. This is completed within seconds (concepts needed to understand this example are elaborated in [13]).

Example 2: Privacy protection. Suppose that a web query tries to access an external database, but the database access control system denies access on grounds of privacy protection. The requesting agent may have been programmed to appeal this decision by automatically sending a query to an electronic system set up by a body such as a Privacy Commissioner. The latter, after considering the privacy status of the requesting agent and of the data being requested, may prescribe that access should be provided. This e-judgment would be sent to the database access control system, which would allow access.

Example 3: *Security.* This is an area were many types of violations can occur, some of which can be reliably logged. Some of these can be covered by laws or regulations for which the premises can be objectively checked. If an independent, certified log exists that A's machine has tried to snoop in B's, B's machine can automatically request that A by fined, or requested to pay damages by an e-court. Similar examples can be found in the areas of *privacy* and *copyrights*.

 ISBN: 978-1-61208-122-9

Example 4: Electronic bidding. A government provides regulations for electronic bidding processes. Bidders deal with individual departments, but a central e-authority has been set up for appeals of contractors against decisions of the departments, regarding compliance with governmental regulations. Departments whose software is not up to date with the current regulations may see their decisions automatically reversed. Suppose that recently the central authority has simplified bidding procedures, but this has not yet been implemented locally.

Example 5: Tax law. Local tax laws may be in contradiction with principles of state or federal law. Or some businesses could charge taxes according to erroneous criteria. Since in many situations taxes are calculated by computer, can these calculations be corrected rapidly by intervention of an e-authority?

Example 6: E-commerce. An online buyer receives goods that do not have the advertised characteristics, or receives them later than promised. Can a quick decision on fair compensation be reached with the help of an e-authority?

Such scenarios are not realistic today because they depend on much relevant information being electronically available, e.g., for Example 1 a precise agreement is needed, together with methods to check whether the terms of the agreement are satisfied, as much as possible independent of human intervention. However, setting up such systems seems to be feasible in many cases.

Once these judicial or quasi-judicial processes are put in place, one can see that in time more areas of application will open up, towards judicial areas that have been traditionally occupied by human courts, especially in situations where decisions can be taken in terms of elementary facts and basic reasoning. The area of commercial law seems to be a prime candidate. In some cases the fact-finding may have to remain in human hands, but still the legal consequences can be automatically derived.

III. HISTORICAL PRECEDENTS

We will show in this section that some structuring concepts that are important for the design of electronic legal system have been known for a very long time, some in fact from the historically recorded beginnings of legislation. The following examples are only a few out of many that could be cited.

A. Ancient examples of precisely formulated laws

The first codes that we know are the Sumerian and Babylonian codes of 4000 years ago. These codes were written in a precise, concise and factual style that is familiar in IT today. Here is an article from the Ur-Nammu code, said to be the earliest law code known [18]:

"If a man had let an arable field to a(nother) man for cultivation, but he did not cultivate it, turning it into wasteland, he shall measure out three kur of barley per iku of field."

We find here the Event-Condition-Action (ECA) style that is familiar in event-driven architectures and active database system [3]. Further, the event consists of three parts: subject, verb, object according to the structure familiar in access control systems, e.g., in the XACML language [12], namely:

Subject: *a man*
Verb: *had let*
Object: *an arable field*
Condition: *but he did not cultivate it, turning it into wasteland*
Action: *he shall measure out three kur of barley per iku of field*

Here the action is a penalty, with a precise method to measure it. In other cases in this code the action is a legal effect, such as the loss of property. There are also articles that do not quite fit this pattern, but will fit other patterns that can be easily formalized. The famous code of Hammurabi of about 300 years later [11] follows the same style, is much more extensive, and is worth reading (although not for people averse to cruel and extreme punishments…).

The Chinese Tang code of year 653 A.D. [4] is another example of a code which is remarkable for its clear style and the intricate decisional procedures it describes. Essentially it is ECA, with few legal concepts. But in terms of Computer Science, one can recognize well-known concepts such as method invocation with parameters, loops with arithmetic, if statements, case statements etc., in the action part, for the calculation of penalties.

Here are two articles from this code:

Ex. 1: "In cases in which someone at first hit a person …, and then snatched his goods, calculate the value of the stolen goods to apply the law on robbery by force. When death resulted, the sentence is exile with labour. When he took the goods by stealth, use the law on robbery by stealth, but increase the penalties one degree. When killing or injuring resulted, apply the laws on intentional battery."

Ex. 2: "Those who plant public or private land they do not have rights to are liable to a beating of thirty strokes for the first mu or less, increasing one degree for each five mu. After the penalty reaches one hundred strokes, it increases a degree for every ten mu. The maximum penalty is one and a half years penal servitude. The penalty is reduced one degree if the land had been uncultivated. If force was used, the penalty is increased one degree. The crops belong to the government or the owner. "

These laws show that some legislators in the past have tried to control tightly the work of courts, so that the decisions were almost automatically determined by logical inference once the facts had been established.

In these starkly simple laws we can see the convergence of two conceptual worlds: the real world where situations can take many different aspects, sometimes difficult to classify precisely; and the logical world where a definite, verifiable decision has to be

 ISBN: 978-1-61208-122-9

reached by logical deduction. The interface between the two worlds is impersonated by the judge, who has to map the complexity of the reality into a template leading to the decision

B. *Precisely regulated legal process: the Roman formula process*

The Roman civil law formula procedure [14] is another example of tightly controlled legal procedure of the past, in some cases reducing the final phase of the process to simple fact finding, followed by a logical deduction. Essentially, for each type of litigation there were pre-set formulae consisting of several parts where the main elements of the litigation were expressed in precise, stylized language. In the first phase of this process, the plaintiff approached a magistrate and the magistrate convened the defendant. The three consulted and the magistrate produced, with the agreement of all, a formula and the name of a judge. The judge was essentially an arbitrator, who was responsible for the second phase, where he carried out the instructions of the formula, resulting in a legally binding decision.

The components of the simplest formulae were the *Demonstration*, the *Intention*, the *Adjudication*, the *Condemnation*. The following description of the four basic elements is partly paraphrased from [14].

The principal function of the Demonstration was to indicate the subject matter of dispute (the cause of the action, the title of the plaintiff's right, the origin of his claim), as in the following example: "*Whereas A sold a slave to B"* or, "*Whereas A and B have asked to be assigned a judge for the partition of a farm"*. The Demonstration expressed prerequisites that were uncontested between the parties.

In the Intention, the claim of the plaintiff was expressed in conditional form, thus: *"If it is proved that A ought to convey the sum of … to B*" or: "*If it is proved that the slave in question belongs to A"* or yet: *"If it is proved that A has given silver to B, and A has kept it in bad faith"*.

The Adjudication empowered the judge to transfer the ownership of a thing to one of the litigants, and occurred most commonly in the actions for partitioning an inheritance, for dividing common property between co-partners, and for determining boundaries between neighbouring landholders, e.g.: *"Let the portion of the property that ought to be transferred to A be transferred to him."*

The Condemnation empowered the judge to condemn or absolve the defendant, thus: *"If it proved, condemn A to pay B the sum of … ; if it is not proved, let him be absolved"*.

These components could be varied in several ways, and other components were possible: this type of process was in use for hundreds of years and had to be adapted to many situations. In particular, there were elements by which each party could state other facts and respective rebuttals (*Exceptions*), all to be checked by the judge. This created a nested structure in the formula.

The formula was essentially an instantiation of the law for a specific case. It reduced what could be complex law into a format whose core was essentially ECA, Event-Condition-Action: the Event is specified in the Intention, the Condition in the Demonstration and in the Intention, and the Action in the Adjudication or in the Condemnation. In simple cases, the formula could be set up in such a way that the judge did not need to know the law, and had simply to check facts, i.e., whether the condition in the Intention was true or false (which he could do by using witnesses, inspection, etc.) The Adjudication or Condemnation followed by a simple syllogism [14], i.e., an elementary deduction in predicate calculus.

Reference [7] cites a view by which this procedure was "one whose rapidity, brevity and effectiveness has, perhaps, never been equaled" and it goes on by saying that this view is an understatement.

Today, stylized and agreed formulae are used in legal documents such as land transfer acts, insurance contracts, etc. but not normally in judicial procedures.

C. *What can be learned from these precedents*

From the Sumerian and Tang codes we can learn that many straightforward laws and regulations can be formulated in ECA style and then easily compiled into software code. A natural choice would be to compile them into a logic-based programming language such as Prolog. Once the facts are determined, decisions are reached automatically. The external interface for a system designed to provide the applicable decisions in real cases could be implemented by clickable boxes. When the facts have been determined and the boxes clicked accordingly by the judge, the sentence is automatic.

But of course modern legal systems are much more complex. The formula system of the Romans provided a step through which everyday legal decision-making was simplified: from it we can learn that, even in complex legal systems, the decision criteria for many legal cases can be expressed in ECA format after instantiation. In Roman times, it was the magistrate who instantiated the law in ECA form. In order to use this method in modern e-business systems, we could perform an analysis and classification of common legal complaints in this environment and then, following the law, set up appropriate formula templates for each of them in a web server. The plaintiff would scan the available formulae to find one that matches her complaint, and would fill it with her parameters. In common text, a formula may run roughly as follows: *"Whereas A has purchased cloud services from B, specifying a minimum QoS and penalties if the QoS is not delivered: A claims that the promised QoS is not being delivered (details: it is too slow, etc.) If A's claim is proved, B must pay A the sum of $X."* An e-entity can be appointed to perform arbitration; the entity will perform tests or consult logs, and may be able to reach a decision within seconds. Alternatively, a remote human arbitrator can be appointed who would examine evidence and fill in templates that would lead to quick automatic decisions according to the pattern pre-set by the formula.

 ISBN: 978-1-61208-122-9

Such procedures could consist of several steps. For example, if it is impossible to reach a verdict on the base of the available information, there could be formulae to request the parties to make available additional information.

The very existence of such efficient mechanisms may lead to quick agreements between the parties, without even having to use them. Although popular e-business providers such as eBay offer complaint procedures, they are not beyond improvement.

IV. THE ROLE OF ONTOLOGIES

A. Ontologies of subjects and objects

From a modern point of view, the very ancient codes we have mentioned have the shortcoming of being applicable only in very specific, punctual situations. Modern codes achieve greater generality by the use of structured concepts, called ontologies.

For example, the Ur-Nammu article of law given above can be generalized by introducing a classification of things that can be let, a classification of types of damages, and a classification of possible penalties. Such classifications can be represented in precise form by the use of ontologies.

Trivially, in a situation where there are two things that can be let: fields or houses, and two possible damages, burning or flooding, a norm of the type: *"If a man had let something to another man, but he damaged it, he shall pay the value of the thing to the other man"* can be instantiated in four possible ways:

"If a man had let a field to another man, but he burned it …"

"If a man had let a field to another man, but he flooded it …"

"If a man had let a house to another man but he burned it …"

Etc.

Such ontologies and instantiations are used by lawpeople when they apply the law. They originate from daily life knowledge, specialized knowledge such as engineering, or legal knowledge.

The term ontology has a history in philosophy. It has become a keyword in Computer Science, with a somewhat different meaning, and it is in its second meaning that will be used here. An ontology in this sense is definition of a set of concepts together with their relationships. Various ways of representing ontologies are: sets of logical axioms involving constants, data types, or diagrams (e.g., UML diagrams). Many different ontologies can be present, explicitly or implicitly, in a legal system. For example, inheritance law involves (at least) an ontology describing the structure of a family, an ontology describing rights that the deceased may hold, an ontology describing the objects on which rights can be held, and an ontology of the structure of wills.

By expressing relations in ontologies, powerful generalizations can be obtained. Following are some examples.

Judges and lawyers generalize the application of law by using analogical thinking. But this is based on implicit similarity relationships and assumptions (i.e., ontologies) such as: a norm that applies to *x* also applies to *y* if *x* is similar to *y*.

The Islamic legal system is one of many legal systems where analogical thinking has a very important role. In the Koran, the use of wine is forbidden because of its intoxicating effects. Islamic tradition then forbids the use of intoxicating drugs. This is an application of the argument *a fortiori* (for stronger reasons). This reasoning can be modeled in logic with the help of an ontology, which in this case is a partial order between intoxicating media, including the fact: *wine* < *drugs*. Then we need an axiom, e.g:

$x < y \rightarrow (Forbidden(x) \rightarrow Forbidden(y))$

If we wish to model the fact that performing a more serious offence involves a more serious penalty, then we need to add an ontology for penalties, with a partial order among penalties, and a corresponding axiom. For example, in an enterprise there may be an ontology of degrees of confidentiality of the type *UnClassified* < *Classified* < *Secret* < *TopSecret*. There may also be a hierarchy of degrees of protection. Then it is possible to introduce axioms stating that for higher degrees of confidentiality, there must be higher degrees of protection, or of stiffer penalties in case of breaches.

Many types of legal reasoning can be implemented precisely by defining appropriate ontologies. So an e-law should contain not only rules (such as ECA-style rules), but also the appropriate ontologies and axioms needed to define the full extent of the rules.

B. Ontologies of legal concepts

Over the millennia, the men and women of law have developed sophisticated ontologies of legal concepts. For example the Roman 'Law of the XII tables' (fifth Century BC) said in Table III [17]:

"A person who admits to owing money or has been adjudged to owe money must be given 30 days to pay".

So here we have the right of the creditor to the money in a specific time span. And:

"After then, the creditor can lay hands on him and haul him to court".

So here is the power of the creditor to take the debtor to court.

Much of modern western legal theory is constructed in terms of concepts such as these.

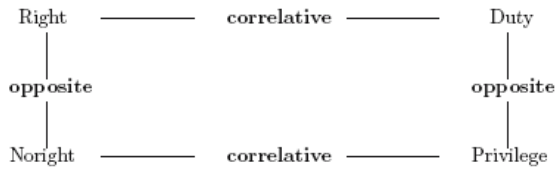
The American jurist Wesley Newcomb Hohfeld (1879-1918) developed a well-known ontology of these concepts, as follows:

Jural opposites: 1. Right/No-Right 2. Privilege/Duty 3. Power/Disability 4. Immunity/Liability

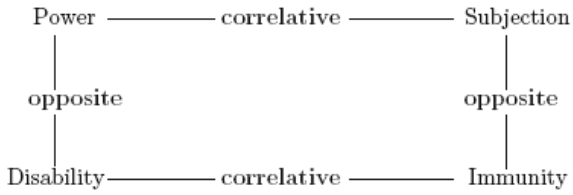
Jural correlatives: 1. Right/Duty 2. Privilege/No-Right 3. Power/Liability 4. Immunity/Disability.

Reference [16] proposed a representation of this ontology in terms of two conceptual squares: the obligative square and the potestative square. The obligative square is as follows:

 ISBN: 978-1-61208-122-9



While the potestative square is:



The connection between obligative and potestative rights is provided by the fact that one subject's *x* obligative right that another subject *y* does an action is protected through *x*'s potestative right to activate the corresponding sanction against *y*.

These two squares are already implicit in Hohfeld's work. References [15] Ch. 19 and 22 and [16], complete this work by providing formal definitions of the concepts in terms of deontic concepts of obligation and permission.

Many important legal concepts are based on the concepts just mentioned. Hence, the precise formal expression of Hohfeld's ontology continues to be the subject of much interesting research.

Reference [6] presents various ontological networks of legal concepts, not related to Hohfeld's and mostly related to criminal law.

Many legal concepts are fairly precisely specified, but their complete formalization is elusive. It is elusive because they have to remain adaptable to the many situations of real life. And it is elusive because they involve reference to many concepts that one can try to formalize by using complex ontologies, higher order logics, modal logics, etc. Even if complete formalization could be achieved, automatic derivation of consequences from such complex logical systems would be daunting, because of intrinsic computational complexity issues.

V. CHECKING LEGAL SYSTEMS FOR CONSISTENCY AND COMPLETENESS

The matters of legal systems *consistency* and *completeness* were addressed in ref. [1], with citations to earlier work in Philosophy of Law. The remarks of these authors are still valid today. A legal system that is incomplete may not be able to infer a decision for legally relevant situations; a system that is inconsistent may be able to infer more than one decision. The author has presented some considerations on this topic in [7] and we should not repeat what has already been published. We will present here some concepts in order to complete the outlook of this paper. When real-life legal systems are (or appear to be) inconsistent or incomplete, this is taken care of by the (human) judges who use their discretion and knowledge (both of the law and of real life) to interpret the law. Equity and analogy are often used: what is a fair decision in this case? What was the intent of the legislator? Are there similar situations for which there is a known solution? We have seen that such thinking can be represented to some extent by using ontologies, however for complex reasoning we straddle in the area of Artificial Intelligence methods, see below.

From logic we know that first-order theories that include a significant portion of the theory of natural numbers cannot be both consistent and complete. However in practice some consistency and completeness checks can be performed by assuming a small, finite number of elements in the theory. In this case these problems reduce to the problems of consistency and completeness in propositional calculus, and can be addressed by the use of satisfaction algorithms. Although the best known satisfaction algorithms are of exponential complexity, in practice they lead to solutions in reasonable time in most cases [10]. In other words, partial completeness and complexity checks are often feasible.

Consistency. An interesting discussion of the use of 'preferences' to resolve inconsistencies in law and ethics is presented in [15] Ch. 7. This solution is similar to resolution methods already known in computing: sets of rules where inconsistencies can occur are ranked in order of priority and only the highest ranking rule is used in case of conflict.

Completeness. What does it mean for a set of rules to be complete? If complete and finite ontologies exist, it may be possible to check whether all theoretically possible situations have been considered, that a rule exists for each of them (for example for each of the four cases mentioned in Section 4.1). However since many practical situations are legally irrelevant this may lead to many unnecessary questions. The realm of possibilities can often be limited by considering the intent of the law. Suppose that the intent of the law is that no explosive packages should be sent over the mail and suppose that preventing this for different types of explosives should lead to different penalties. Since a single blanket rule is not possible, there will have to be a number of rules, for different types of explosives. Is there a rule for each possible type of explosive? This can be checked if an enumeration (i.e., an ontology) of such type exist. The knowledge that the intent of the law is limited to explosive packages makes it unnecessary to consider what should be the law for other types of packages.

VI. THE ROLE OF AI METHODS, AND HOW FAR SHOULD WE GO?

There is a very considerable research area whose aim is to study the use Artificial Intelligence methods in order to create models of legal thinking (for a brief overview and bibliography, see [2]). This work is very interesting, however often these AI methods do not lead to

 ISBN: 978-1-61208-122-9

incontestable results, since they use heuristics that yield 'acceptable' solutions of which there can be several.

If AI decision heuristics were used to help deciding legal cases, one can think that different parties could bring different methods to the table, each arriving at a conclusion coherent with the submitting party's position! This seems to be hardly worth the trouble, since it would complicate the work of the human judge who would still have to decide between the two positions, using human intelligence.

Using such heuristics would be problematic in the case of e-judgment where a single predictable decision must be reached. It appears that in this case it is necessary to use strictly deductive methods based on established facts and precisely, consistently formulated law. On this basis, any judge or any computer should arrive at the same conclusion. The existence of such laws and the possibility of such univocal deductive decisions seem to delimit the area in which the approach we are discussing is feasible.

VII. E-ENFORCEMENT AND E-PENALTIES

How to give teeth to e-laws? This does not seem difficult. One can easily see two types of penalties: monetary (fines, reparation) and exclusion, one leading to the other. So an e-party could be asked to pay a sum of money, to the plaintiff or to the platform provider, and if it does not pay, it could lose its platform or its certificate. Penalties could apply also in cases where a party refuses to collaborate in the e-judicial process, e.g., it does not reply within a specified delay. In many cases this will need no human intervention. Appeals to a human court should be allowed always.

VIII. CONCLUSIONS AND DISCUSSION

We have presented the desirability of developing automated systems of e-laws, e-regulations and e-judgments, in e-business and cyberlaw contexts, based on formally specified laws and logic deduction. Applications were found in several areas. In the initial stages, such systems will not be real legal systems; they will be used mainly in order to attempt quick resolution of complaints. However a time when a legal value will be given to them may not be distant: note that automatically produced tax assessments already have such value. A practical problem concerns how to make available the necessary evidence (e.g. system logs) in normalized electronic forms, however we can expect that in many cases this will be done eventually. Logs are required for other purposes, such as auditing.

The proponents of the use of formal logical deduction in the legal process have pointed out that such use helps towards predictability in the process, which is required for assuring the principle of certainty of law, proposed by Max Weber, among others, as necessary condition for the achievement of economic goals. The results of the legal process are more predictable and uniform if the law is logically clear and consistent and the decisions are reached by formal logical inference from the law and the established facts.

Today, a technological argument for the use of formal logic in the legal process is provided by the fact that information systems are increasingly entrusted roles of legal relevance and the most obvious mechanism for computers to draw legal conclusions is logical deduction. Multi-agent systems are very similar to social systems with their policies, which essentially have the function of laws, and are inferred and enforced automatically. However when human subjects and real-life facts are involved, the decision process may have to be more complicated, possibly requiring human participation.

We have seen that some conceptual base for such systems can be found in very ancient legal systems, and that some of the ideas used in ancient times are still valid today. More recent legal systems have tended to give more importance to the factual and human insight of the courts, something that can't be handled by automatic systems.

It is an unfortunate habit in IT to start projects without considering what has been done before. It would be really most unfortunate if such an attitude was followed in this area.

Background discussion and references on topics related to this paper can be found in [9].

ACKNOWLEDGMENT

This research was partially funded by a grant of the Natural Sciences and Engineering Research Council of Canada. I am indebted to my students Sofiene Boulares and Waël Hassan for useful ideas.

REFERENCES

[1] C.E. Alchourròn, E. Bulygin, Normative Systems, Springer, 1971.

[2] T.J.M. Bench-Capon, H. Prakken, "Introducing the logic and Law corner," J. logic and computation, 18(1), 2008, 1-12.

[3] K.R. Dittrich, S. Gatziu, A. Geppert, "The active database management system manifesto: A rulebase of ADBMS features," Lecture Notes in Computer Science 985, Springer 1995, 3-20.

[4] P.B. Ebrey (Ed.), Chinese Civilization: A Source Book, The Free Press, 1993.

[5] W. Hassan, L. Logrippo, "Requirements and compliance in legal systems: a logic approach," In: Requirements Engineering and Law, 2008 (RELAW '08), 40-44.

[6] J.C. Joerden, Logik im Recht, 2te Aufl., Springer, 2010

[7] A. Kokourek, "The formula procedure of roman law," Virginia Law Review, 8 (5) 1922, 337-355.

[8] L. Logrippo, "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 SoMeTConference, IOS Press, 2007, 343-354.

[9] L. Logrippo, "A formal logic perspective on law, jurisprudence, and information technology in a historical context," http://www.site.uottawa.ca/~luigi/papers/LegalLogicBlog.htm (Consulted Dec. 2010).

[10] S. Malik, L. Zhang, "Boolean satisfiability – from theoretical hardness to practical success", Comm. ACM 57 (8), 2009, 76-82.

[11] "Mesopotamia: The Code of Hammurabi," http://www.wsu.edu/~dee/MESO/CODE.HTM (Consulted Aug. 2010).

[12] OASIS eXtensible Access Control Markup Language (XACML) Technical Committee. http://www.oasis-

open.org/committees/tc_home.php?wg_abbrev=xacml (Consulted Aug. 2010).

[13] P. Patel, A. Ranabahu, A. Sheth, “Service level agreement in cloud computing,” Technical report, Ohio Center of Excellence in Knowledge Enabled Computing (Kno.e.sis), http://knoesis.wright.edu/library/resource.php?id=742 (Consulted Oct. 2010).

[14] E. Poste, Elements of Roman Law by Gaius. Clarendon Press, 1875.

[15] G. Sartor, Legal Reasoning: A Cognitive Approach to the Law. Chapter 5 in: A Treatise of Legal Philosophy and General Jurisprudence, Springer 2005.

[16] G. Sartor, “Fundamental legal concepts: a formal and teleological characterization”, Artificial Intelligence and Law 14 (1), 2006, 710-770.

[17] “Twelve tables”, http://www.fact-index.com/t/tw/twelve_tables.html

[18] “The Ur-Nammu law code”, http://realhistoryww.com/world_history/ancient/Misc/Sumer/ur_nammu_law.htm (Consulted Dec. 2010).

 ISBN: 978-1-61208-122-9