Formal Methods as a Link between Software Code and Legal Rules

Daniel Le Métayer

INRIA Grenoble Rhône-Alpes
France
Daniel.Le-Metayer@inria.fr

Abstract. The rapid evolution of the technological landscape and the impact of information technologies on our everyday life raise new challenges which cannot be tackled by a purely technological approach. Generally speaking, legal and technical means should complement each other to reduce risks for citizens and consumers : on one side, laws (or contracts) can provide assurances which are out of reach of technical means (or cope with situations where technical means would be defeated); on the other side, technology can help enforce legal and contractual commitments. This synergy should not be taken for granted however, and if legal issues are not considered from the outset, technological decisions made during the design phase may very well hamper or make impossible the enforcement of legal rights. But the consideration of legal constraints in the design phase is a challenge in itself, not least because of the gap between the legal and technical communities and the difficulties to establish a common understanding of the concepts at hand. In this paper, we advocate the use of formal methods to reduce this gap, taking examples in areas such as privacy, liability and compliance.

Keywords: regulation, law, legal, liability, accountability, privacy, compliance, formal model, causality.

1 Motivation

The rapid evolution of the technological landscape and the impact of information and communication technologies (ICT) on our everyday life raise new challenges which cannot be tackled by a purely technological approach [Poullet - 2006]. For example, the protection of privacy rights on the Internet or in pervasive computing environments is by definition multidimensional and requires expertise from disciplines such as social sciences, economics, ethics, law and computer science [Rouvroy - 2008]. Other examples of the ever-growing intermingling of ICT and law include electronic commerce, digital rights management (DRM), software contracts, social networks, forensics, cybercrime, Internet regulation, e-government, and e-justice - and this list is far from limitative. As far as research is concerned however, there are still very few links between the ICT and law communities. This situation is unfortunate considering the importance of the interests at stake (not only in economic terms but also for society as a whole).

G. Barthe, A. Pardo, and G. Schneider (Eds.): SEFM 2011, LNCS 7041, pp. 3–18, 2011.

Starting from this observation, the general goal of the research outlined here is to contribute, in partnership with lawyers, to the development of new approaches and methods for a better integration of technical and legal instruments.

In practice, the interactions between ICT and law take various forms and go in both directions [Le Métayer, Rouvroy - 2008]:

- ICT "objects" are, as any other objects, "objects of law": on one hand, there is no reason why new technologies and services should escape the realm of law; on the other hand, it may be the case that existing regulations need to be adapted to take into account the advent of new, unforeseen technological developments (e.g. certain provisions of privacy regulations become inapplicable in a pervasive computing context, intellectual property laws are challenged by the new distribution modes of electronic contents). Understanding precisely when this is the case and how regulations should evolve to cope with the new reality is a complex "technico-legal" issue with potential impact on both disciplines.
- ICT can also provide new enforcement mechanisms and tools for the benefit of the law: Privacy Enhancing Technologies (PETs) [Goldberg - 2007] help reduce privacy threats , certified tools can be provided to support electronic signature, DRM technologies are supposed to "implement" legal provisions and contractual commitments, computer logs can be used as evidence in courts, etc. At another level, data mining or knowledge management systems can be applied to the extraction of relevant legal cases or the formalization of legal reasoning.

Generally speaking, legal and technical means should complement each other to reduce risks and to increase citizens' and consumers' trust in ICT : on one side, laws (or contracts) can provide assurances which are out of reach of technical means (or cope with situations where technical means would be defeated); on the other side, technology can help enforce legal and contractual commitments [Le Métayer - 2010c]. These interactions are quite subtle however and this synergy should not be taken for granted: if legal issues are not considered from the outset, technological decisions made during the design phase may very well hamper or make impossible the enforcement of legal rights; similarly, new regulations or contracts drafted without proper consideration for the possibilities offered by the technology are bound to remain ineffective. But the consideration of legal constraints in the design phase of an IT system is a challenge in itself, not least because of the gap between the law and computer science communities and the difficulties to establish a common understanding of the concepts at hand. In this paper, we argue that formal methods, both for fundamental and practical reasons, can help reduce this gap (Section 2). We illustrate the feasibility and the interest of this approach through examples in software liability (Section 3), privacy (Section 4), and compliance (Section 5). We then identify further challenges for both disciplines (Section 6), showing that the link between ICT and law is a fruitful research area both for computer scientists and for lawyers, and we conclude with a discussion on a methodology for interdisciplinarity (Section 7).

2 Formal Methods as a Link between ICT and Law

Beyond their many differences, ICT and law share a strong emphasis on formalism. This commonality is not without reason: in both cases, formalism is a way to avoid ambiguity and to provide the required level of rigour, transparency, and security. As an illustration, L. Fuller in his book "The morality of law" [Fuller - 1964] puts forward the following distinctive features of a legal system: (1) a set of rules (2) without contradiction, (3) understandable, (4) applicable, (5) predictable, (6) publicized and (7) legitimate. Even though they were obviously not proposed with such a comparison in mind, it is interesting to note that, among these features, the first five are also often used in computer science to characterize a good software specification and the sixth one can be related to the notion of open access to source code. The last one, legitimacy, is actually a key distinctive feature of legal normativity with respect to technical normativity. We come back to this critical issue in the conclusion.

As far as software is concerned, the fact that both disciplines refer to the word "code" is not insignificant and the exploration of the commonalities can be very fruitful - and not only from a theoretical perspective. Indeed, there are many situations where the frontier between the two notions seems to be blurring[1]. Just to take a few examples:

- Software contracts typically incorporate references to technical requirements or specifications which can be used, for example, to decide upon acceptance of the software by the customer or validity of an error correction request. In case of litigation, these specifications can also be used by the judge as they are part of the contract executed by the parties. In this perspective, the contract can thus be seen as an extension of the technical specification including further legal provisions such as intellectual property rights, warranty, and liability.
- The DRM technologies are supposed to implement legal provisions and contractual commitments about the use of digital content such as music or video.
- More and more transactions are performed on the basis of electronic contracts (SLA, or "Service Level Agreements" for web services, electronic software licenses, e-commerce contracts, etc.).

In fact, the convergence has developed so much that lawyers have expressed worries that "machine code" might more and more frequently replace "legal code", with detrimental effects on individuals. This topic has stirred up discussions in the legal community (see, for example, [Lessig - 2001], [Lessig - 2007] and [Reidenberg - 1998]) and is bound to remain active for quite a long time. Indeed, the implementation of contractual commitments by computer code raises a number of issues such as the lack of flexibility of automated tools, the potential inconsistency between computer code and legal code, the potential errors or

[1] Lawrence Lessig refers to East Coast Code and West Coast Code to denote respectively law and software code [Lessig - 2007].

flaws in the computer code itself, not to mention the legitimacy issue pointed out above.

In any event, the reality is that software code and legal provisions are increasingly intermingled, sometimes with complementary roles, sometimes in a fuzzy or conflicting relationship. It is also the case that legal provisions, just like software code, are assumed to meet specific goals or requirements. Just like software specifications, these requirements can be defined precisely, even formally (at least to a certain extent, because legal provisions must usually leave some room for interpretation by the judge) using dedicated logics (see, for example, [Farrell et. al. - 2005] and [Prisacariu, Schneider - 2007]). Based on this double observation, we argue that the first step for a fruitful and useful exploration of the relationships between legal provisions and software code is the definition of a formal framework for expressing the notions at hand, understanding them without ambiguity, and eventually relating or combining them. Stated in so general terms, one may wonder whether such an approach can really be turned into practice and if it can have any impact beyond theoretical considerations. In the next three sections, we show the feasibility of the approach through its application to three areas in which the link between law and technology is of prime importance, namely software liabilities, privacy and compliance.

3 Liability Issues in Software Engineering

As mentioned above, software contracts between professionals ("B2B contracts") typically include references to technical requirements or specifications which can be used, for example, to decide upon acceptance of the software by the customer or liability in case of failure of the system. It is often the case that these requirements are not stated very precisely though, which may lead to misunderstandings between the parties or potential conflicts between them during the execution of the contract.

The legal situation is often simpler, at least apparently, in typical licenses for "off the shelf" software, which usually include strong liability limitations or even exemptions of the providers for damages caused by their products. This situation does not favour the development of high quality software though, because software vendors do not have sufficient economic incentives to apply stringent development and verification methods (see, for example, [Anderson, Moore - 2009], [Berry - 2007] and [Ryan - 2003]). Indeed, experience shows that products tend to be of higher quality and more secure when the actors in position to influence their development are also the actors bearing the liability for their defects. In addition, the validity of contractual liability limitations and exemptions can sometimes be questioned. For example, most regulations provide specific protections to consumers which make such clauses invalid in B2C contracts. Even in B2B contracts, liability limitations are usually considered null and void when the party claiming the benefit of the clause has committed acts of intentional fault, wilful misrepresentation or gross negligence. Another case is the situation where the limitation would undermine an essential obligation of a party and would thus introduce an unacceptable imbalance in the contract [Steer et. al. - 2011]. This situation is more difficult to assess though, and left to the appraisal of the judge who may either accept the limitation, consider it null, or even fix a different liability level.

Whether liability clauses are defined too vaguely or unequally with risks of being invalidated in court, they result in contracts with high legal uncertainties, which is not a desirable situation, neither for business nor for society in general. The usual argument to justify this situation is the fact that software products are too complex and versatile objects whose expected features (and potential defects) cannot be characterised precisely, and which thus cannot be treated as traditional (tangible) goods. Admittedly, this argument is not without any ground: it is well known that defining in an unambiguous, comprehensive and understandable way the expected behaviour of software systems is quite a challenge, not to mention the use of such a definition as a basis for a liability agreement. But the fact that specifying entire software systems and all associated liabilities is usually out of reach does not mean that the most significant scenarios and sources of liabilities cannot be identified and formally specified. Actually, specifying formally all liabilities would not even be a desirable goal. Usually, the parties wish to express as precisely as possible certain aspects which are of prime importance to them and prefer to state other aspects less precisely (either because it is impossible to foresee, when signing the contract, all the events that may occur or because they do not want to be bound by overly precise commitments).

To address this need, we have proposed a framework providing different levels of services which can be used by the parties depending on factors such as the economic stakes and the timing constraints for the drafting of the contract [Le Métayer et. al. - 2010a]:

1. The first level is a systematic (but informal) definition of liabilities based on a library of (parameterized) legal clauses [Steer et. al. - 2011].
2. The second level is the formal definition of liabilities. This formal definition can be more or less detailed and does not have to encompass all the liability rules defined informally. In addition, it does not require a complete specification of the software but only the properties relevant for the targeted liability rules.
3. The third level is the implementation of a log infrastructure or the enhancement of existing logging facilities to ensure that all the information required to establish liabilities will be available if a claim is raised and will be trustable to be used as evidence for the case.
4. The fourth level is the implementation of a log analyser to assist human experts in the otherwise tedious and error-prone log inspection task.

Each level contributes to further reducing the uncertainties with respect to liabilities, and the parties can decide to choose the level commensurate with the risks linked to potential failures of the system.

The keystone of the formal specification of liabilities is the notion of "claim property". Basically, claim properties represent the grounds for the claims: they correspond to failures of the system as experienced by the users. In practice, for them to give rise to liabilities, such failures should cause damages to the plaintiff, but damages are left out of the formal model. As an illustration, a claim property can express the fact that a signature application has sent to the server a message indicating that a given user has signed a specific document (identified by a stamp) when the user has never been presented any document with this stamp [Le Métayer et. al. - 2011]. Claims can be expressed as trace properties using temporal or predicate logics. The choice of the language of properties does not have any impact on the overall process but it may make some of the technical steps, such as the log analysis, more or less difficult.

The liabilities arising under a given contract can be expressed as a function mapping claims and traces onto sets of (liable) parties. One way to define the liability function is to specify typical faults in the execution of the components and to associate a set of liable parties with each claim and combination of faults. Faults can be expressed in the same trace property language as the claims. Another possibility is to define a causality relationship between the occurrences of certain types of faults and failures [Goessler et. al. - 2010]. Causality has been studied for a long time in computer science [Lamport - 1978], but with quite different perspectives and goals. In the distributed systems community, causality is seen essentially as a temporal property. In [Goessler et. al. - 2010], we have defined several variants of logical causality allowing us to express the fact that an event e_2 (e.g. a failure) would not have occurred if another event e_1 had not occurred ("necessary causality") or the fact that e_2 could not have been avoided as soon as e_1 had occurred ("sufficient causality"). We have shown that these causality properties are decidable and proposed trace analysis procedures to establish them.

Another key design choice is the distribution of the log files themselves. Indeed, recording log entries on a device controlled by an actor who may be involved in a claim for which this log would be used as evidence may not be acceptable to the other parties. In [Le Métayer et. al. - 2010b], we have introduced a framework for the specification of log architectures and proposed criteria to characterize "acceptable log architectures". These criteria depend on the functional architecture of the system itself and the potential claims between the parties. They can be used to check that a log architecture is appropriate for a given set of potential claims and to suggest improvements to derive an acceptable log architecture from a non-acceptable log architecture. On the formal side, we have shown that, for a given threat model, the logs produced by acceptable log architectures can be trusted as evidence for the determination of liabilities: technically speaking, any conclusive evaluation of a claim based on these logs produces the same verdict as the evaluation of the claim based on the sequence of real events.

As far as the log analysis itself is concerned, we have proposed a formal specification of the analyser using the B method in [Mazza et. al. - 2010] and we have shown the correctness of an incremental analysis process. This result makes it possible to build upon the output of a first analysis to improve it by considering additional logs or further properties.

The overall approach has been applied to several representative case studies: an electronic signature application on a mobile phone [Le Métayer et. al. - 2011], a distributed hotel booking service [Le Métayer et. al. - 2010b] and a cruise control system [Goessler et. al. - 2010].

4 Privacy

Another area where technical and legal issues become more and more entangled is privacy. Even in countries where they benefit from apparently strong legal protections, many citizens feel that information technologies have invaded so much of their life that they no longer have suitable guarantees about their privacy. Indeed, the fact that the massive use of information technologies is the source of new risks for privacy is unquestionable. Many data communications already take place nowadays on the Internet without the users' notice and the situation is going to get worse with the advent of "ambient intelligence" or "pervasive computing". One of the most challenging issues in this context is the compliance with the "informed consent" principle, which is a pillar of most data protection regulations. For example, Article 7 of the EU Directive 95/46/EC states that "personal data may be processed only if the data subject has unambiguously given his consent" (unless waiver conditions are satisfied, such as the protection of the vital interests of the subject). In addition, this consent must be informed in the sense that the controller must provide sufficient information to the data subject, including "the purposes of the processing for which the data are intended".

Technically speaking, the consent of the subject can be implemented through a "privacy policy" which should reflect his choices in terms of disclosure and use of personal data. We have proposed an implementation of privacy policies through "Privacy Agents", dedicated software components acting as "surrogates" of the subjects and managing their personal data on their behalf. The subject can define his privacy requirements once and for all, with all information and assistance required, and then rely on his Privacy Agent to implement these requirements faithfully. However, this technical solution raises a number of questions from the legal side: for example, to what extent should a consent delivered via a software agent be considered as legally valid? Are current regulations flexible enough to accept this kind of delegation to an automated system? Can the Privacy Agent be "intelligent" enough to deal with all possible situations? Should subjects really rely on their Privacy Agent and what would be the consequences of any error (bug, misunderstanding. . .) in the process?

In order to shed some light on these legal issues, we have focused on three main aspects of consent : its legal nature (unilateral versus contractual act), its essential features (qualities and defects) and its formal requirements. In a second stage, we have drawn the lessons learned from this legal analysis to put forward design choices ensuring that Privacy Agents can be used as valid means to deliver the consent of the data subject [Le Métayer, Monteleone - 2009]. Several kinds of Privacy Agents have been proposed (Subject Agents, Controller Agents and

Auditor Agents) and the roles of the different actors involved in the process have been defined precisely. Privacy policies themselves can be expressed in a restricted (pattern based) natural language. In order to avoid ambiguities in the expression of the policies, a mathematical semantics of the privacy language has been defined. This mathematical semantics characterizes precisely the expected behaviour of the Privacy Agents (based on the privacy policies defined by their users) in terms of compliant execution traces.

This work is an illustration of the privacy by design approach which is often praised by lawyers as well as computer scientists as an essential step towards a better privacy protection [Le Métayer - 2010d]. The general philosophy of privacy by design is that privacy should not be treated as an afterthought but rather as a first-class requirement during the design of IT systems; in other words, designers should have privacy in mind from the moment they define the features and architecture of a system and throughout its life cycle. The privacy by design approach has been applied in different areas such as electronic health record systems [Anciaux et. al. - 2008], location based services [Kosta et. al. - 2008], electronic traffic pricing ([De Jonge, Jacobs - 2008], [Balash et. al. - 2010]). More generally, it is possible to identify a number of core principles that are widely accepted and can form a basis for privacy by design. For example, the Organization for Economic Co-operation and Development (OECD) has put forward the following principles [OECD - 1980]:

- The collection limitation principle: lawful collection of data with the "knowledge or consent" of the data subject.
- The purpose specification and use limitation principles: specification of the purposes, collection and use limited to those purposes.
- The data quality principle: accuracy of the data, relevance for the purpose and minimality.
- The security principle: implementation of reasonable security safeguards to avoid "unauthorised access, destruction, use, modification or disclosure of data".
- The openness and individual participation principles: right to obtain information about the personal data collected, "to challenge" the data and, if the challenge is successful, to have the data "erased, rectified, completed or amended".
- The accountability principle: data controllers should be accountable for complying with these principles.

These principles have inspired a number of privacy regulations. They are also very much in line with the European Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data[2].

One must admit however that the take-up of privacy by design in the ICT industry is still rather limited. This situation is partly due to legal and

[2] The latter however puts more emphasis on the explicit consent of the subject.

economic reasons: as long as the law does not impose binding commitments[3], ICT providers and controllers do not have sufficient incentives to invest into privacy by design. But part of the reason is also technical: computer scientists have devised a number of privacy enhancing tools, but no general methodology is available to integrate them in a consistent way to ensure suitable privacy properties. In the same way as the use of cryptography is by no means a guarantee of security, the use of privacy enhancing tools does not bring by itself a guarantee of privacy. The next challenge in this area is thus to go beyond individual cases and to establish sound foundations and methodologies for privacy by design [Le Métayer - 2010d].

As a first step in this direction, we have proposed a formal framework for the implementation of the data minimization principle which stipulates that the collection should be limited to the data strictly necessary for the purpose. This framework allows us to define:

- The service to be performed, expressed as a set of equations characterizing the values to be computed.
- The actors involved.
- The requirements of each actor, defined as constraints on the variables used in the equations. Typical requirements may express the fact that a given value should not be collected or that it should be collected only in a specific form (aggregated, sampled, ciphered, etc.).

An operational semantics defines the effect of each action on the state of the actors and the underlying threat model (possibilities of tampering with variables, properties of cryptographic commitments or secure components, etc.). An inference system based on this operational semantics allows us to derive properties of the variables such as, for example, the fact that an actor can obtain enough knowledge to identify an error (or potential fraud) in the computation of a variable. This inference system can be used to explore the design space systematically, for example to infer architectures meeting the requirements of the parties (e.g. limited disclosure on side the data subject and ability to discover certain types of frauds on the side of the data controller) or to detect conflicting requirements.

Even if much work has still to be done in this area, as suggested in Section 6, we believe that the added value of the formal approach for privacy by design can be decisive: in addition to the usual benefits (precise definitions of assumptions and requirements, detection of inconsistencies, verification), it can be used to provide designers with practical means for the systematic exploration of the available options and for the justification of their architectural choices.

5 Compliance

Compliance is yet another legal area where the use of formal methods can be very beneficial. Nowadays, organizations have to comply with a growing

[3] This situation might change in Europe though, with the revision of the European Directive 95/46/EC which is currently under discussion.

number of legal rules stemming from law, regulations, corporate policies or contractual agreements. These rules have a potential impact on all their activities and breaches may lead to different types of damages, including financial losses, lawsuits, competitive disadvantages and disrepute. But manual compliance enforcement or verification are error prone and tend to exceed the capacity of most organizations. IT systems, even if they cannot provide the full answer to this complex issue, can help organizations in the management and monitoring of their obligations.

To address this need, we have proposed a framework based on a formalism called Flavor[4] [Thion - 2011] which provides the following combination of features:

- Contrary to duty obligations [Prakken - 1996]: a contrary to duty obligation consists of a primary obligation and an alternative obligation which becomes effective when (and if) the primary obligation is breached. Contrary to duty obligations are useful to express penalty clauses in contracts as well as compensations and sanctions for breaches of legal rules.
- Combinations of temporal and deontic modalities: one of the most pervasive characteristics of legal rules is the interaction between temporal ("always", "eventually") and deontic ("obligatory", "prohibited") modalities [Pace, Schneider - 2009]. This interaction clearly appears in constructions such as "shall . . . within . . . days after . . . ", or "must . . . within . . . ". Actually most obligations or prohibitions come with a deadline which may be defined by a fixed date, by a delay or by a specific event.
- Conditions and contexts: legal rules are generally expressed as abstract and general statements intended to be applied in a variety of circumstances. To this aim, the wording of a legal rule generally distinguishes the effect of the rule (action to be performed or prevented) and its context of application. The context of application typically involves parameters and data (e.g. price, reference number, time, . . .) related to specific events.

We have defined a semantics for the language which is suitable for the implementation of an auditing tool and which avoids the paradoxes and counter-intuitive meanings often arising in modal logics. Based on this semantics, we have provided criteria for analysing obligations and defined a strength ordering which can be used to reason on contractual clauses. The framework has been illustrated with typical business contracts and privacy policy rules.

6 Further Challenges

The contributions sketched in the previous sections have been presented here only for illustrative purposes, to show that the use of formal methods as a link between law and software code is not a purely speculative idea. Needless to say, much work remains to be done, not only in the application areas mentioned here, but also more generally on the interactions between law and ICT.

[4] Formal Language for A posteriori Verification Of legal Rules.

The notion of causality, for example, is extremely rich and complex, and it represents in itself a very fruitful area for further research. First, it would be interesting to express causality in a more abstract way, independently of the underlying computation and communication models, and to establish precise links with related notions in dependability, diagnosis and security. The study of the correspondence between formal characterisations and legal definitions of causality is obviously another area for further work. To this respect, it would also be interesting to introduce probabilities in the formal framework in order to reflect certain interpretations of causality in the legal sense, the differences between several causes being often considered with respect to their effects on the likeliness of the occurrence of the damage [Busnelli et. al. - 2005].

As far as compliance is concerned, a number of key issues have already been investigated but still require further work [Pace, Schneider - 2009], especially to ensure that formal models are consistent both with the legal views and with the practical constraints that organizations have to face [Governatori et. al. - 2006]. Among these issues, we should mention the possibility to detect conflicts between obligations [Fenech at. al. - 2009], to verify statically the compliance of a system or to monitor its actions in order to ensure that no obligation can be violated. There are also other significant aspects of the problems faced by organizations that are not fully taken into consideration by previous work:

1. The first aspect is the dynamic nature of contracts. Most companies execute new contracts on a daily basis and these contracts usually have termination provisions. The execution of new contracts and their termination represent a substantial part of the difficulty and must be integrated in formal frameworks for obligations.
2. The second aspect is the fact that organizations have to cope with events which are not within their control and must take them into account before deciding to enter into new legal agreements.
3. The third aspect is the observation that, in practice, conflicts between obligations do not necessarily take the form of sheer contradictions: the situation is often more subtle, for example the consequences of a breach can be more or less significant; sometimes the conjunction of obligations does not lead to a contradiction but to a detrimental reduction of the choice space of the organization. Last but not least, following point 2 above, potential breaches may or may not be under the control of the organization.

Needless to say, privacy is also an area where a lot of difficult problems remain to be solved (and many others are bound to arise in the future). The main challenges in this area concern both privacy by design and privacy evaluation. First, much work remains to be done to turn privacy by design into practice, both from a formal point of view and from a methodological perspective. The work sketched in Section 4 is a first step in this direction, addressing the minimization principle, but other principles such as, for example, transparency or accountability require more attention. Indeed, Transparency Enhancing Tools (TETs) have been called for by lawyers (see, for example, [Hildebrandt - 2008] and [Hildebrandt - 2006]) but they have not yet become a reality. These tools should provide ways for individuals to understand how their personal data (and, ideally, any data that can be used in a processing with potential effects on them) are collected, generated, managed, transferred, etc. The transparency requirement is of upmost importance in a context where information flows are growing dramatically and the data mining and inference techniques become more and more powerful.

The concept of accountability is already applied in certain areas such as the finance and public governance and it is likely to be included in the future version of the European Directive on Data Protection 95/46/EC currently under discussion. Accountability puts emphasis on "how responsibility is exercised and making it verifiable". Technically, it involves at least two dimensions: transparency (making processing visible) and security (in the sense of integrity and non repudiation of the accountability data). More generally, it is a multi-faceted notion, involving social, legal and political aspects. The relationship between accountability and privacy is also rather complex: accountability can be used to strengthen privacy rights (when it applies to the data controllers) but it can also represent a threat to privacy (when it applies to the data subjects, e.g. within financial transactions, or when it requires to record excessive amounts of personal data). More research is needed to clarify the technical definition of accountability and associated requirements (in line with the legal view), to ensure that accountability can go hand in hand with privacy, and to provide practical and trustworthy implementation methods and tools helping organizations to comply with the transparency and accountability requirements.

The definition of realistic and formally grounded measures of privacy is also a challenging task. Several proposals have been made to define relevant privacy metrics such as k-anonymity [Sweeney - 2002] or differential privacy [Dwork - 2006] but the problem remains open : some of these metrics do not necessarily measure a true protection level because they are vulnerable to certain types of attacks, while others provide guarantees which are difficult to reach in practice because they would result in unacceptable reductions of data utility. Also, it is not clear whether a single type of metric can be suitable for different application areas corresponding to varied needs and expectations in terms of privacy.

Needless to say, the above challenges concern the lawyers as well as the computer scientists. As an illustration, key notions of European data protection laws such as "personal data", "informed consent", "subject" or "controller" are challenged, if not made ineffective, by new technologies. Another illustration is the role of the consent of the subject in current data protection regulations. Some lawyers have expressed the view that putting too much stress on consent can lead to an exclusively individualistic view of privacy disregarding the collective value of privacy as a fundamental right. To avoid this drift, clear limitations should be placed on the legitimacy of consent: for example, certain data should be considered as inalienable and, when consent is authorized, it should come with strong requirements in terms of transparency to ensure that the subject really understands the consequences of his consent. But where to place the red line and on which grounds are difficult questions, and, as suggested above, the effective implementation of transparency and consent delivery is also a challenge for computer scientists. In certain cases, the implementation of transparency can even create conflicts with the legal protection of intellectual property rights (e.g. with respect to profiling algorithms). Legal, social and technical dimensions are thus strongly intermingled and an interdisciplinary approach is required to make any progress on these topics.

7 Conclusion: Interdisciplinarity in Practice

In this paper, we have argued that the development of the new information society raises a number of challenges which require stronger collaboration between lawyers and computer scientists. But setting up this kind of interdisciplinary collaboration also represents a challenge in itself, especially when it concerns disciplines which have very different histories and cultures and have built very different modes of functioning (research development, assessment, collaborations, etc.). On one hand, each discipline should keep its criteria of excellence; on the other hand, disciplines should find together new ways of creating, communicating and evaluating research results. Needless to say, researchers in each discipline have also to overcome any misconception about the other discipline and accept points of views from “outsiders” questioning their own discipline. As shown by the pieces of work sketched in Sections 3, 4 and 5, this objective is not out of reach though. Drawing on the lessons of these projects, we believe that such an inderdisciplinary collaboration should be based on a precise methodology and it should include at least the following steps:

- The comparison of the terminologies and notions used in the different disciplines: often the same term is used in two disciplines with different meanings or intentions; vice versa, it also happens that the same notion is named in different ways in different disciplines. Indeed, there is no shortage of terms which may lead to confusion in discussions between lawyers and computer scientists (e.g. “causality”, “accountability”, “effectiveness”, “proof”, “security” , etc). The analysis of these shifts is a pre-requisite for mutual understanding; in addition it can shed new light on each discipline and help refining the underlying concepts.
- The comparison of the procedures, modes of operation in the different disciplines: for example how are the instruments conceived, how are they accepted, monitored, revised? How is their effectiveness defined and measured? Such a comparison, in addition to enhancing mutual understanding, can be a source of inspiration and improvement in each discipline. For example, the legal procedures can be a source of inspiration to provide a more transparent or democratic process for the development of new technologies, to devise technologies with “contradiction means” (possibility to bypass the procedure implemented by the tools). Vice versa, new ideas can come from the technology concerning criteria such as evolutivity or effectiveness.

– The study of the problems at hand in an iterative way where each discipline can bring its own analysis, views and findings before confronting them to the findings of the other disciplines and, based on this enlarged view, proposing a refined solution, which can be confronted again to the other ones.

Beyond research collaborations, the complex issues raised in this paper also question the relationships between the legal and technological normativities: how can the law face the "over-effectiveness" of technological norms and their opaque dissemination mode? How can the stability required by the legal systems adapt to the fast evolution of technologies? At what stage should the legal dimension be taken into account in the deployment of new technical infrastructures? How to introduce a mode of contestation or democratic debate in the elaboration of technological choices? Needless to say, these issues go beyond law and technology, they are by essence political, which should not come as a surprise considering the tremendous (and still growing) impact of information technologies on our everyday life [Jacobs - 2009].

Acknowledgments. This work was partly supported by the French ANR projects LISE under the grant ANR-07-SESU-007 and FLUOR under the grant ANR-07-SESU-005.

References

[Anciaux et. al. - 2008] Anciaux, N., Benzine, M., Bouganim, L., Jacquemin, K., Pucheral, P., Yin, S.: Restoring the patient control over her medical history. In: 21st IEEE International Symposium on Computer-Based Medical Systems, pp. 132–137. IEEE Computer Society, Los Alamitos (2008)

[Anderson, Moore - 2009] Anderson, R., Moore, T.: Information security economics – and beyond. Information Security Summit (IS2) (2009)

[Balash et. al. - 2010] Balasch, J., Rial, A., Troncoso, C., Geuens, C., Preneel, B., Verbauwhede, I.: PrETP: privacy-preserving electronic toll pricing. In: Proc. 19th USENIX Security Symposium (2010)

[Berry - 2007] Berry, D.M.: Abstract appliances and software: the importance of the buyer's warranty and the developer's liability in promoting the use of systematic quality assurance and formal methods. Scientific Literature Digital Library and Search Engine (2007), http://www.scientificcommons.org/42749418

[Busnelli et. al. - 2005] Busnelli, F.D., et al.: Principles of European tort law. Springer, Heidelberg (2005)

[Dwork - 2006] Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)

[Farrell et. al. - 2005] Farrell, A.D.H., Sergot, M.J., Sallé, M., Bartolini, C.: Using the event calculus for tracking the normative state of contracts. International Journal of Cooperative Information Systems (IJCIS) 14(2-3), 99–129 (2005)

[Fenech at. al. - 2009] Fenech, S., Pace, G., Schneider, G.: Automatic Conflict Detection on Contracts. In: Leucker, M., Morgan, C. (eds.) ICTAC 2009. LNCS, vol. 5684, pp. 200–214. Springer, Heidelberg (2009)

[Fuller - 1964] Fuller, L.L.: The morality of law. Yale University Press, New Haven (1964)

[Goessler et. al. - 2010] Gössler, G., Le Métayer, D., Raclet, J.-B.: Causality analysis in contract violation. In: Barringer, H., Falcone, Y., Finkbeiner, B., Havelund, K., Lee, I., Pace, G., Roşu, G., Sokolsky, O., Tillmann, N. (eds.) RV 2010. LNCS, vol. 6418, pp. 270–284. Springer, Heidelberg (2010)

[Goldberg - 2007] Goldberg, I.: Privacy-enhancing technologies for the Internet III: Ten years later. In: Digital Privacy: Theory, Technologies, and Practices (2007)

[Governatori et. al. - 2006] Governatori, G., Milosevic, Z., Sadiq, S.W.: Compliance checking between business processes and business contracts. In: EDOC, pp. 221–232. IEEE, Los Alamitos (2006)

[Hildebrandt - 2006] Hildebrandt, M.: Profiling: from data to knowledge. DuD: Datenschutz und Datensicherheit 30(9), 548–552 (2006)

[Hildebrandt - 2008] Hildebrandt, M.: Profiling and the rule of law. Identity in the Information Society 1(1), 55–70 (2008)

[Jacobs - 2009] Jacobs, B.: Architecture is politics: security and privacy issues in transport and beyond. Data Protection in a Profiled World. Springer, Heidelberg (2010)

[De Jonge, Jacobs - 2008] De Jonge, W., Jacobs, B.: Privacy-friendly electronic traffic pricing via commits. In: Degano, P., Guttman, J., Martinelli, F. (eds.) FAST 2008. LNCS, vol. 5491, pp. 143–161. Springer, Heidelberg (2009)

[Kosta et. al. - 2008] Kosta, E., Zibuschka, J., Scherner, T., Dumortier, J.: Legal considerations on privacy-enhancing location based services using PRIME technology. Computer Law and Security Report 24, 139–146 (2008)

[Lamport - 1978] Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Communications of the ACM 21(7), 558–565 (1978)

[Le Métayer, Rouvroy - 2008] Le Métayer, D., Rouvroy, A.: STIC et droit: défis, conflits et complémentarités. Interstices (2008), `http://interstices.info/jcms/c_34521/stic--et--droit--defis--conflits--et--complementarites`

[Le Métayer, Monteleone - 2009] Le Métayer, D., Monteleone, S.: Automated consent through privacy agents: legal requirements and technical architecture. The Computer Law and Security Review 25(2) (2009)

[Le Métayer et. al. - 2010a] Le Métayer, D., Maarek, M., Mazza, E., Potet, M.-L., Frenot, S., Viet Triem Tong, V., Crépeau, N., Hardouin, R.: Liability in software engineering: overview of the LISE approach and application on a case study. In: International Conference on Software Engineering, ICSE 2010, pp. 135–144. ACM/IEEE (2010)

[Le Métayer et. al. - 2010b] Le Métayer, D., Mazza, E., Potet, M.-L.: Designing log architectures for legal evidence. In: 8th International Conference on Software Engineering and Formal Methods, SEFM 2010, pp. 156–165. IEEE, Los Alamitos (2010)

[Le Métayer - 2010c] Le Métayer, D.: (ed.) Les technologies au service des droits, opportunités, défis, limites. Bruylant, Cahiers du CRID 32 (2010)

[Le Métayer - 2010d] Le Métayer, D.: Privacy by design: a matter of choice. Data Protection in a Profiled World, pp. 323–334. Springer, Heidelberg (2010)

[Le Métayer et. al. - 2011] Le Métayer, D., Maarek, M., Mazza, E., Potet, M.-L., Frenot, S., Viet Triem Tong, V., Crépeau, N., Hardouin, R.: Liability issues in software engineering. The use of formal methods to reduce legal uncertainties. Communications of the ACM (2011)

[Mazza et. al. - 2010] Mazza, E., Potet, M.-L., Le Métayer, D.: A formal framework for specifying and analyzing logs as electronic evidence. In: Davies, J. (ed.) SBMF 2010. LNCS, vol. 6527, pp. 194–209. Springer, Heidelberg (2011)

[Lessig - 2001] Lessig, L.: The future of ideas: the fate of the commons in a connected world. Random House (2001)

[Lessig - 2007] Lessig, L.: Code and other laws of cyberspace, Version 2.0. Basic Books, New York (2007)

[OECD - 1980] OECD guidelines on the protection of privacy and transborder flows of personal data. Organization for Economic Co-operation and Development (1980)

[Pace, Schneider - 2009] Pace, G.J., Schneider, G.: Challenges in the specification of full contracts. In: Leuschel, M., Wehrheim, H. (eds.) IFM 2009. LNCS, vol. 5423, pp. 292–306. Springer, Heidelberg (2009)

[Poullet - 2006] Poullet, Y.: The Directive 95/46/EC: ten years after. Computer Law and Security Report 22, 206–217 (2006)

[Prakken - 1996] Prakken, H., Sergot, M.J.: Contrary-to-duty obligations. Studia Logica 57, 91–115 (1996)

[Prisacariu, Schneider - 2007] Prisacariu, C., Schneider, G.: A formal language for electronic contracts. In: Bonsangue, M.M., Johnsen, E.B. (eds.) FMOODS 2007. LNCS, vol. 4468, pp. 174–189. Springer, Heidelberg (2007)

[Reidenberg - 1998] Reidenberg, J.: Lex informatica: the formulation of information policy rules through technology. Texas Law Review 76, 3 (1998)

[Rouvroy - 2008] Rouvroy, A.: Privacy, data protection and the unprecedented challenges of ambient intelligence. Studies in Ethics, Law and Technology. Berkley Electronic Press (2008)

[Ryan - 2003] Ryan, D.J.: Two views on security and software liability. Let the legal system decide. IEEE Security and Privacy (2003)

[Steer et. al. - 2011] Steer, S., Craipeau, N., Le Métayer, D., Maarek, M., Potet, M.-L., Viet Triem Tong, V.: Définition des responsabilités pour les dysfonctionnements de logiciels : cadre contractuel et outils de mise en oeuvre. Actes du colloque Droit, sciences et techniques: quelles responsabilités, LITEC, collection Colloques et Débats (2011)

[Sweeney - 2002] Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10(5), 557–570 (2002)

[Thion - 2011] Thion, R., Le Métayer, D.: FLAVOR: a formal language for a posteriori verification of legal rules. In: IEEE International Symposium on Policies for Distributed Systems and Networks. IEEE, Los Alamitos (2011)