

Law as code, code as law – general remarks on legal requirements engineering¹

Aernout Schmidt and Han Franken, Center for eLaw@Leiden

COMMERCE AND LAW IN THE DIGITAL ERA

In the expression ‘e-commerce and law’, ‘e-commerce’ refers to practical forms of trade that are mediated by electronic communication (like internet), by electronic processing (like authenticating a message) and by electronic processes (like software agents that are sometimes referred to as intelligent); ‘law’ refers to the legal aspects involved, not merely contract and commercial law.² Although practices of e-commerce stem from the seventies of the last century, the global emergence of internet in the nineties has nurtured the general feeling of a communication revolution taking place. The internet revolution brings us into a digital era and poses challenges to the paradigms of law and commerce alike, leaving considerable turbulence in the information industry and considerable room for legal research. The challenges will have to be faced by many, also by legal professionals. Traditional academic legal education may have to adapt itself accordingly. We wonder whether new areas of knowledge should be made available to law students who nourish the ambition to contribute to ordered commerce in the digital era. True to academic tradition, our argumentation starts where legal research fails us.



How to approach the research domain of e-commerce and law is open to discussion. Looking at IT and law in general, we distinguish three problem fields: open texture (i.e. judge made law by interpretation), legal engineering (the adaptation of IT practices in the law office, for instance) and inherent ambiguity (issued by the difference between the physical world and cyberspace). In 1997, the Dutch government published its leading principle³ and proposed that what counts as⁴ law off-line, should count as law online. This perspective promotes the further interpretative exploitation of open texture⁵ in law and seems appropriate when the new behaviour fits in current legal understanding and can be accommodated by case law in the civil-law tradition, like email may fit in regular mail regulation. In 1996/1998, looking at the IT threats and opportunities for the legal profession,

¹ Adaptation of the paper presented by Aernout Schmidt at the Leiden-Oxford staff exchange on e-commerce on may 16 and 17, 2001.

² See for instance the EU Directive 2000/31 EG (8-6-200) on Electronic Commerce (e.g., consideration 40 and articles 12-14).

³ Kabinetsnota Wetgeving voor de elektronische snelweg (Communication of the government on legislation for the electronic highway), 1997. This publication has been prepared mainly by legal academics.

⁴ Meant in the sense of John R. Searle, *The Construction of Social Reality*, Penguin Books 1995, p. 43.

⁵ As introduced by F. Waisman (*Verifyability*) in : A. Flew (ed.), *Essays on logic and language*, first series, Oxford 1951.

Richard Susskind adopted an essentially engineering perspective.⁶ This perspective (we baptize it the legal requirements engineering perspective) prompts legal research towards re-engineering legal services and processes. It may be directed to the electronic commerce and law domain, when new services and processes in electronic commerce are developed and deployed under the requirement of legal compliance (like the filter in the Napster case). In 1999, working from a constitutional-review perspective alien to the Dutch legal system, Lawrence Lessig coined the problem of inherent ambiguity, identifying the semantic void sometimes preventing the direct conversion of pre-internet law into cyberspace regulation.⁷ This perspective opens research up towards rethinking the law, specifically for cyberspace, if the old law remains silent to the new behaviour (like during the period of time it has been silent towards digital documents and signatures). In civil-law tradition, inherent ambiguity is accommodated by some attempts to legislation in general and by litigation in individual cases.

The three perspectives represent different positions in the legal system, relevant to electronic commerce. The open-texture perspective often coincides with *post hoc* legal assessment, as professed by courts, magistrates, solicitors and barristers in litigation. The engineering perspective is appropriate in commerce itself, notably when an e-commerce service is being designed and deployed. The service itself may change the business.⁸ The relevant legal research in e-commerce is business-oriented and *ex ante* concerned with modelling the service into an architecture that supports legitimate use. The responsibility risks rest *prima facie* on the (legal) persons designing and deploying the service, often relying upon legal advice. The latent-ambiguity perspective links tradition with times to come and belongs to litigation in some individual cases but mainly to the legislator – providing enacted rules, addressing areas that may be considered legally void.⁹

By definition, compliance with the law in electronic commerce presupposes applicable law that can be known. New questions of law arise, however, precisely because the IT revolution sustains new forms of civil behaviour unknown to, and not directly fitting in the current legal system. The tempo and the scale of emerg-

⁶ Richard Susskind, *The Future of Law: Facing the Challenges of Information Technology*, paperback edition, Oxford university press 1998.

⁷ The Lycos Tech glossary (see: <http://webopedia.lycos.com/TERM/c/cyberspace.html>) gives the following definition of cyberspace: “A metaphor for describing the non-physical terrain created by computer systems. Online systems, for example, create a cyberspace within which people can communicate with one another (via e-mail), do research, or simply window shop. Like physical space, cyberspace contains objects (files, mail messages, graphics, etc.) and different modes of transportation and delivery. Unlike real space, though, exploring cyberspace does not require any physical movement other than pressing keys on a keyboard or moving a mouse. Some programs, particularly computer games, are designed to create a special cyberspace, one that resembles physical reality in some ways but defies it in others. In its extreme form, called virtual reality, users are presented with visual, auditory, and even tactile feedback that makes cyberspace feel real. The term was coined by author William Gibson in his sci-fi novel *Neuromancer* (1984).”

⁸ Like legal IT-services may change the legal business, as predicted by Susskind *o.c.*

⁹ Our legal system of reference is the Dutch system, where the meaning of court decisions is limited to the case at hand and more general legal rules must be legitimated by parliament.

ing new behaviour in electronic commerce do not only challenge our understanding of their economics, they also challenge our legal understanding of them and of our ability to adapt the law in time and in a principled way.¹⁰ In electronic commerce and law these areas of legal ambiguity become visible through new IT services. The companies launching them are not just facing economic risks, they may be facing a legal void as well. In these instances legal requirements engineering and inherent ambiguity of the law go hand in hand. There are three predicaments here. One during litigation, interpreting a new world with old law. One for the e-commerce companies, designing services while in the dark about legal constraints (when they are not yet there). The third for the legislator, designing sustainable rules for an area of civil behaviour while in the dark of its shape next year. In litigation, legal professionals must (a) be able to distinguish between open texture and inherent ambiguity; in e-commerce, the company lawyer must (b) know what line separates safe from dangerous IT models when confronted with an area of legal ambiguity; in legislation, the legal professional must (c) know how the development and shape of future e-commercial services will go and will be influenced by the rules being designed. Using recent Dutch case rulings, we will analyse relevance and content of these three areas of knowledge for law students.



Recent Dutch case rulings on e-commerce do not come out of the blue. E-commerce is in many respects the equivalent for electronic data interchange (EDI), the analogous concept in pre-internet times. Starting from the seventies of the last century, their characteristics can be read from the architectures¹¹ of computer resources, of the communication infrastructure and of the application design. The architecture of computer resources has moved from mainframes to workstations and personal computers; the architecture of the communication infrastructure has moved from leased lines to internetworking; the architecture of application design has remained client/server based but is being complemented by peer-to-peer.¹² The architectural evolution of commercial computer messages has been of tremendous importance to the development of EDI and e-commerce – starting with the UN/Edifact¹³ initiative and unfolding through SGML, HTML and XML. These linguistic frameworks have supported the development of the artificial languages and dialects that help computers to communicate, negotiate and act independent of

¹⁰ *RIAA vs Napster* (<http://www.riaa.com/pdf/napsterdecision.pdf>) and *United States vs Microsoft* (http://www.usdoj.gov/atr/cases/ms_index.htm) are cases in point. See the tremendous amount of, often contradictory, legal opinions on these cases.

¹¹ Architecture plays an important role in Lessig's concept of regulation (Lessig 1999, *o.c.*).

¹² Mainframes, workstations, leased lines, internetworking, client/server and peer-to-peer are technical concepts lifted from the computer-science domain. Understanding them is necessary to all who want to understand what is going on in cyberspace. Since we expect a large part of e-commerce to realise itself in cyberspace, and our aim is to investigate how much of it should be common ground to legal professionals, we refrain from explaining the "technical" concepts during our argumentation. For reference, if necessary at all, we will use the exact formulation of the Lycos Tech Glossary (see: <http://webopedia.lycos.com>).

¹³ See for instance: <http://edicity.com/Resources/EDIFACT.htm>.

direct human intervention, thus moving much of EDI and e-commerce into cyberspace.

Functionally, EDI and e-commerce have developed through emulating existing off-line business practices (business to business or B2B) and through emulating mail-ordering services (business to consumer or B2C) towards emulating community co-operation. Virtual community co-operation uses team groupware to create and maintain services, often on a *quid pro quo* basis. The internet has revitalised this ancient business scenarios, as may be found in the open source movement that has proven to be competitive with Microsoft as well as in the MP3 sharing communities (Napster, Kazaa, MusicCity, *etc.*) that currently challenge the off line music industry. Perhaps virtual community co-operation will bring e-commerce to the consumer to consumer (C2C) level. It certainly belongs to the domain of e-commerce, since it generates and distributes economic value. It is so new, however, that its economics are not yet understood very well, and legally compliant business models are not yet stable.

EDI and e-commerce have generated many legal activities over time.¹⁴ They include efforts to understand the traditional concepts of agreement and contract formation (between IT-systems), archiving and deeds (paperless), general conditions (handing over), signatures (digital), competition (and closed B2B communities), data protection and crime (in cyberspace), intellectual property (copying, distribution, on computer programs, databases and web methods), freedom of information (web advertising), *etc., etc.* The methods used range from interpretation (open texture: *e.g.*, digital copying in intellectual property), through functional interpretation (legal requirements engineering: *e.g.*, digital signatures) towards rule construction (legislation: *e.g.*, data protection, intellectual property, cybercrime, telecommunications and e-commerce itself). Apparently, the principle of the Dutch Government, *i.e.*, that what counts as law off line should count as law online cannot be upheld over time: often, the creativity that is part of the open-texture and engineering solutions to e-commerce related legal questions are followed by the construction of specific cyberspace regulations. And almost as frequently, the resulting laws are out of date even before they have been ratified.

OPEN TEXTURE VS INHERENT AMBIGUITY

*RIAA vs Napster*¹⁵ promises to be a seminal case on virtual community systems in e-commerce. Napster is an information service allowing internet users to become members of a 'community', sharing music files in MP3 format. The files are exchanged directly between members' computers. An index, or catalogue, is

¹⁴ As for instance described by R.E. van Esch, *Electronic Data Interchange en het vermogensrecht (EDI and property law)*, Deventer: W.E.J. Tjeenk Willink 1999 and by Michael Chissick and Alistair Kelman, *Electronic Commerce: law and practice*, London: Sweet & Maxwell 2000.

¹⁵ See <http://www.napster.com/pressroom/legal.html>.

maintained centrally on a server and it is used by members to find out which user has which track of music on offer for downloading for free. The service has been created during the first half year of 1999 by a student, Shawn Fanning. On June 1st he launched the beta test version, making it available to 30 friends and acquaintances. A short time later, millions of music tracks were shared by millions of users all over the world and the American music industry became uncomfortable, so much, that the RIAA and others started a law suit on contributory copyright infringement. The judge ordered Napster Inc. to create a filter, preventing the sharing of copyrighted music tracks in the community, by checking them against a list provided by the RIAA. The filter was built, did not function faultlessly and by august 2001 the Napster community has been virtually dead.

~

Kazaa is a service, functionally superior to the Napster service. It not only supports audio-file sharing, but also document-, video- and program file sharing. Its architecture is also slightly different: there is no central server with a catalogue. The Kazaa catalogue is constructed dynamically on member computers from the information supplied by members. The Kazaa service is freely made available by Kazaa BV as a program file. On December 30th 2001 there were '558250 users online, sharing 80660K files (444720.0 GB).' Compared with the blooming period of Napster, much seems to have remained the same. Most members of the Napster community have moved on to Kazaa or a similar service and the RIAA has initiated a law suit against Kazaa in the USA. In the Netherlands, the District Court of Amsterdam has issued a summary judgment on November 28, 2001.¹⁶ Litigants were Kazaa BV, providing a proper peer-to-peer service for file sharing, and Buma/Stemra, defending the Dutch music industry. The initiative in the case was Kazaa's, its aim was to force Buma/Stemra into an agreement on copyright compensation. Buma/Stemra claimed in reverse that Kazaa should stop copyright infringement through the use of its service forthright. In the judgment, judge Orobrio de Castro, ordered amongst other things as follows:

... "orders Kazaa, within 14 days after ratification of this judgment, to take such measures that copyright infringement, distributing and copying music works that belong to the Buma/Stemra repertoire, using computer programs made available by Kazaa , is no longer possible, under forfeit of f 100.000,- per day with a maximum of f 2.000.000,-" ...

On January 31, 2002, the notice was long overdue. Kazaa had filed the appeal case and was told in an intermediary summary judgment that the forfeit would not be postponed.¹⁷ The Kazaa business was sold to an Australian company.

In our opinion the first-instance Kazaa judgment is ruled from an inherent-ambiguity position, and not argued very well at that. We give the arguments exchanged. Buma/Stemra stated:

¹⁶ See <http://www.rechtspraak.nl> LJN-nummer: AD6395 Zaaknr: KG 01/2264 OdC (in Dutch).

¹⁷ See <http://www.rechtspraak.nl> LJN-nummer: AD8749 Zaaknr: KG 02/0105 OdC (in Dutch).

“Buma/Stemra claims that Kazaa, by means of its computer program, infringes copyrights. The making available of music files by users to all other users constitutes to make them available to the public in the sense of the Dutch Copyright Act. Downloading these music files, made accessible to other users by Kazaa, constitutes infringing multiplication, not excluded in article 16b of the Dutch Copyright Act.”

Kazaa defended itself by arguing:

“Kazaa challenges Buma/Stemra’s claim that it infringes copyrights by making her computer program available to the public. As an intermediary party, Kazaa does not copy or distribute music files. Neither is there complicity or liability, because Kazaa acts as a producer of the computer program proper, and cannot be held responsible for the behaviour of its users.”

Judge Orobrio de Castro straightforwardly decided as follows:

“First must be put that Kazaa by giving its users the means, through its program, to download music files without licence, means an infringement of copyright.”

That was it, no further arguments provided. However, several observations have to be made. The Dutch Copyright Act does not accommodate contributory infringement (as is the case in the USA). Consequently, the judge has interpreted Kazaa’s behaviour as direct copyright infringement. The provision of an instrument (a computer program) by producer and distributor of that instrument, which may be used by the recipient to infringe, is seen as infringement proper. We do not think that reading the facts in this way can be classified as making use of open texture validly. Too many instances of the provision of instruments that may be used for good as well as for bad are kept outside the responsibility of the provider in the Dutch legal system. Hammers (murder) and copiers (copyright infringement), radio transmitters (copyright infringement) and nursery lights (weed) can be produced and distributed without the distributor becoming responsible for the use they are put to. Even fire arms are in this category, although there is a law prohibiting their free production and distribution. Making the maker of an instrument responsible for its use simply does not fit in the Dutch legal system very well. Apparently judge Orobrio de Castro has chosen an inherent-ambiguity position, perceiving an ‘empty space’ in the law, abused by Kazaa. Unfortunately, he considered his opinion self evident and left us in the dark about his reasoning. Perhaps even more unfortunate, his judgment may affect legal security in the software industry: it warns that possible abuse should be an important issue during design, implementation and deployment of computer programs in general, including e-commerce services. Consequently, it may be felt in the industry that possible abuse must be foreseen, and, if so, the development of tools otherwise considered beneficiary must be prevented. Imagine the development of e-mail services aborted out of fear for their possible abuse.

On March 28, 2002, the Amsterdam Court of Appeal ruled in the Kazaa-case as follows:¹⁸

¹⁸ See <http://www.rechtspraak.nl> LJN-nummer: AE0805 Zaaknr: 1370/01SKG (in Dutch).

... “Furthermore, the Court of Appeal mentions that Kazaa rightly opposes the first-instance ruling that first must be put that Kazaa infringes copyright by giving its users the means, through its program, to download music files without licence. As far as copyright infringing behaviour is at stake at all, the behaviour is performed by the users of the computer program and not by Kazaa. Making available the means to distribute or copy copyrighted works does not in itself constitute distribution or multiplication of those works. Neither is it the case, at least it cannot be assumed for the present, that Kazaa’s computer program is exclusively used for downloading copyright-protected works. Kazaa has shown in appeal a large number of examples of works that have permission of the author, have fallen in the public domain, do not enjoy copyright protection at all or are legally excepted from copyright protection. Moreover does Huizer in his ‘expert opinion’ provide examples of other uses. Buma/Stemra may state that the only, at least only essential, function of Kazaa’s computer program is to provide users with the means to exchange copyright protected works, but this claim is contradicted by Kazaa with force of arguments and is not further elucidated by Buma/Stemra at all. That the other use is meaningless counts doubtlessly for Buma/Stemra, but that does not imply its irrelevance to those other users.” ...

Clearly, the Court of Appeal takes the open-texture approach and is brought to a reverse decision. Where lies the line between inherent-ambiguity reasoning as used in the first instance and open-texture reasoning as used in appeal?

~

Prima facie, questions about the line between open texture and inherent ambiguity in e-commerce are no different from questions about that division in any other area. Presumably, its location is affected by the role and position of court judgments in the legal system under investigation. In the Netherlands, the role and position of courts forbid them to judge the legitimacy of enacted rules, prevent them from judging *contra legem* in individual cases, order them to give judgment in individual cases even if relevant enacted rules are nonexistent and deny them the right to formulate general rules. As a corollary, Dutch jurisprudence expects an open-texture line of reasoning, if valid, to take priority over inherent-ambiguity argumentation: the judge may not assume the role of the legislator. The importance of the distinction between open texture and inherent ambiguity lies in the necessity in the last case to employ non-legal argumentation to solve legal disputes. Naturally, the issue is not reserved for Dutch law and Dutch courts. It boils down to three daunting questions: what counts as valid legal reasoning in an open-texture context, what counts as valid legal reasoning in an inherent-ambiguity context and what separates the open-texture context from the inherent ambiguity context.

~

What lines of legal reasoning are valid has been researched and commented upon extensively. It is not our aim to enter into the swamp of academic dispute in this area of jurisprudence. For making my point here we need some practical characteristics to break down the field heuristically, so we can go on with the main argument as soon as possible. We have chosen to use two dimensions for this,

irreverently and mercilessly¹⁹ borrowed from the work of the constitutional law scholar Bobbitt²⁰ and the psychologist Fodor²¹ respectively. Bobbitt analysed the types of arguments used during the history of constitutional review in the USA and comes up with a limitative²² list: historical (addressing the intentions of the rule makers), textual (addressing the present sense of the rule wording), doctrinal (addressing the principles and relations derived from precedent), prudential (self-consciously addressing role and position of the party or institution that is reasoning), structural (addressing implicit meaning from the existence and role of legal institutions) and ethical (arguments addressing the *ethos* of a²³ jurisdiction or *politeia*). Fodor is interested in natural and in artificial intelligence and in how human knowledge representation relates to knowledge representation in IT. He finds the distinctions of abstraction layers as a basic similarity and considers the positioning of abstraction layers relative to reasoning of fundamental importance. Imposing Fodor's abstraction layers upon legal argument we distinguish six layers: beliefs (like in a free market, the ten commandments), principles (like in human rights, in the constitution), rules (like in acts, for instance democratically formulating exceptions to human rights), policies (like in administrative plans, interchange agreements and common-law precedent), individual judgments (like in civil-law precedent, in transactions) and facts (making up the facts of a case). Types of argument content can be related to levels of abstraction in the following framework:

	History	Text	Doctrine	Prudence	Structure	Ethics
Beliefs						
Principles						
Rules						
Policies						
Judgments						
Facts						

Table 1: Abstraction levels and argument types

The framework helps to present our heuristic distinction of valid from invalid legal reasoning in the different contexts. First of all, the different levels of abstraction represent different sources for legal arguments. In an open-texture context, only facts, judgments, policies, rules and principles are valid sources for argument, and only the reasoning is valid, which brings facts within the constraints of valid judgments, that are within the constraints of valid policies, that are within the constraints of valid rules, that are within the constraints of valid principles. In normal circumstances, beliefs do not provide valid sources for legal arguments in

¹⁹ Irreverently, because put to use in a very blunt, practical and generalizing way that does not and cannot respect the finesses of their fine work.

²⁰ Philip Bobbitt, *Constitutional Fate – theory of the constitution*, Oxford University Press 1982.

²¹ J.A. Fodor, *Psychosemantics*, MIT press 1987.

²² Despite of his claim not to be exhaustive on p. 8, the exercise proposed on p. 94 and 95 suggests the opposite (Bobbitt, *o.c.*)

²³ Here we generalize over Bobbitt's typification – he mentions the *ethos* of *American* polity.

an open-texture context and neither are ethical arguments admissible – anyway, in a stable *politeia*, arguments at the belief level and arguments of the ethical type are most often coherent with valid legal reasoning.

If not – when an open-texture judgment is incoherent with beliefs and ethics, when valid open-texture reasoning is not possible at all – then there is a case to be made for a shift in perspectives and try inherent-ambiguity reasoning. Valid inherent ambiguity reasoning allows, if necessary, ethical arguments at all levels and allows, if necessary, common beliefs to provide a closure for reasoning.

Administration	Accommodation
Accessibility	Transparency
Confidentiality	Flexibility
Reliability ²⁴	
Authenticity	

Inherent ambiguity presents itself when there is a dispute, that cannot be solved using valid open-texture legal reasoning. This happens often, when IT-solutions are outside reasonable interpretation of the facts in traditional law, but somehow need to be dragged into the legal system. Lessig

Table 2: Franken’s principles

shows some wonderful examples in American constitutional law and we will encounter an example in Dutch law presently. My (heuristic) claim is, that legal reasoning in an inherent-ambiguity context may validly use ethical arguments and common beliefs. In Dutch jurisprudence, Franken²⁵ has proposed six ethical principles to guide reasoning about the use of IT in cases of legal ambiguity. In e-commerce cases, they rest upon two beliefs: (1) that a business must at all times keep a valid administration and (2) that legal rules, if generated by future law must be accommodated in future business procedure. We mention these principles in Table 2. They were introduced in 1993 and have in the meantime become more or less self-evident. These principles can be seen on the one hand as a specification of the general principles of law which are applied in every off line law practice. In the ‘real’ world we live with principles of tort law, principles of contract law, principles of reasonable government decision-making etc. We can point at legal notions as the fair play principle and *audite et alterem partem* in the procedural law, *ne bis in idem* in the criminal law, carefulness in operations and the *nemo auditur* rule in civil law, the prohibition of *détournement de pouvoir* in administrative law etc. These principles can be specified and applied to the technical and the societal demands of IT relations and will give the practitioner, the judge and the legislator the possibility to express a rational consideration of interests and recognition of values.



²⁴ This principle is often referred to as ‘integrity.’ Reliability (claiming a computer service to behave in accordance with its specifications) implies integrity, which often has the restricted connotation of messages not being corrupted during communication.

²⁵ H. Franken, *Juridisch Theoretische achtergronden (Jurisprudential background)* in: H. Franken, H.W.K. Kaspersen and A.H. de Wild (eds.), *Recht en computer (Law and Computer)*, 4th edition, Kluwer 2001, p. 33-74.

We look back at the three cases summarily presented above for illustration. In the Napster case, in the USA, contributory copyright infringement could be validly argued with open-texture reasoning, because it is a legal concept in the USA at the rule level and the facts fit the rules. The Napster software was made with the intention to use it for sharing music files, also, and mostly, copyrighted music files. The architecture of the service allowed Napster Inc. to make an end to naïve, abusive music sharing by implementing a filter in the server software. The remedies ordered by the judge, making the music industry provide a list with copyrighted works and making Napster Inc. implement a filter, preventing the sharing of listed music files is generally considered within the limits of valid legal ruling in the USA.

Had the Napster case been filed in the Netherlands, its result could have been different. The concept of contributory copyright infringement does not exist in Dutch copyright law and the available legal notions of tort and criminal complicity would have been very difficult to prove and use successfully. It seems reasonable that an open-texture result would have been considered contradictory to common beliefs and commercial ethics by many. In Dutch polity, the principled way to handle the case would have been to judge in accordance with existing law, generating sufficient public indignation to prompt new legislature. However, our current preference for what has become known as our *poldermodel* (i.e. consensus model), resulting in many deviations from principled methods, might have induced the court to settle things directly via precedent, venturing to express a new policy.²⁶ It seems to us that judge Orobrio de Castro has tried to do just that in the Kazaa case in first-instance. An ethical principle that may have been behind his judgment could be the lack of flexibility of the Kazaa program. Once made available to the public, it drifted outside the influence of the distributor, can still be distributed freely by others and cannot directly²⁷ be adapted by the Kazaa company for future use, if future law would require it. This issue is of great importance: Lessig's digital commons, including the open source movements, do depend on architectures that prevent influence by the distributor. In the Kazaa appeal, a more principled approach has been applied.

We wonder what the American legal system will in its turn make of the Kazaa case; due to its multi functionality and its true peer-to-peer architecture, it may be much more difficult than in the Napster case to construe contributory infringement from an open-texture position in a valid way.

~

²⁶ In this, it seems that the principled divide between common-law and civil-law systems is reduced, as widely defended in the literature.

²⁷ This seems more of a theoretical than of a practical problem: IT services are constantly adapted and *versioning* of services is general practice. Indirectly, flexibility could be implemented through adaptations in later versions of the program.

Legal professionals handling e-commerce using open-texture reasoning need to be able to decide on the legal meaning of legal concepts in trade law, for business situations, that are realized with a significant support of IT processes. From trade-law perspective, we summarised these concepts by mentioning legal areas of interest: the traditional concepts of agreement and contract formation (between IT-systems), archiving and deeds (paperless), general conditions (handing over), signatures (digital), competition (and closed B2B communities), data protection and crime (in cyberspace), intellectual property (copying, distribution, on computer programs, databases and web methods), freedom of information (web advertising). Legal professionals that address e-commerce issues need to know what tools and models are standard available and what their functions are in these legal area's, of interest to the e-commerce and law community. Mentioned in an unordered manner, they need to know about virus scanners and firewalls, about web-browsers, search engines and web servers, about databases, about encryption, security, intranets and virtual private networks, about chat functions and groupware, about mail functions and pay functions, about logging and testing, about content/workflow management functions, about content communication standards and transport communication standards, about communication scenarios, about TCP/IP and ports, about agents and worms and sniffers, about media players and operating systems, about client-server and peer-to-peer architectures, about objects, messages and methods. Not at the level of IT professionals, but at the level of sufficiently understanding them functionally²⁸, sufficiently to be able to identify their relevance and position (as in Table 1) in open-texture and inherent-ambiguity legal reasoning. We cannot yet relate these IT-functions with the legal areas of interest mentioned in an orderly manner. First we will have to break down the engineering approach in a similar manner as we have been doing with the open-texture approach in the current chapter.

SAFE VS DANGEROUS BUSINESS MODELS: COMPLIANCE ENGINEERING

The Dutch Society of Estate-Agents (NVM) has a web service that lets house hunters specify their requirements and subsequently provides a complying list of real-estate objects for sale. In the back-end of the service operates a database that is maintained by the NVM members collectively. The service is open to all and free of charge. The Dutch newspaper De Telegraaf has subsequently designed, implemented and deployed a web service named El Cheapo. One of the functions of El Cheapo is similar to the NVM-service. Moreover, El Cheapo employs an intermediary software agent that automatically communicates with the NVM-service and presents the NVM-results, as part of its reply to the users request. Although the presentation of the results does specify its sources, and although further inquiries about specific real-estate objects is handled through direct linking

²⁸ Many of these have been treated that way in A.H.J. Schmidt and L. Mommers, *Informatie en communicatie (Information and Communication)* in: H. Franken, H.W.K. Kaspersen and A.H. de Wild (eds.), *Recht en computer (Law and Computer)*, 4th edition, Kluwer 2001, p. 1-31.

to the NVM data repository, the NVM organisation was not amused by El Cheapo and filed suit. The resulting litigation between NVM and De Telegraaf (we later refer to it as the El Cheapo case) has thus far been through three stages. A first-instance summary judgment by the The Hague court awarded the NVM claim of database right infringement, the appeal summary judgment by the The Hague Court of Appeal awarded the De Telegraaf claim and the cassation judgment by the Supreme Court of the Netherlands redirected the case back to the appeal-court level with instructions that seem to be in favour of the NVM. The El Cheapo case is still work in progress, it seems, and in an area of legal insecurity. In this chapter we adopt the legal requirements engineering perspective in order to examine the knowledge needed by the legal professional that has the task to advise De Telegraaf during design, implementation and deployment of the El Cheapo service. The task implies *ex ante* knowledge of the legal risks involved and it implies careful tracking of the (al)location of responsibilities and liabilities during design, implementation and deployment of the service as well.

~~~~~

Imagine being employed by De Telegraaf in an early stage of the El Cheapo project – even before a functional design of the service has been made. Imagine that the chief executive officer of The Telegraaf is, for sometime now, pregnant of an idea that will bind a considerable amount of readers: an internet service with houses for sale, where consumers can specify wishes, and where the service in reply will provide information on the available houses for sale. As a major country-wide daily newspaper, De Telegraaf has a famous weekly special with advertisements of real-estate for sale, it could become the backbone source of information for the service. Additional information can be found on the internet, where several sources on real-estate for sale are in the offering for free. The CEO makes a sketch of a business plan and decides the service to be economically feasible. He attracts an IT-business as contractor and concurrently appoints you to provide legal advice throughout the project, to be presented as a part of the milestone reports he is to receive. To prepare your work plans, indeed, in order to do your work professionally, you will have to know what the milestones are, what issues your reports are to address and in what context they will be presented.

~~~~~

IT projects have in the past often proven difficult to manage. During the sixties and seventies of last century, millions and millions of any currency have been lost in failed IT projects and a considerable effort has been made in computer science to construct effective knowledge about methods that reduce the risks in IT-development projects. The work has converged in ‘unified’ heuristics for planning and realisation of computer services. A huge heap of learned writing has emerged in this area, ranging from addressing the formal (*e.g.*, Codd’s²⁹ famous *relational*

²⁹ E. F. Codd: A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM* 13 (6): 377-387 (1970)

data model and Dijkstra's³⁰ almost as famous *structured programming*) via method comparison and fusion (e.g., Wieringa's *requirements engineering*)³¹ and object orientation (e.g. Jacobson, Booch and Rumbaugh's³² trilogy) to finally including the psychological and organisational (e.g. Schreiber et. al.'s³³ *knowledge engineering*). The tremendous amount of work in this area has resulted in much better manageable development projects, resting on a stable paradigm – so much so, that in the IT world specialization has resulted. It makes good sense to be knowledgeable about the paradigm as well as about the specialists if you want your advice to make any impact (or even sense) in the non-legal culture where the decisions are, and have to be, made.

The paradigm is treacherously simple: if you want to develop a computer service, you take the idea, make a model of it in natural language and translate that to more formal models until it is a working computer program. Unlike while developing a subway system, where the idea will also be modelled into formalities, it needs not to be realised in tunnels and steel. The seductive and wonderful thing about the most formal model in the IT development process is that it can be fed into a simple, generic machine and will spring into life forthright – operating. Then and there, the model is its own object, so to speak. In this line of reasoning, the meaning of a compiled computer program is purely referential to its own behaviour. No wonder that computer scientists tend to be reluctant towards anti-realist philosophy entering their discipline: if the computer does not behave in accordance with its program, it assumes a contextual meaning (e.g., computer behaviour when the power supply gets out of order) that is simply wrong – the program does not show its *true* behaviour. This attitude is firmly rooted in the IT-discipline, and rightly so – be it not without its drawbacks. Rightly so, because in the IT-business much depends on trust on proven products, and *ceteris paribus*, software products can be proven in a formal way. Dangerously so, because *ceteris* tend not to remain *paribus*, and because the meaning of a computer program in a social world is rather its behaviour and use related to the original idea in an organisational context, then to the most formal model in the context of an operating computer. The history of IT-development methods has, without forgetting about the formal, gradually shown additional attention to the organisational meaning of the service being developed. The mainstream paradigm of the IT world is now also concerned with the construction of a working model of an idea, as used in an organisation. Herewith it has shed much of its simplicity. The literature mentioned unfolds the modelling process into descriptive and operational themes. Addressing these themes has become good practice in the IT world and may consequently be of importance to the practitioners advice.

³⁰ O.,-J. Dahl, E.W. Dijkstra and C.A.R. Hoare, *Structured programming*, Academic press 1972.

³¹ Wieringa, R.J., *Requirements Engineering: Frameworks for Understanding*, Wiley, 1996.

³² Most relevant of these is: Jacobson, I., Booch, G., Rumbaugh, J., *The Unified Software Development Process*, Addison-Wesley, 1999.

³³ Schreiber, G., Akkermans, H., Anjewierden, *Knowledge Engineering and Management*, MIT Press, 2001.

The descriptive themes in IT service development are best understood as chapters³⁴ in the report, describing the model of the idea under realisation. There are five chapters: on the contextual organisation (e.g., De Telegraaf, internet, newspaper publics), on processes and tasks performed by the organisation (e.g., publishing news and commentaries, specials, advertisements and information on real-estate for sale), on agents that perform these tasks and their qualification as either automated (e.g., web-publishing real-estate for sale) or not (e.g., writing commentaries), on the communication between the agents – where communication between automated agents is fully formalised (e.g., between web servers and databases) and the description of mixed communication gives the user interfaces (e.g., between users and web services) and on the knowledge that the different agents need to perform their tasks (e.g., about the information format in other web services to be used automatically). Together, these chapters provide the model of the idea to be realised in its context; we name it the *contextual model*. Where and when the legal paragraphs in the different chapters are due we will have to address later, because we have to sort out some procedural aspects first.



The contextual model of an IT service under development is a moving target. It starts in very general and summary terms as the sketch of a business model. It finishes in extremely formal and specific tones, often incorporating the source code of executable computer programs in its appendices. As the descriptive element of the paradigm, the contextual model will reflect growing stages, closely related to the process of development in a procedural sense.

The first effective procedural view (again developed in the seventies of last century) of the generalised IT development process was represented by the waterfall metaphor, breaking down the work in an almost self-evident sequential chain of IT sub-projects – the general idea being that an information analysis must be available in order to make a functional model, that implementation should not start before the functional model had been completed, that testing could not take place before implementation, that the service should not be deployed before it had been tested and that maintenance of the service should be considered only after substantial use experience had been gained. Again, the waterfall metaphor is of a treacherously simple nature. It has proven to be of great value, but has in the meantime proven to be almost impossible to uphold in real world situations. Consequently, the procedural development paradigm has unfolded into themes rather than sequential sub-projects. As we see it, six of them have settled in the nucleus of unified methods: requirements analysis, functional design, implementation and graphical design, testing, deployment and maintenance.

³⁴ These chapters have been analyzed and described very well by Schreiber et. al, *o.c.* for academic education. They need 439 pages. Within the constraints of this contribution, we treat them like we treated Bobbitt and Fodor: irreverently, because put to use in a very blunt, practical and generalizing way that does not and cannot respect the finesses of their fine work, which is the result of a continuous research line started in 1982 by Wielinga and Breuker at the research group on artificial intelligence of Amsterdam University.



Requirements analysis addresses the specification of the initial idea in a manner supporting contract formation between initiating company and the contractor of the IT development project. From a legal orientation, several kaleidoscopically scattered aspects require attention. First, the context for agreement formation should be valid – implying clearly defined responsibilities and sufficient available information to support valid agreement. The initial contextual model is performative here. Requirements analysis thus unfolds the initial idea in context, addressing all chapters in terms of the functional requirements the initiating company wishes for and the contractor promises to meet and deliver. Naturally, you would insist on a clear wording of the promises made, since the contracting parties will almost always show considerable asymmetries in their knowledge – the company CEO seldom being a computer scientist and the contractor seldom being *au courant* with the intricacies of the business it will make the service for. The requirements analysis theme thus aims first of all at elevating their understanding to make true agreement possible. Second, as legal advisor of the initiating company you would consider the risks involved and the relevance of legal solutions to possible problems. As a legal advisor you too will have to understand what is going on here. For instance, the functional use of publicly available and free internet services provided by others like the NVM will have become visible at this stage and it is your task to push the breaks and explain the legal risks involved. It may well be that De Telegraaf will discontinue the project if it considers these risks unacceptable. Third, you should prevent falling in the natural trap of assuming that the requirements analysis theme, including contract formation, is definitive before further steps are taken. Very often, almost every time, the development process itself propels parties’ understanding into progression and will unveil the need for contract adaptation. Moreover, the IT-world around has not frozen into immobility during the development process and it is highly probable that new and better tools become available during the ride. As a legal advisor you should anticipate and choose contract wording accordingly. Fourth, you should be aware that the contractor will almost always need subcontractors. Information asymmetries will be incumbent there also.

The IT-world has developed several specialisms that are suitable for the themes under investigation. The requirements analysis theme is often addressed by a mixture of information- and organisation-analyst expertise; the functional design theme is often looked after by system-analyst expertise; implementation and graphical design by programming and human-machine interface expertise respectively; testing has to be approached from the formal angle by application programmers, and from the organisational angle by a mixed team of expected users and representatives of the initiating company and the contractor; deployment is a completely different cup of tea, handled by

Information analyst
Application programmer
System analyst
System programmer

Table 3: IT specialisms

network- and system programming specialists; activating the maintenance theme either means breaking open the initial contracts during the ride, or implies starting a new development process altogether. Breaking down the IT-development process in themes naturally leads to the appointing different specialists to be responsible for different themes. Because addressing the themes cannot be usefully organised sequentially, it should be organised in parallel project groups, with clear conditions and concurrency points (milestones) for inter-theme communication. Your advice to incorporate these conditions and milestones in the relevant agreements between contractors and sub-contractors makes sense, because the current breakdown is in accordance with what has become good practice in IT and may consequently be used in a normative manner.

The *functional design* theme addresses the functional specification of tasks to be performed by agents that are themselves or communicate with electronic agents in the service under development. In this theme, the focus is on computer applications, the context is reduced to users. As a result, the functional design theme will add agent related content to the chapters of the contextual model, and in a semi-formal format at that. After all, functional designs tend to be jotted down in the unified modelling language (IT-people say UML) . You will be inclined to leave the theme to the specialists. Several legal issues may need your attention, however. First, because functional models are more formally expressed than requirements. This simply implies loss of context-oriented semantic wealth and gain of computer-oriented semantic accuracy, and consequently many choices have to be made. This may lead to discrepancies that should be herded outside the danger area wherein the resulting service becomes unacceptable. An agreement on prototyping the design for evaluation might be taken into consideration. Second, because communication amongst electronic agents and communication between users and electronic agents may raise issues of standardisation, privacy, intellectual property and, using ethical argument, issues of accessibility, confidentiality, reliability and authenticity. Here it is, that you will have to provide important advice (and that you may find a second chance if you missed out on the database protection issue during requirements analysis). Here it is also, that you may provide information that can help the design to result in a legally compliant service. Here it is then, that you start fiddling with the design models and your advice becomes compliance engineering. We will address compliance engineering separately, after completing our current breakdown of the IT development process.

The *implementation and graphical design* theme is concerned with two disciplines. Implementation is a programmer and system engineering activity, extending the contextual model in the areas of the design theme. No single service is developed from scratch, nowadays. Using UML during design shows an object oriented approach, which has been prompted by the urge not to have to redesign functional program objects that have been made available elsewhere. There is a huge reservoir of available objects out there, that is continually being supplemented. Before coding new objects, research is necessary for available

resources. Decisions must be made about using them and about acquiring the rights to do so, if necessary. You could advise to review and discuss these decisions in an early stage. The graphical design theme is performed by graphical artists and adds to the contextual model in area of agent-user communication. Graphical design is extremely important and needs be approved by the principal contractor separately. Self-evidently, the legal issues of intellectual property must be addressed in your advice.

The *testing* theme is complex and is legally important: acceptance or refusal of the resulting service belongs to it. In the requirements engineering theme the conditions for acceptance and refusal are set, initially, and, if necessary, reset when the initial agreement is adapted. Pilot testing of the design models may be agreed upon. Agreements about procedures for conflict resolution are necessary.

The *deployment* theme addresses the issue of live hosting and maintaining the service on servers in the network. It adds heavily to the contextual model in the organisational area. The issue is very much technical and system- and network engineers will do the work. The knowledge asymmetries will traditionally be huge. Legally, the use of service level agreements has become standard. They regard many issues pertaining to interchange agreements. The problems of scaling should be addressed also if there is any chance of the service becoming a country-wide or world-wide success. In an internet context, things can get out of control real quick.

The *maintenance* theme has two sides, one side aims at improving the quality of the service and may be realised through closure of the current development process and restart is after relevant experience has been gained during operational use. The other side is concerned with continuity – which may be challenged by changing circumstances in the context (new versions of TCP/IP or HTML become popular, the original contractor is out of business, software objects used in implementation or network and hosting service providers are discontinued). Two issues may be legally addressed here: (1) contractual provisions about making available the source code may be of solace in some of these eventualities and (2) requiring the parties active in the themes to provide adequate documentation, thus contributing to the knowledge chapter of the contextual model and to the ethical principles of flexibility and transparency.

~

The reader who has persisted thus far may ask himself why there is no theme, devoted to the operational use of the service. In our breakdown, this issue is dispersed over several themes and chapters. The deployment theme addresses the service provider side of operational use, while users are represented as agents, and their behaviour is modelled in their sections in the tasks- and communication chapters.

In our breakdown, we identified several areas of legal interest. Of essential legal importance are the task, agent and communication chapters in the design theme. These are the areas that accommodate compliance engineering. Compliance engineering constitutes the regulating architecture in the sense of Lessig's (*o.c.*) 'code as law'. We now focus on compliance engineering in the El Cheapo case.

~

Earlier, in your capacity as an advisor of the CEO of De Telegraaf, you told him that difficulties may arise when the El Cheapo service under construction will use publicly available data from data bases that may nonetheless be protected by law. Several threats have to be faced, you mention. The data in these databases may be protected by copyright on their own merits; using the data may pass by the advertisements on the home page of the databases, resulting in damages; trade marks may be passed by in a similar manner; the structure of the database may be protected by copyright; substantial sums of money may have been invested in acquiring, management or presentation of the data, resulting in database law protection against. The CEO asks you to provide risk assessments of all these threats with respect to the NVM service. You find that the copyright threats are nonexistent since there is no personal connection between the presentation of information on the houses for sale and the maker of these presentations, hardly any individual choice is made. You find that passing by the home page of the public service will hardly be a risk, if it is performed in a manner that does not violate database protection since then it will be almost identical to the issue of deep linking,³⁵ provided that El Cheapo generates a list of available houses and uses the technique of deep linking for providing further data on individual houses. Consequently, the important questions concern whether substantial sums of money have been invested in acquisition, management or presentation of the data in the NVM service and whether the El Cheapo use of the NVM database constitutes substantial use of the database. The first question seems easily and factual. In first-instance judge Du Pon states the following:

... "Set aside that generally speaking it is highly probable that the maintenance of a countrywide collection of 45.000 records that – considering their short-lived reliability – must be updated daily in order to keep them up to date continually will require a quantitative substantial investment," ...

and your advice that, knowing these facts, you insist on the substantial-investment criterion to be fulfilled is reasonable – we further leave the issue alone.³⁶ More to the point is the additional criterion of substantial use of the database. Reviewing the **kranten.com** case mentioned earlier, you will advise to make use of the public NVM service in the following manner: let the search agent come up with a list of

³⁵ A question decided upon in the **kranten.com** case by the Rotterdam court in a summary decision of 22-08-2000, LJN-nummer: AA6826 Zaaknr: 139609/KG ZA 00-846 (in Dutch).

³⁶ This notwithstanding the Appeal Court's decision denying this reading of the substantial-investment criterion – which, by the way, was reversed again by the Supreme Court. Anyway, this criterion is not of interest for the compliance engineering issue, since it is not directed towards efficient, but to compliant IT design.

addresses and prices of the relevant houses for sale, integrate the list with the list based on the relevant houses from De Telegraaf's own housing special and other lists from other sources, and make additional information available through deep linking. In this operational model, you will run only a small risk of a successful NVM claim against De Telegraaf based on database protection right infringement. This is an example of what we consider to be compliance engineering. To do so, you need to understand the tasks-, agent- and communication chapters of the contextual model at the design-theme level.

In the El cheapo case the following happened. In first instance judge Du Pon said:

... "According to Article 1, Section 1, under c, of the Database Act is to be understood under 'use request': the permanent or temporary transport of the contents of a database or a part of it on another medium, irrespective of method and format employed. In our summary judgment is what El Cheapo does after a visitor of her site has made a search request inside this definition. For it has been established that El Cheapo in such a case does copy a selection of files of NVM's database, stores them on its own server and saves them there for some time." ...

Interestingly, the court's reading of what is going on after a use request on the El Cheapo server (as design description of the tasks performed) is so far beside the point, that the appeal court decided the wording had to be rephrased:

... "The principal claim 3 includes that in argument ... of the first-instance ruling the word 'file' has been incorrectly used to denote one single house. This claim is founded, but cannot lead to annihilation of the ruling. In those arguments will be read 'record' or 'file-element' instead of 'file.'" ...

This in itself is an argument for some IT-schooling in legal education. Anyway, nobody doubts or doubted that El Cheapo makes us of 'use requests' in the sense of the Database Act. More to the point is whether these use requests result in substantial use. Judge Du Pon handles the issue as follows

... "The principal defence against NVM's claim that such a substantial use is indeed the case implies that El Cheapo does not decide on the size of the use, but that it depends on the way the user specifies his conditions (general or specific). Thus, El Cheapo merely offers the possibility to the user, to make El Cheapo request a substantial part." ... "This defence does not hold ... The president himself has established by experience that in such a case often at least 10 files [records] appear, predominantly from the NVM database. This must count as a substantial part of that database. Added can be that smaller amounts of requested and presented files [records] often in any case will be considered as a qualitative substantial part of NVM's database, taking into account the type of the data (houses for sale) and the interest that also a limited number of these data can have for a visitor that more specifically is looking for a particular house and therefore has formulated a limited request." ...

and informs us that substantial use must not be interpreted with the size and interest of the selection, relative to size and interest of the data collection in the database, but relative to size and interest of the selection the requesting user needs. In the appeal case, no further light was shed on the substantial use issue. And the supreme court redirected the case to the appeal court level, where, hopefully there

will be found a more satisfactory interpretation. However, we will have to wait and see.

As an advisor, you don't yet know of these decisions. You start an open-context analysis of the relevant pieces in the design chapter, looking for the IT-reconstruction of a substantial use issue. You keep the Dutch Database Act from 1999 and the EU directive 99/6/EU in mind. The EU directive mentions the prevention of free-riding duplicating services as an overall goal. According to the explanatory memorandum of the Dutch act, the goal of the Database Act is to prevent damages to the investor in a database through its re-use, it states as a principle that the investor has to prove these damages in order to make his claim stick. Combining these with the El Cheapo design, you decide that there is a real risk here: the type of re-use by El Cheapo is precisely what the directive and the Database Act aim to prevent. Looking at the way this aim is worded in the text of the Act, however, you find that presenting 10 out of the 45.000 records in the NVM database is hard to understand as a substantial selection. The legal literature does not provide consistent guidance on the issue. Finally, you are aware of the enormous impact any first legal decision on this issue will have on future e-commerce developments. Consequently, in your advice you will argue that there is a considerable risk of a court order to discontinue the service under development if one of database providers challenges the re-use in court, because its operation may be in conflict with the goals of the Database Act; that, nonetheless, the wording of the Act may find small selections for re-use acceptable; that otherwise the financial risks are limited to proven damages through re-use.

From a legal engineering point of view, you will mention that it will not be possible to remodel the service into compliance with the goals of the EU directive, without fitting in agreements with the providers of databases that are meant to be used. If De Telegraaf is accepting the risk of the service eventually being closed down by a court order, it may choose to accommodate the textual reading of 'substantial use' as relative to database size. This is easy, by limiting the list of results to be presented with full record information. Another approach would be to remodel the service into only supplying a list of deep links to the relevant records in the NVM web service to the user, for these links are themselves not records in the NVM-database. From the legal engineering point of view, you could investigate whether remodelling the El Cheapo service to sort and manipulate the list of links before being presented to the user will be acceptable. Even this is not clear – manipulation of selection results does technically imply copying the selection to the El Cheapo back-end temporarily, and feeding it into the manipulation-agent might in itself be considered as re-use.

~

In fact, the service under investigation will have to be launched, if at all, facing genuine legal ambiguity (what is substantial use?) – even where the applicable rules are very young. Engineering legal compliance into the El Cheapo service

boils down to legal risk analysis of the e-commercial services on internet and (re)modelling the functional design towards an acceptable level of the legal risks to be faced. To perform this task professionally, a legal advisor must be able to read and understand the design chapters of the contextual model as well as relate them to legal issues. As a matter of fact, this may also be a prerequisite for judges that have to decide on controversial e-commerce practices in summary cases.

LAW AS CODE – CODE AS LAW: THE FUTURE OF E-COMMERCE

The substantial use clause has been formulated in the EU directive and adopted by the Dutch legislator. The legislator has to take into account how e-commerce on cyberspace will prosper and suffer, with and without the rules he is making. He will have to foresee the ways e-commerce practice might grow, the ways judges will read the open texture concepts in individual cases and the result of e-commerce practice complying with this reading. His task is a difficult one, if we can take the Kazaa and El Cheapo cases to be exemplary.

From the inherent-ambiguity perspective, the legal reasoning takes into account beliefs and principles that are considered implemented in the legal system if we consider the open-texture context. The whole area of inherent-ambiguity reasoning seems to be lifted to a more abstract level: here we are not producing solutions to individual conflicts, but rules that are to guide all. We are no longer talking about Kazaa and El Cheapo, but about trends in area's of e-commerce more in general. If we focus, as we do, on e-commerce in cyberspace, we distinguish these areas by types of e-commerce (business to business, business to consumer and consumer to consumer) and by levels of services that support activities in cyberspace. These levels have been adopted from the framework Lessig³⁷ uses in his discussion of the way intellectual property law influences life on cyberspace. At the physical network level rest the physical resources whereon cyberspace exists, like copper, fibre and frequencies. At the code level we make a distinction between the areas of communication code (where the protocols and communication interfaces are) and application code (where application programs like word processors and Kazaa-clients and web servers are). At the content level, we find the knowledge and information, as provided and understood by businesses and consumers.

	B2B	B2C	C2C
Content			
Code: application			
Code: communication			
Physical networks			

Table 4: e-commerce areas

An interesting aspect of Table 4 is, that it helps to show dependencies between industries. From a functional point of view, the content industry on cyberspace must make use of applications, communications and networks; the software industry

³⁷ Lawrence Lessig, *The future of ideas*, Basic Books 2001.

must make use of communications and networks and the communication industry must make use of networks. The services on the lower levels are the infrastructures for the services at higher levels. In reverse, these infrastructure services are dependent on the demand from higher levels: when there is no interest for content, the infrastructure services remain idle. Consequently, the four layers of e-commercial services are convicted to each other. As Lessig pointed out, the way these layers are constricted by intellectual property law may influence the manner in which new services may be developed and deployed. The Kazaa and El Cheapo cases make that clear, too. The other area of legal influence is likely to be competition law, considering the interdependencies. No wonder, competition law is the main issue in telecom regulation.

Another interesting aspect that can be shown using Table 4, concerns the areas of consumer to consumer e-commerce. These areas will be populated by peer-to-peer applications. Although in our current legal system we have plenty of attention for consumer protection in the business to consumer areas, consumer to consumer e-commerce is new. The main legal issues it will generate seem to be: When should consumer to consumer e-commerce be treated as business to business, when as business to consumer and when as a *sui generis* business relationship? How can business to business regulation be upheld under the numerical explosion of “businesses” in consumer to consumer e-commerce? Are the infrastructure services at lower levels to be considered as ‘essential facilities’? Again, the Kazaa case gives us an indication how these issues may unfold in the future.

~

In *Code as Law*, Lessig brings four regulating forces to the fore, legal, social, economic and architectural. He focuses on the architectural regulating powers of code in areas of communication and application. These powers are emerging *per se*, through e-commercial practice: code as law, as it were, closely related to our legal-requirements-engineering position; they are emerging through law as well, leading to new implementations of regulatory architectures in code: law as code, as it were, closely related to the inherent-ambiguity position. His argumentation leads to a reserved approach to legislation for cyberspace, to a warning against naively translating traditional intellectual property law for cyberspace and the legal engineering of a ‘commons’ in cyberspace (where e.g. the open source movement lives), leaving room for free interaction and cooperation in communities. In *The future of ideas*, Lessig elaborates on the threats of intellectual property rights for cyberspace and argues for ‘commons’ at the physical level, at both code levels and at the content level, proposing changes in property law on the fly. His view on the future of cyberspace is dim, because of the direction e-commerce is taking under current law.

~

In our opinion, Lessig is right and wrong. Concerning the ‘commons’ we fully agree. Concerning the threats by current intellectual property rights for the

commons, we heartily disagree: current intellectual property rights shape and protect the commons. To translate the problem to the El Cheapo case level: nobody prevents those that have real-estate for sale, to start a community web service functionally equivalent to and content-wise better than the NVM-service. Very successful services of the type exist – in the Netherlands notably ‘www.tweakers.net’ as a hardware and software marketplace. In our opinion, it is perfectly all right when creator-related additional value is protected, as long as community-related additional value can be published in the commons and can be protected against being usurped by commercial free riders that bring it outside the commons with no value added.

With these arguments in mind, we reiterate to the future of e-commerce on cyberspace, from the inherent-ambiguity perspective. The central, practical problem in both the Kazaa case and the El Cheapo case are related to the possibility of parties, to know whether a file or a record in a database is actively protected by intellectual property law or not. When this information were available for the agents searching the internet, they could behave compliantly. When the information is not available, using and re-using information that is freely published on the internet may prove to be infringement. These risks do hamper the evolution of ideas and services alike. It is not the interpretation of ‘substantial use’ that will solve these problems in general, neither will fiddling with copyright- and database right protection periods solve much. Transparency of the protection status of published content will. We need the ©-sign back, in a digital version, and several more signs to label various database protection levels and GNU-licences.

For the time being, these labels have not been generally accepted in practice. It may be within the power of the legislator to introduce them. Of eminent importance for the future of the internet will be the choice about the default status of any digitally published unlabeled piece of information. A free default will fit nicely in the traditional ethos of cyberspace. Furthermore, protection by labelling will not be very laborious or difficult in e-commerce. In stead of making further data-protection law, we would advise the legislator to enact a simple cyberspace labelling paragraph in the Copyright and Database acts.

REQUIRED KNOWLEDGE

We wondered whether new areas of knowledge should be made available to law students who nourish the ambition to contribute to ordered commerce in the digital era. We distinguished three different perspectives for legal reasoning by legal professionals: the open texture perspective for litigation, the legal engineering perspective for legal advisors and the inherent ambiguity perspective for rule making. We provided a unified framework of themes and chapters, providing the contextual model of an IT-service and describing the basic architecture of good IT development practice. We sketched the structure of e-commerce on cyberspace,

using four service levels and three business types. We finally used the Kazaa case and the El Cheapo case to analyse the knowledge required by the legal professionals involved. For litigation professionals, we found it necessary to be knowledgeable about the legal meaning of existing IT-functions. For the advising professional, we found it necessary to be knowledgeable about the legal risks, that will be relevant to future IT functions. For the legislator, we found it necessary to be knowledgeable about ethos of and principles for cyberspace. The legal professional, contributing to ordered commerce in the legal era must know enough of IT, to be able to read and understand the analysis and design themes of contextual models and to be able to detect the legal issues involved. He must furthermore understand the ethos of internet, and the finesses of the ethical principles involved. We apprehend that an analogous argument can be made for legal professionals that are directing their attention to e-government. We further expect that, ten years from now, commerce will have mainly become *e-commerce* and government will have mainly become *e-government*. We think that academic legal education should accommodate these needs.