# Process Modelling towards e-Government – Visualisation and Semantic Modelling of Legal Regulations as Executable Process Sets

**Sebastian Olbrich[1] and Carlo Simon[2]**
**[1]Institute for Management Information Systems, Phillips University of Marburg, Germany**
**[2]Fachbereich Wirtschaftsinformatik, Provadis School of International Management and Technology, Germany**
Sebastian.Olbrich@zeon.de
Carlo.Simon@provadis-hochschule.de

**Abstract:** The paper discusses the visualisation and formal modelling of a legally regulated process. The approach is motivated by a historic retrospective. The technical innovation is not only to consider the given law when developing business process models - like many other approaches do - but to explicitly derive a process structure which is implicitly specified within the paragraphs themselves. To translate paragraphs into process models the Semantic Process Language (SPL) is used, since it enables us to articulate language structures into executable workflow models. The paper illustrates its approach with a demonstration example which considers the obligation right of Switzerland. It selects those paragraphs which participate in the definition of a causal ordering. The presented approach provides means for verifying whether process-like behaviour fulfils the selected paragraphs formally.

**Keywords:** e-Government, business process modelling, legal visualisation, legal design.

## 1. Introduction

Government processes are usually more regulated (by laws and other regulations) than business processes – especially those process parts that are carried out within a governmental organisation. For this reason, an abstract definition of legal concepts in terms of data and information, however, is not sufficient for the needs of modern organisational aspects like interoperability, process orientation and the need to build workflow management systems to support these concepts.

The coding of laws and obligations is typically done with the aid of natural language. Hence the visualisation of the encoded information is restricted to simple organisational mechanisms such as lists (numbered paragraphs). Dependencies between them are loosely and can be visualised by pointers to other paragraphs. Laws, however, are also operating guidelines for the behaviour of humans in their societies. They regulate how to behave in order to achieve specific goals and therefore encode process-like information of human behaviour. From this observation we derive our research thesis: *laws can be visualised and modelled like other governmental processes and these models can be used as guidelines to develop workflows*.

To confirm our thesis, we choose to build an executable and formally verifiable model, an approach based on Design Science (March and Smith 1995, Hevner et. al. 2004). The paper discusses the visualisation and modelling of a set of paragraphs of the Swiss Obligation law (Obligationsrecht, OR) in a process model. In the subsequent chapter we give an overview of historical approaches to visualising legally-defined processes. Chapter 3 discusses the state-of-the-art of novel approaches to legal visualisation on the example of the Swiss Obligation law. In chapter 4 we introduce the Semantic Process Language (SPL) as our modelling method. In chapter 5 we use the SPL to translate the paragraphs – illustrated in Chapter 3 – into process models. The paper closes with some concluding remarks and an outlook for possible applications.

## 2. Early visualisation of legal contexts

The fact that there is almost no effort to abstract the code of law in models seems to be rather strange – especially in Germany. Firstly, because there is an undoubted need to include legal constraints and regulations in almost any aspects of our lives and, secondly, because this code of law is being written in coded form for only a relatively short time.

Until the introduction of Roman law in Germany by the Staufer, law was articulated – in the name of God - by the ruling force (Laufs 1996). It was orally communicated until circa 1220-1235 when Eike von Repgow documented in the so called *Sachsenspiegel* (Lit. “Saxon Mirrow”) the local Saxon law in a mixture of texts


Reference this paper as:
Olbrich, S. and Simon, C. (2008) “Process Modelling towards e-Government – Visualisation and Semantic Modelling of Legal Regulations as Executable Process Sets” *The Electronic Journal of e-Government* Volume 6 Issue 1, pp 43 - 54, available online at www.ejeg.com

and pictures (Oppitz 1990). Since then most medieval laws were illustrated in pictures and sequences of pictures to give an overall comprehension of the context of a law and the consequences of violation to the common people (Schott 1986). The *Sachsenspiegel* served as a model and an eponym for other law books in many states of the Holy Roman Empire like the Augsburger *Sachsenspiegel*, the Heidelberger *Sachsenspiegel*, *Schwabenspiegel* etc.

The *Sachsenspiegel* contains two branches of law: common law (Landrecht) and feudal law (Lehnrecht). First is the law of free people, i.e. many people who belonged to a liege lord are not included. It contains important regulations concerning property rights, inheritances, matrimonies, the distribution of goods and the regulation of various civil disputes. It also regulates the criminal law and the constitution of courts. In terms of modern legal systems it can be thought of as including criminal and civil law. Feudal law determines the relationship between the different German states, for example the election of emperors and kings, feudal rights, etc. Though it has no modern equivalent, it can be compared to today's constitutional law.

Figure 1 depicts the election of the German king as a sequence of events according to the *Sachsenspiegel* of Heidelberg (around 1300) as an example. In the first image the three ecclesiastical Princes can be observed choosing the king by pointing to him. The second image shows Count Palatine of the Rhine as he hands over a golden key. The Duke of Saxony, standing next to him, carries his marshal's staff. Both are followed by the Margrave of Brandenburg who brings a bowl of warm water. The last image shows the newly elected king in front of the great men of the empire.

**Figure 1:** Election of the German king (Universitätsbibliothek Heidelberg 2006)

The major advantage of drawings (to be precise: the combination of drawings and explanations) to the purely textual coding of law is their quick overview of legal contexts. A specific description of the election procedure is also given in the text beneath. In order to create law books for common people, many rhymes, metaphors and literal description were used that can still be found in German proverbs, e.g.:

- "Wer zuerst kommt, mahlt zuerst" (First come, first served, literally: "Who comes first, grinds first"), which is a rule for the order for grinding corn by a miller.
- "Wo der Esel sich wälzt, da muss er Haare lassen" (lit: "Where the donkey rolls, he looses hairs"), describes a rule for the jurisdiction of courts.

 

As the legal documentation for more than 700 years, the *Sachsenspiegel* is still the most successful law book ever written. In Germany it was only replaced by the code of civil law in 1900. Quotations of the *Sachsenspiegel,* however, can still be found in many today's Supreme Court verdicts.

## 3. Novel approaches to legal visualisation

In recent years, lawyers, graphic designers, (art-) historians, psychologists and sociologists have again tried to answer the question of how to illustrate our code of law with pictures, models or other graphical representations in order to make it more comprehensible (Sachs 2005). The goal is to expose the pictures that are used in the legal discussion and in the legal terms and to create new illustrations of legal contexts. These illustrations can then be used to explain the meaning of laws to a broad audience. The discussion about *legal visualisation* is based on the thesis that in modern days the visualisation is not only needed in many aspects of daily life, but unavoidable in a world that relies more and more on multimedia presentations (Boehme 2005).

The potential of a combination of pictures and texts is well known– for example, in learning (Thiele 2000). Today, based on experiences with the visualisation of regulations over a long time, didactic and visual approaches are used (Ulbrich 2000) to address people interested in the subject of law such as students (Kreuzbauer 2005). This pedagogic and psychological background is the main difference between today's efforts and the goals of the *Sachenspiegel* in the past - since in the medieval times illiterates were the main subjects of address.

Modern legal visualisation is no longer designed to replace the code of law by a visual form of coding. It is rather an additional interpretation for a broader group of readers (Wegschneider 2005). Given a certain level of competence on the field of visual literacy (Pettersson 1993), the reader can quickly understand and judge legal contexts. Brunschwig (2001) defines in her PhD thesis a new kind of normative pictures (Rechtsnormbilder) that visualise the basic messages of paragraphs. The caption of such a normative picture is its paragraph's text (see Figure 2). A sequence of normative pictures explains a process defined by more than one paragraph. Brunschwig (2001) uses the Swiss obligation law (Obligationenrecht - OR 1911) as a demonstration example.

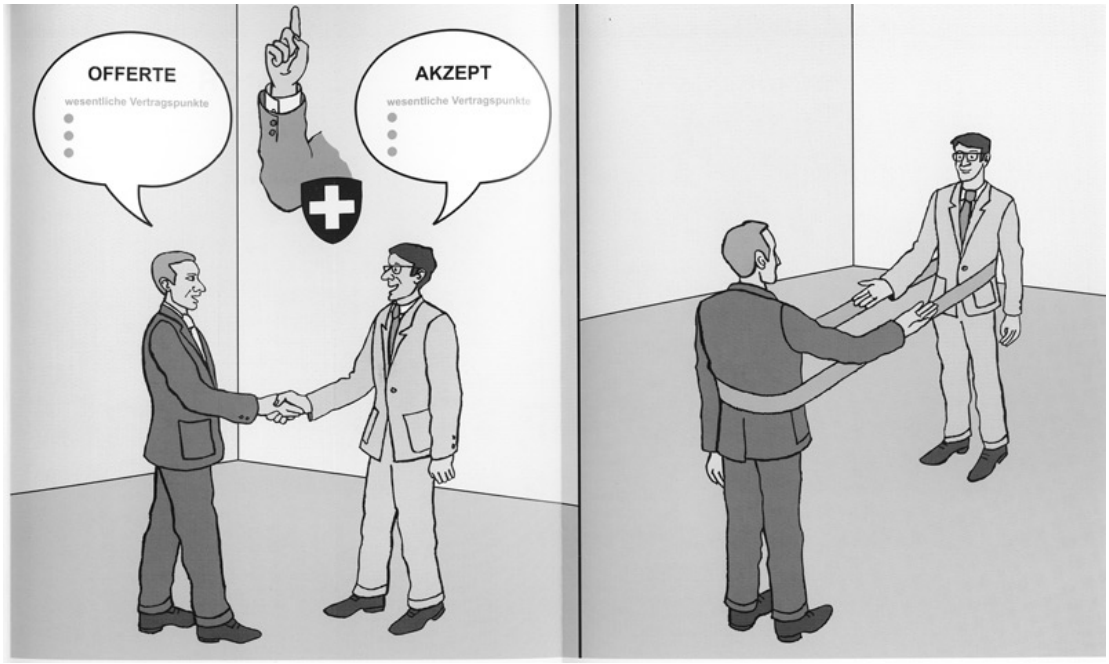


**Figure 2:** Art. 1 Abs. 1 OR: „*Zum Abschluss eines Vertrages ist die gegenseitige übereinstimmende Willenserklärung nötig*" (engl.: A contract is enacted if all parties execute a declaration of intend) (Brunschwig 2001).

Figure 2 includes two pictures that depict the closing of a contract on the left hand side and the consequences of the contract on the right hand side. The two men can be identified as the contractors. On

the left picture the two contractors move towards each other and agreeing on a contract which is symbolised by the handshake. The speech bubbles communicate the main points of the contract as a bullet list (articulation of contractors' will – i.e. offer and demand). The hand over the Swiss flag belongs to the legislative power. It symbolises the contract to be made under current Swiss law. On the right picture both contractors are bound to each other representing the demands and obligations that result from the contract. Their open and extended hands symbolise the obligation on the contract to be profitable for both parties.

The *Sachsenspiegel* was developed at a time when – to the best of our knowledge – no concepts for the modelling of information and processes existed. It was rather a substitute for the written word for illiterates than an actual model, i.e. a representation of a real world phenomenon in order to analyse it or to use it as a description of an observation or a plan. In the following sections, we investigate the question of whether from a contemporary perspective modelling legal circumstances has an additional value and which kind of modelling approaches should be chosen for this. The following section begins with general comments on the term model and the act of modelling.

## 4. Formalisation and modelling

In order to find an appropriate modelling language for the description of legal regulations, it makes sense to have a general view of what a model is and to draw conclusions from these general considerations.

In accordance with Stachowiak (1973, pp. 131-133), models typically fulfil the following properties:

- *Reproduction*: models describe something and represent natural or artificial originals which themselves might be models.
- *Reduction*: models typically describe selected attributes of the represented original which are relevant from the modeller's perspective.
- *Pragmatic*: models are used as a substitute:
    - For subjects (agents) which understand and use the models,
    - during a specific time period and
    - under the restriction of notional or effective operations.

Applied to the modelling of legal regulations, models should especially help to understand the interplay of the single paragraphs on a process level. A systematic approach to represent this should then support the development of process fragments for the isolated paragraphs and the integration of these models to the whole picture after the event. Consequently, any process modelling language for the representation of legal regulations must support process integration.

Integration is, however, also an important operation for a second reason: The development of formal models of legal constraints is not the ultimate goal. Rather it makes sense to provide models which enable business analysts to develop and control business processes in a legally correct manner. This, however, has an immediate impact on the actual modelling task. Any kinds of models which are developed to describe legal constraints must be formulated in notations which support their automatic integration within other existing models. Unfortunately, in computer science and information systems the formal integration of models is still an open research question and only some work has so far been conducted for data and process models.

In the integration of data models, there is work reported on *view* and *schema integration of data models*. Batini et al. (1986) provide a comparative analysis of schema integration methodologies. They structure the schema integration into the activities *pre-integration*, *comparing*, *conforming*, *merging*, and *restructuring*. Several contributions focus on specific activities. Rahm and Bernstein (2001) present a survey on how matches across different schemas can be identified automatically. Rizopoulos and McBrien (2005) discuss a support merge operation. Schmitt and Saake (2005) provide a comprehensive integration method.

Compared with the integration of data models, even less work has been conducted on the integration of process models. Most of these contributions offer integration procedures on Petri net process models (Simon 2006) with only a few also covering generic aspects (Grossmann et. al, 2005). Olbrich and Simon (2006) report on the modelling of government processes and use the integration approach to identify best practices among a set of different processes (see also (Simon, and Olbrich 2007)). All these approaches on process integration have in common that they also visualise the described processes. The integration of conceptual process models (given as Event-driven Process Chains, EPC) which are also visual models was lately

 

described by Mendling and Simon (2006, 2007) under the assumption of a given specific semantics. Although this approach is of interest, if business process models are extended by legal constraints as reported by Alpar and Olbrich (2004), it does not seem to be appropriate here, since the resulting models leave room for contradicting interpretation. Finally process algebras support integration, however, no proper visualisation.

This observation induced Simon (2007) to consider Workflow nets (WfNet) (Aalst 1998), Event-driven Process Chains (EPC) (used in the context of e-Government in (Scheer, Kruppke, Heib 2003)), Flow-chart diagram (FC) (DIN 66001, 1983), UML Activity diagrams (UML) (Rumbaugh, Jacobson, Booch 1999), Process algebra (PA) (a comprehensive overview is given by Fokkink (2000)), and the Business Process Execution Language for Web Services (BPEL) (Andrews et al. 2003) as typical representatives of process modelling languages in computer science concerning the criteria whether the resulting models are explanatory, visualisable, verifiable, executable, integrative (concerning different views and subjective interpretation), and scalable. These criteria have been chosen since the first two are necessary for the development of models that help to understand processes while the others are of importance if these models need to be integrated and executed. The result is shown in Table 1.

**Table 1:** Evaluation of existing process modelling languages

| | **WfNet** | **EPC** | **FC** | **UML** | **PA** | **BPEL** |
|---|---|---|---|---|---|---|
| **Explanatory** | – | – | – | – | √ | – |
| **Visualisable** | √ | √ | √ | √ | – | √ |
| **Verifiable** | √ | – | – | – | √ | – |
| **Executable** | √ | – | – | – | √ | √ |
| **Integrative (views)** | √ | √ | – | – | – | – |
| **Integrative (subjective interpretation)** | √ | – | – | – | √ | – |

All graphical modelling languages (WfNet, EPC, FC, UML, and BPEL) support visualisation, purely mathematical models do not. Formal (mathematical) models provide means for verification and execution; informal models cannot be used for execution and verification by definition. The integration of models has only been defined for specific modelling languages as shown in the table. Although formulas of the process algebra are relatively hard to read for non-mathematics, due to the possibility to integrate models they can be developed stepwise which supports the augmentation of explanations to the models. For the same reasons, process algebras rather than other modelling languages support scalability.

Unfortunately, none of the considered languages do sufficiently fulfil all requirements of process languages for the description of legal regulations. The Semantic Process Language (SPL) overcomes these limitations by combining a language for the formal definition of processes with their visualisation as Petri nets (Petri 1962; Reisig 1991), or to be me more precise, a variation of Workflow nets. The meaning of SPL words (which are called modules) is not defined by a grammar but by canonical building rules which take modules as input and synthesise a special class of Petri nets (called Module nets). Specific firing sequences of these nets are then interpreted as the (sequential) processes defined by a module. Figure 3 shows the relationship between these concepts.

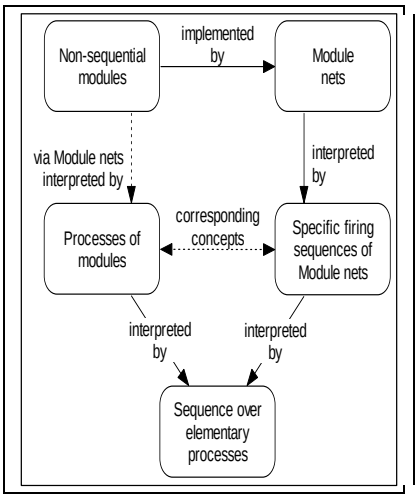


**Figure 3:** Use case for application processes

The 'words' of SPL are built over elementary activities which occur or are explicitly forbidden. Especially the possibility to describe prohibitions is of major importance for modelling legal constraints and therefore e-Government processes (Simon and Olbrich 2007). If *a* and *b* are activities, then processes can be built as a sequence of *a* and *b* (*a* before *b* is specified by $a<b$, *a* after *b* is specified by $a>b$ which has the same interpretation as $b<a$). An alternative between *a* and *b* is specified by $a \vee b$, and the concurrent execution of *a* and *b* by $a \wedge b$. Finally, it is also possible to define various kinds of iteration. Like elementary activities, the operators can also be applied to already defined modules. Hence, complex process structures can be recursively defined.

The semantics of modules are defined by canonical building rules which take the modules as input and generate module nets out of them. Figure 4 shows the implementations for the above mentioned examples. As can be seen with the examples, module nets are Petri nets with explicit *start* and *goal* transitions. The process structure of the modules is defined between these borders. A process of a module net is a firing sequence which reproduces the empty initial marking by firing *start* and *goal* transition exactly once. All of these processes of a module net implementation of an SPL word (i.e. of a module) define the semantics of this module.

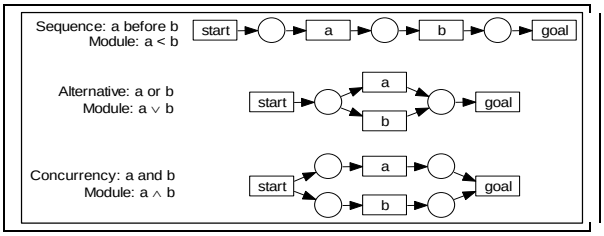


**Figure 4**: Exemplary modules and their implementation in Module nets

The concurrency operator of SPL plays an important role. If the operands *m1* and *m2* of a module $m1 \wedge m2$ do not share any activity, the processes defined by these operands can be executed independently of each other. Otherwise, two situations have to be distinguished:

1. If *m1* and *m2* share the same activity which occurs in both modules, then their processes need to be synchronised within this action.
2. If *m1* and *m2* share the same activity which occurs in one of the modules and is forbidden in the other, then the parallel execution and prohibition must be ruled out.

We do not show, from a Petri net perspective, the relatively simple implementation here but want to discuss the consequences of this operator. The result of its application leads in principle to the intersection of the process sets of *m1* and *m2*. If we now assume that *m1* is a business process and *m2* specifies a legal constraint, then *m1* partially contradicts *m2*, if at least one process of *m1* is lost due to this operation. The compliance of a business process specification can therefore simply be verified this way.

 

The following section takes the obligation right of the Switzerland (see chapter 3) as an example and considers whether process-like structures can be observed in modern laws and whether these laws can be formalised using the approach of SPL. It also makes use of a time extension of SPL (Simon 2002).

## 5. Standard process according to the Swiss obligation law

The Swiss obligation law is derived from Roman law (similar coding can be found throughout continental Europe) and belongs to the private law. Even though the obligation law is 30 years older (1881) than the private law it can be found in the 5th book of private law of Switzerland (ZGB). In private law, the protection of goods and property is guaranteed by the possibility to demand amends. The threat of having to make amends can also be interpreted as a preventive action; e.g. not to violate a contract, protect the environment etc. The private law distinguishes between absolute laws and relative laws. First, protect the individual from all other citizens and institutions (e.g. property rights, data protection). The latter describes demands to another person. The other person has the obligation to fulfil these demands. Bonds and demands are mostly derived from contracts (Art. 1-40 OR) which – implicitly or explicitly – regulate all exchanges of money, goods and services in free market organised societies.

A contract is made if the parties execute a respective declaration of intent (Art 1 OR). If we assume that the contract is made by two parties, then this can be modelled as a concurrent process where each party declares her/his intent explicitly or tacitly. The following SPL module specifies this behaviour.

*Art 1 := ((Explicit intent declaration first party)* $\vee$ *(Tacit intent declaration first party))* $\wedge$
*((Explicit intent declaration second party)* $\vee$ *(Tacit intent declaration second party))*

The assumption of acting in good faith (Art. 2 ZGB) means that the contract is valid if there is an agreement on the main subjects. Hence, any additional condition must not interfere with the main conditions of the contract – in case of doubt, a judge must decide on additional conditions (Art 2 II OR).

Since we focus on making a contract and Article 2 deals with negotiation of additional aspects after the event, modelling this aspect formally is beyond the scope of this paper. It would, however, be an important element if the full set of laws was to be formalised. In this case, Art 2 OR would then be a link between finding a contract and fulfilling it.

Normally the contracting process is started with an offer which might only be valid for a certain period of time (Art 3 OR). The offer must be formulated in a way that one can simply accept or reject it (by a *yes* or *no* reply). In SPL this is formulated by the following module. The intervals in front of the activities of the alternative describe that *First party accepts reaction* must occur within the time limit while the offer is retracted upon the time limit expiring.

*Art 3 := First party makes offer <*
*(*$_{[0;\ time\ limit]}$ *First party accepts reaction* $\vee$ $_{[time\ limit+1;\ \infty]}$ *First party retracts offer)*

If no time limit is set the contract must be accepted immediately (i.e. while they stay together or during telephone contact), as the contractor is not bound to his offer upon the expiration of the time-limit (Art 4 I OR).

*Art 4 := First party makes offer <*
*(* $_{[0;\ time\ staying\ together]}$ *First party accepts reaction* $\vee$
*(Second party does not react in time < First party retracts offer) )*

Article 5 deals with the written form. If no regulations on respite are taken the legal time period is assumed. The time period starts with the arrival of the offer. In case the response arrives beyond the time period and the offer should no longer be valid, one must declare the late arrival (Art 5 III OR) also in written form.

*Art 5 := First party sends offer <*
*((*$_{[0;\ typical\ reaction\ time]}$ *Second party sends reaction < First party accepts reaction)* $\vee$
*(*$_{[typical\ reaction\ time+1;\ \infty]}$ *Second party sends reaction <*
*(First party accepts reaction* $\vee$
*(First party retracts offer < First party sends retraction information)*
*)*
*))* $\vee$
*(Second party does not react in time < First party retracts offer)*

Article 6a was recently added as a matter of customer protection against aggressive distribution. It dictated that one is not responsible for items sent without explicit agreement, except in the case of a clear error.

*Art 6a := First party sends good <*
*(Second party ignores offer ∨*
*Second party accepts offer) ∨*
*(Second party recognises mistake < Second party informs first party)*
*)*

The first article describes the last step of initiating an obligation in a precise way. It states that both parties (if there are two) have to explicitly or implicitly act within this phase. Articles 3, 4, 5, and 6a describe an explicit activity of the first party (*first party makes offer, first party sends offer, or first party sends good*). The behaviour of the party is partially explicitly given and the second party is mentioned (*second party ignores offer, second party accepts offer, second party sends reaction*). We also find examples that the (explicit) behaviour of the second party can be derived from the described behaviour of the first party (*first party accepts reaction*). Beside this, the obligation right describes in Article 6 (not further addressed here) implicit behaviour of the second party. Finally, Article 5 also describes a situation where the second party does not react at all (or at least not in the proposed time). In compliance with the first article, this situation shows that no contract is established.

All discussed articles can be linked as follows:

*Init Obligation := Art 3 ∨ Art 4 ∨ Art 5 ∨ Art 6a*

Figure 5 shows the implementation of *Init Obligation* as a module net. For improved readability, it uses the abbreviations of Table 2.

**Table 2:** Abbreviations of the process model

| Abbreviation | Activity |
|---|---|
| FmO | First party makes offer |
| FsO | First party sends offer |
| FsG | First party sends good |
| FaR | First party accepts reaction |
| FrO | First party retracts offer |
| FsI | First party sends retraction information |
| SsR | Second party sends reaction |
| SaO | Second party accepts offer |
| SiO | Second party ignores offer |
| notSrIT | Second party does not react in time |
| SrM | Second party recognises mistake |
| SiF | Second party informs first party |

An example would be a telephone company offering reduced telephone and internet rates on a promotional basis for new contractors within a certain period of time (e.g. “sign up by the end of the month and you will only pay…”). Contractors usually make the first contact by phone, fax or email. Since telephone contracts contain standing orders and other legal consequences, the telecommunication law of most countries demands a written form. Hence, after the first contact is made, the written and mostly very distinct contracts are sent – surprisingly often already with the hardware. If the contractor does not send the signed documents back to the company, the contractor is not bound to the contract. In contrast, if the documents arrive after the promotion ends, the telephone company is not bound to their special offer and could charge the standard fee. In this case they must notify the contractor of the late arrival.

There remains the question of what happens to the hardware, if the documents are not signed. Even after the signed contract is received by the company it normally takes a certain amount of time until the new line opens. Hence, there would be plenty of time for the phone company to send the requested hardware later. If the company, however, sends the hardware together with the (unsigned) contract, Art 6a comes into play. Thus, if the contract is not signed, sending hardware is regarded as aggressive distribution. The contractor has consequently no responsibility to return the hardware or to inform the phone company about the hardware.

 

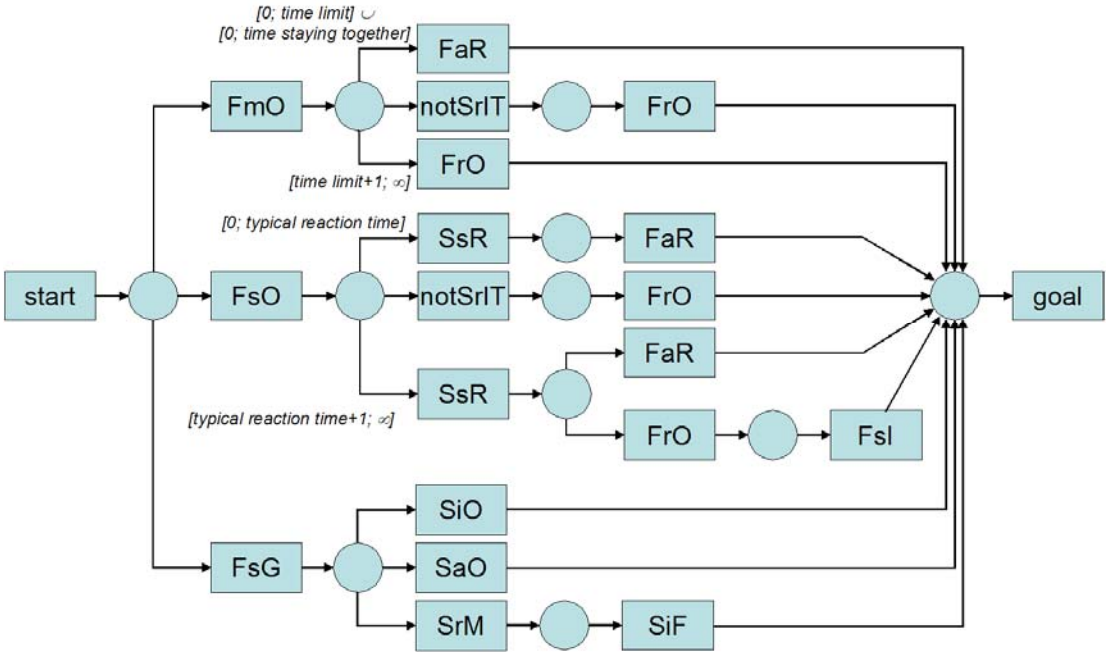


**Figure 5**: Module net of selected articles (*Init Obligation*)

Figure 5 illustrates the basic paragraphs of the Swiss OR in a formal process model. The upper control flow describes the standard process of contracting. The two parties are together – e.g. in a store or office – and negotiate the offer made. The second control flow is the procedure of the telephone company that sends the contract that requires signature. We assume, however, that the contract is being accepted by both parties according to Article 2 OR. The latter control flow depicts our example which is regulated by the new Article 6a OR: if the phone company sends hardware without waiting for the signed contract the potential contractor is not responsible for returning the hardware.

This model is, of course, far from complete. For example, it is argued in the legal theories when an offer is accepted; and henceforth becomes a contract. Normally one has to accept an offer explicitly (Art 6 OR). There are, however exemptions if an explicit agreement is not to be expected (gifts, standing orders etc.). We simply assume the offers to have been explicitly accepted. Our analysis of Article 1-6a describes the basic process of contracting. The following Articles (7-40) deal with further circumstances (indistinct offers - *invitatio ad offerendum,* treatment of minor or non-judicious persons etc.) and are therefore ignored. Nevertheless, they describe further process sets which can be addresses in a similar way.

## 6. Conclusion and outlook

A recent survey identified the strict legal framework regulating of the public processes as one of the major obstacles to introduce an efficient e-Government program (Scheer et. al. 2003). It mostly delays ambitious programs because expert opinion is needed frequently. Delivering expert opinion needs to involve further specialists, since the information about the legal constraints is mostly limited to specialised and experienced lawyers. Regulations and prohibitions, however, can be found in almost every aspect of our private and business lives. An overall comprehension of the code of law is, therefore, not limited to trained personnel. This problem seems to be emphasised in the current e-Government programs, since almost any public process is defined in public law. Hence, to introduce an efficient e-Government system, the information about the legal constraints is needed quickly and in any part of the development.

We therefore suggest documenting the knowledge about the legal constraints within the process model itself. Programmers and business analysts can then implement or improve the process further without any legal consultation or the danger of violating given law. Hence, the models should be more resistant to a change in the legal framework. If the legal framework changes, one only needs to change the model, which can then be compared with the implementation. Such an approach might also tell which laws are barriers for an efficient implementation and which laws might need to be changes in order to implement e-Government at all (e.g. change the law to accept electronic documents as legal documents, etc.).

In order to proof our thesis that many laws directly implicate executable process sets, we derived a formal - and therefore verifiable - model directly from the code of law. We used an existing graphical and textual illustration of the Swiss obligation law as our basis. With the help of the Semantic Process Language, that uses common statements as semantic input, we were able to translate the illustration into an executable Module Net. This Module Net represents the legal procedure and can therefore be considered as a specification of the given law. Hence, we can use this model to verify implemented procedures against the code of law.

While on the first view the integration of legal processes into business process models seems to be a problem only of public administrations (Snellen and Zuurmond 1997), we can observe a comparable trend for companies as well (Knackstedt et. al. 2006). In order to adapt the living and working conditions in the European Union, the number of laws and regulations for processes of private enterprises increases particularly in human resource management, industrial safety and product and service quality assurance. They enforce to extend existing business process modelling methods by means for the representation of legal restrictions of tasks and (partial) processes.

Applying Module Nets as a formal description of public workflows appears to be a useful improvement for several ongoing projects. One of these is the E-Justice concept of the European Union which is part of the 6th Research Framework Program funded by the action plan for eEurope 2005 by the European Commission (European Commission 2006). The E-Justice concept tries to document standard legal procedures in the European Union as a kind of reference processes. The goal to document legal processes in Business models to make them transparent and comprehensible. Cross border selling portals in the European Union can use such models to document the necessary steps to be taken, if e.g. a contractor from a different European member state does not fulfil his obligation (e.g. a customer does not pay, a vendor does not deliver etc.).

Next, we plan to enhance the user-interface management for public services (Freiheit and Zangl 2006) by our method of deriving process models directly from the legal constraints. The goal is to improve European markets by deriving process models automatically that already include the legal constraints of each country. This could bring us closer to unified and transparent process sets within the European Union.

## References


Aalst, W. M. P., van der (1998): *The Application of Petri nets to Workflow Management*, The Journal of Circuits, Systems and Computers.

Alpar P.; Olbrich S. (2004): *Process modeling for e-Government*. In D. Remenyi, (Ed), Conference proceedings of the 4th European Conference on Electronic Government (ECEG): pp 35-44, Dublin, GB.

Andrews, T.; Curbera, F.; Dholakia, H.; Goland, Y.; Klein, J.; Leymann, F.; Liu, K.; Roller, D.; Smith, D.; Thatte, S.; Trickovic, I.; Weerawarana, S. (2003): *Business Process Execution Language for Web Services – Version 1.1*. ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf.

Batini, C.; Lenzerine, M.; Navathe, S. B. (1986): *A Comparative Analysis of Methodologies for Database Schema Integration*. ACM Computing Surveys, 18(4), pp. 323-364.

Boehme-Neßler V. (2005): *Visualisierung des Rechts - Chancen und Risiken. Rechtstheoretische Anmerkungen zum Einfluss der Bilder auf das Recht*. in Eric Hilgendorf (Ed.): Beiträge zur Rechtsvisualisierung, Logos Verlag, Berlin.

Brunschwig, C. (2001): *Visualisierung von Rechtsnormen - Legal Design*. Fögen/Schott/Senn/Weimar (Eds..): Züricher Studien zur Rechtsgeschichte, Band 45, Schulthess.

DIN 66001 (1983): *Informationsverarbeitung: Sinnbilder und ihre Anwendung*.

European Commission (2006): *eEurope 2005 - An information society for all.* http://europa.eu.int/information_society/eeurope/2005/index_en.htm (06-07-06).

Fokkink, W. (2000): *Introduction to Process Algebra.* Springer, Berlin.

Freiheit J.; Zangl A. (2005): Model-*based User-Interface Management for Public Services*. In: D. Remenyi (editor): Conference proceedings of the 6th European Conference on Electronic Government (ECEG), Reading, GB 2006, S. 141-151.

Grossmann, G.; Ren, Y.; Schrefl, M.; Stumptner, M. (2005b): *Definition of Business Process Integration Operators for Generalization*. In 7th International Conference on Enterprise Information Systems, ICEIS 2005, May 24-28 2005.

Hevner, A.R., March, T.S., Park, J., und Sudha, R. (2004): *Design Science in Information Systems Research*, MIS Quarterly (28:1), S. 75-105.

Knackstedt, R.; Brelage, C.; Kaufmann, N. (2005): *Entwicklung rechtsicherer Web-Anwendungen – Strukturierungsansatz, State-of-the-Art und ausgewählte Aspekte der fachkonzeptionellen Modellierung*. In: WI 48 (2006) 1, S.27-35.

Kreuzbauer, G. (2005): *Visualisierung juristischer Argumentation*. in Eric Hilgendorf (Ed): Beiträge zur Rechtsvisualisierung, Logos Verlag, Berlin.

Laufs, A. (1996): *Rechtsentwicklungen in Deutschland*. de Gruyter, Berlin.


 

March, T.S. und Smith, G. (1995): *Design and Natural Science Research on Information Tech-nology*, Decision Support Systems (15:4), S. 251-266.
Olbrich, S.; Simon, C. (2006): *Process engineering towards e-Government – modelling process variety and best practice*. in D. Remenyi (Eds.): Proceedings of the 6th European Conference on e-Government (ECEG 2006), April 2006, Marburg, Deutschland, Academic Conferences Limited, Reading, UK, ISBN 1-905305-19-2cd, pp. 321-331.
Oppitz, U.-D. (1990): *Bechreibung der Handschriften*. Deutsche Rechtsbücher des Mittelalters, Band 1.
OR (1911): *Bundesgesetz betreffend des Schweizerischen Zivilgesetzbuches (Fünfter Teil: Obligationenrecht)* vom 30. März 1911 (Stand am 4. Juli 2006).
Petri, C. A. (1962): *Kommunikation mit Automaten*, Schriften des Institutes für instrumentelle Mathematik, Bonn.
Pettersson, R. (1993): *Visual Information*. Educational Technology Publications, Englewood Cliffs, 2. Edition, New Jersey. pp. 135ff.
Rahm, E.; Bernstein, P. A. (2001): *A Survey of Approaches to Automatic Schema Matching*. VLDB Journal, 10(4), pp. 334-350.
Reisig, W. (1991): *Petrinetze: eine Einführung*, 2. Auflage, Springer-Verlag, Berlin.
Rizopoulos, N.; McBrien, P. (2005): *A General Approach to the Generation of Conceptual Model Transformations*. In: Pastor, O.; e Cunha, J. F. (Edt.): Proceedings: Advanced Information Systems Engineering. 17th International Conference CAiSE 2005, Lecture Notes in Computer Science (LNCS), 3520, Porto, Portugal, Springer, S. 326-341.
Rumbaugh, J.; Jacobson, I.; Booch, G. (1999): *The Unified Modeling Language Reference Manual*. Addison-Wesley, Reading, MA.
Sachs-Hombach, K. (2005): *Vom Text zum Bild - Wege für das Recht*. in Eric Hilgendorf (Ed.): Beiträge zur Rechtsvisualisierung, Logos Verlag, Berlin.
Scheer, A.-W.; Kruppke, H.; Heib, R. (2003): *e-Government – Prozessoptimierung in der öffentlichen Verwaltung*, Springer Verlag, Berlin Heidelberg.
Schmitt, I.; Saake, G. (2005): *A Comprehensive Database Schema Integration Method Based on the Theory of Formal Concepts*. Acta Informatica, 41(7-8), pp. 475-524
Schott, C. (1986): *Sachsenspiegel und Bibla Pauperum*, in: Ruth Schmidt-Wiegand (Ed.): Bild-Text Interpretation - Untersuchungen zu den Bildhandschriften des Sachsenspiegels, 1. Textband, pp. 34-43.
Simon, C. (2002): *A Logic of Actions to Specify and Verify Process Requirements*. in: The Seventh Australian Workshop on Requirements Engineering (AWRE'2002), December 2002, Melbourne, Australia, 12 pages.
Simon, C. (2006): *Integration of Planning and Production Processes*. In: Mathmod 2006, Special Session Petrinets: Current Research Topics and their Application in Traffic Safety and Automation Engineering, Wien, Austria.
Simon, C.; Olbrich, S. (2005): *The Influence of Legal Constraints on Business Process Modelling*, In: e-Government Workshop 2005 (eGOV05), London.
Simon, C.; Olbrich, S. (2007): *Integration Of Legal Constraints Into Business Process Models*, in: Z. Irani (Hrsg.): Transforming Government: People, Process and Policy, Volume 2, Emerald, Bradford, GB.
Snellen A. and Zuurmond I. (1997): *From Bureacracy to Infocracy: Management through information architecture*. In Z. Tyler, Snellen, editor, Beyond BPR in Public Administration - Institutional Transformation in an Information Age, Amsterdam IOS Press, pages 205-224.
Stachowiak, H. (1973): *Allgemeine Modelltheorie*, Springer, Wien, New York.
Ulbrich, S. (2000): *Visuelle Rechtskommunikation*. in Alfons Brora (ed: Zeitschrift für Rechtssoziologie, Band 21, Heft 2, Lucius + Lucius, Stuttgart.
Universitätsbibliothek Heidelberg (2006): *Heidelberger Bilderhandschrift des Sachenspiegels*, [online], http://digi.ub.uni-heidelberg.de.
Wegschneider, H. (2005): *Visualisierung in der juristischen Lehre*. In Eric Hilgendorf (Ed.): Beiträge zur Rechtsvisualisierung, Logos Verlag, Berlin.

 