Workpackage 32
Regulatory Framework

Deliverable 32.6 (Final)
Model for Generic Level DBE Contracts and Agreements

Project funded by The European Community under the “Information Society Technology” FP6 Programme

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

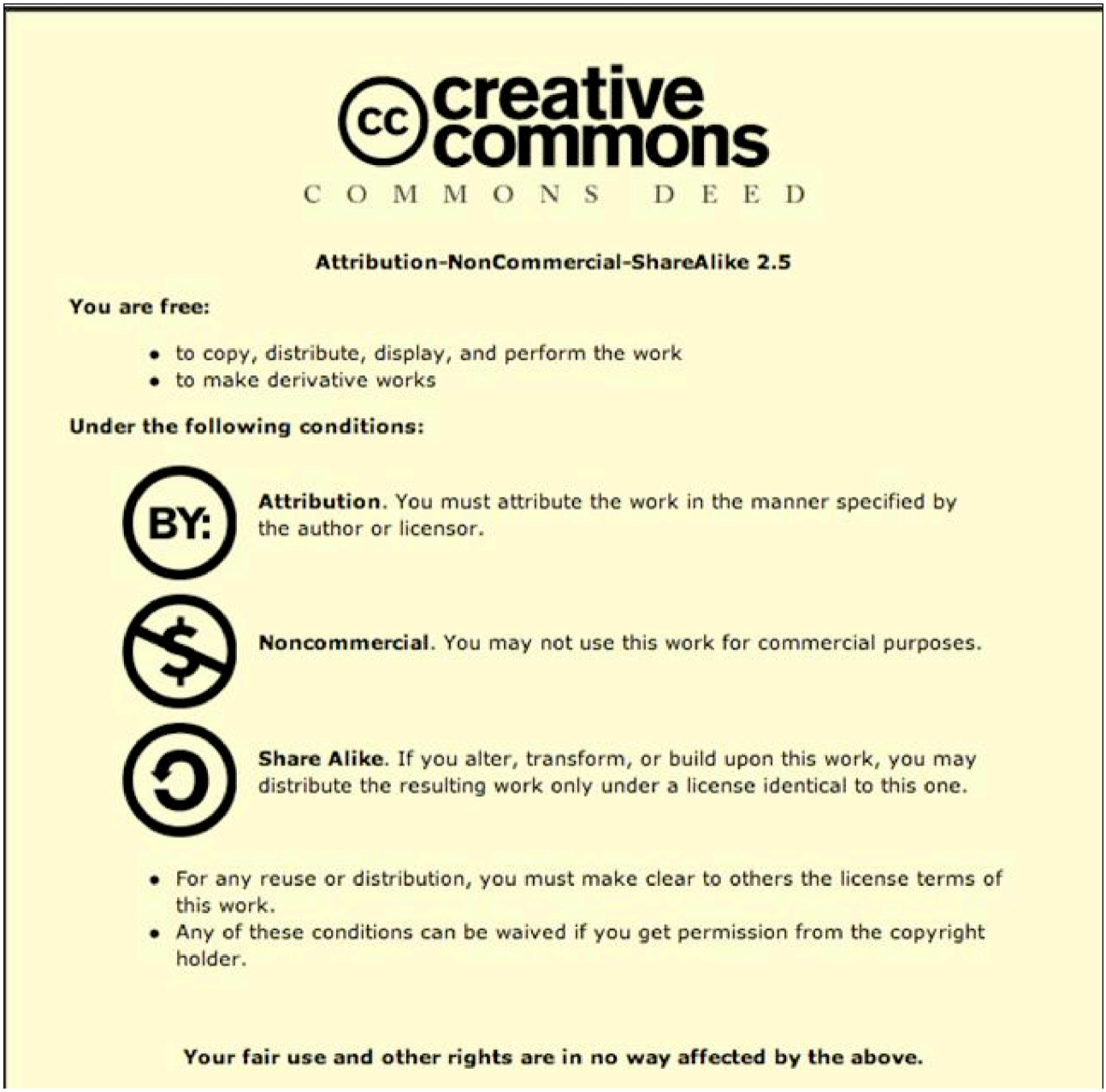
Deliverable Number: Model for Generic Level DBE Contracts and Agreements – D32.6
Due Date: 31/07/2006
Delivery Date: 31/10/2006

Short Description: The purpose of this document is to provide models for generic contracts and agreements in DBE. The models are based on the output of research into open standards for contract and legal document modelling (see D32.3). It is the intention to generically model contracts in a DBE independent manner using regular UML tools. The output of this modelling is represented in this document via a schema visualisation. This forms the basis of the DBE generic level contract which is created using the available languages and tools.

Author: Waterford Institute of Technology
Partners Contributing: Waterford Institute of Technology
Made available to: Public

Versioning		
Version	**Date**	**Author, Organisation**
0.1	03/01/05	Paul Malone, Jason Finnegan, Waterford Institute of Technology
1.0	11/01/06	Incorporated comments from internal review
2.0	03/04/06	Second Draft, Jason Finnegan
2.01	17/05/06	Incorporated comments from internal review
3.01	01/10/06	Final Version

Quality check
1st Reviewer : Pedro Bueso, Unizar
2nd Reviewer : John Kennedy, Intel



creative commons

COMMONS DEED

Attribution-NonCommercial-ShareAlike 2.5

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works

Under the following conditions:

BY: **Attribution**. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

Table of Contents

Table of Figures

1 Introduction

As already discussed in D32.3(An Analysis of Legal ICTs)[1], modelling machine readable contracts that are applicable across global jurisdictions in many business domains and languages is a difficult task which has seen the creation of many standardisation efforts.

However, none of these standardisation efforts have reached a level of maturity to be considered de-facto standards in the field. This is in no small part due to the wide range of variation in practice and regulation across the global jurisdictions participating in e-commerce. Another important reason why standardisation is such a difficult procedure is the diversity of regulations on a sectoral basis. While no widely accepted model for electronic contracts has been reached, most of the initiatives examined provide recommendations and guidelines for what should be encapsulated in such models. These guidelines have been used for the basis of the modelling contained in this document.

2 XML Schema Contract Data Model

The model for a generic contract has been created using the XML schema specification, in order for the maximum level of interoperability, and for compatibility with existing modelling and development tools. This model drew on the knowledge gained from the review of standardisation efforts and the recommendations arising from those initiatives. During this modelling, it was considered important that the model not only reflect the needs of the DBE users but was also applicable at a more generic e-Commerce level.

Below is a discussion of the various elements of this contract model, starting with the base element (*Contract*) and detailing each attribute and child element in turn.

2.1 Contract

This top level element represents the contract at its most basic level. The *Contract* element comprises an attribute set and sub-elements representing a list of participant legal entities (*PartyList*) and a list of commitments those parties agree to (*ClauseList*) and a set of parameters to which the *AgreementList* may refer (for example, price, quantity, start date, end date, etc.). The *Semantics* section can contain a block of XML code that represent the commitments made in this contract, in a machine readable format. Each of these sub-elements are discussed below.

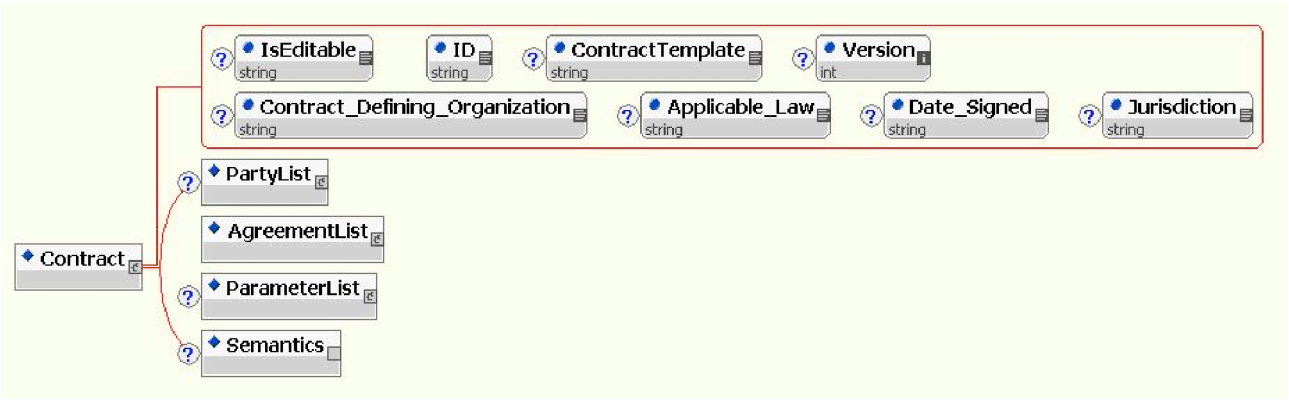


Figure 1: Contract Element

Attributes

IsEditable: indicates whether the associated element (in this case Contract) can be edited by an application.

ID: indicates an unique identifier for this contract instance

Date_Signed: date this contract was signed on

Jurisdiction: indicates the courts that will be responsible to resolve any dispute with regard to this contract

Applicable_Law: The set of laws under which the contract is to be interpreted(e.g. Spanish law)

ContractTemplate: denotes a standardised set of clauses and parameters, which may be specific to a certain business scenario. This template may also be extended by this contract, having extra clauses and parameters. A contract template may also have a corresponding *semantics* section

Version: denotes the current version of the document, incremented during contract negotiation

Contract Defining Organization: denotes the organisation that created this contract or contract template, which may not be a party to the contract.

2.2 PartyList

The *PartyList* element provides a set of participating legal entities(as defined by the jurisdiction of the *Contract*). The number of parties involved can range from zero(in the case of a generic reusable 'Term and Conditions' document) to multiple partners involved a joint agreement.

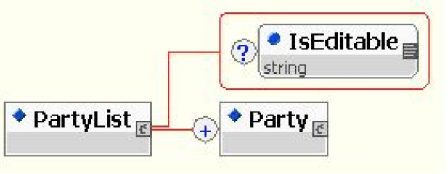


Figure 2: PartyList Element

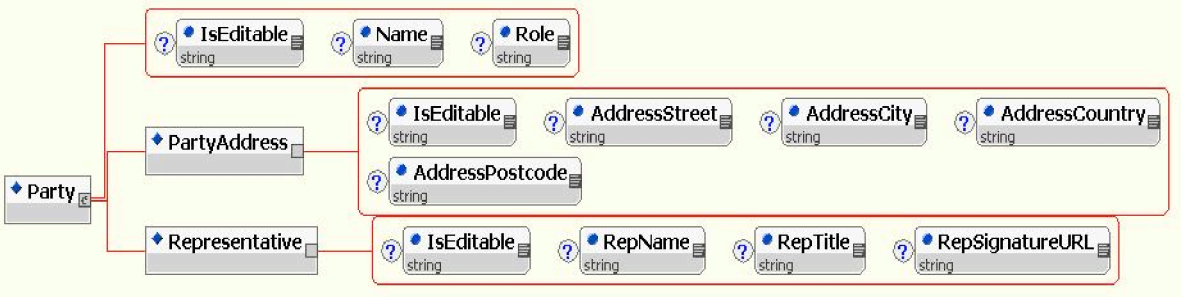
Attributes

IsEditable: indicates whether the associated element (in this case *PartyList*) can be edited by an application.

2.3 Party

The *Party* element represents a single legal entity participant in the contract and contains elements called *PartyAddress* detailing the legal address of the *Party* as well as the optional *Representative* details. The *Representative* provides details of a real person who is acting on behalf of this *Party* in this *Contract*. In the case of a signed contract, this is the name of the person who signs. This is optional as it might not be present in some simple contracts such as *"Terms and Conditions"* type agreements.

Figure 3: Party Element



Attributes

IsEditable: indicates whether the associated element (in this case *Party*) can be edited by an application.

name: indicates the legal name of the *Party*.

role: indicates the role that this *Party* plays in this *Contract*

2.4 AgreementList

The *AgreementList* represents the set of agreements or commitments that the participating parties (via *PartyList*) agree to. *AgreementList* contains one or more *Section* elements.

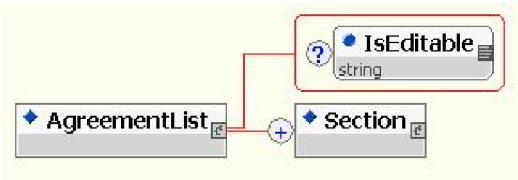


Figure 4: AgreementList Element

Attributes

IsEditable: indicates whether the associated element (in this case *the AgreementList*) can be edited by an application.

2.5 Section

A *Section* is representative of a sub-section of a contract and is itself made of a number of *Clause* elements and *LegalText* elements which refer to the *Section* itself.

Figure 5: Section Element

Attributes

IsEditable: indicates whether the associated element (in this case Section) can be edited by an application.

Title: The name of the section (optional)

ID: A unique identifier of this section.

Include_Optional: a boolean value which if true indicates that a section of the contract is optional,

used only when contract is a template or under negotiation. Default is false for all other situations.

2.6 LegalText

The *LegalText* element represents a section of text (which may be made up from a number of parts) which has a legal meaning as interpreted by the jurisdiction of the contract. It comprises of a number of *LegalTextPart* elements which make up the natural language content of a clause or section.

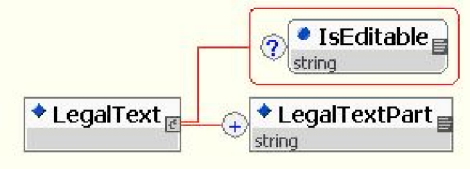


Figure 6: LegalText Element

Attributes

IsEditable: indicates whether the associated element (in this case *LegalText*) can be edited by an application

2.7 LegalTextPart

A *LegalTextPart* can represent either a portion of text, a reference to a parameter in the *ParameterList*, or a reference to to an external resource.

Figure 7: LegalTextPart Element

Attributes

displaytext: for LegalTextParts of type *InternalReference* and *RegulatoryReference*, this is the text that is displayed in the contract

type: describes the type of this part of the Legal text, the possible LegalTextPartTypes are *Text*, *ParameterReference*, *InternalReference*, *ExternalReference*, *RegulatoryReference*

A *LegalTextPart* of type Text contains Text that is to be displayed in the contracting

A *ParameterReference* contains a reference to a parameter in the *ParameterList*, The contents of the parameter is what will be displayed in the contract text.

InternalReference and *RegulatoryReference* both contain an XPath expression, which will not be displayed in a rendered version of the contract. The contents of displaytext will be visible to the user in it place (similar to a html reference)

2.8 Clause

The *Clause* element can contain one *LegalText* element and zero or more *Clause* elements.

Figure 8: Clause Element

Attributes

IsEditable: indicates whether the associated element (in this case *Clause*) can be edited by an application

Title: a title for this clause

ID: a contract wide unique identifier for this clause

Include_Optional: a boolean value which if true indicates that a clause of the contract is optional, used only when contract is a template or before contract has been signed. Default is false for all other situations and if not present.

2.9 ParameterList

A *ParameterList* is made of one or more *Parameter* elements

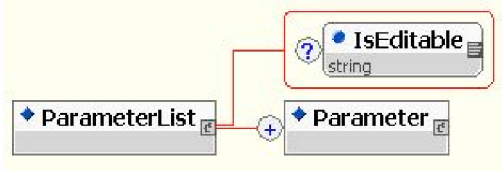


Figure 9: ParameterList Element

Attributes

IsEditable: indicates whether the associated element (in this case *ParameterList*) can be edited by an application.

2.10 Parameter

A *Parameter* is a key-value pair denoting a contract wide static reference. For example to represent a unit price in a contract the *id* might be “Unit-Price” and the contents might be an Integer of “35”(the attribute parametertype would be set to 'Integer'). The currency for this could be

represented by another *Parameter* where the *Key* would be “Unit-Price-Currency” and the *Value* would be “Euro”(the attribute paramtype would be set to 'String') .

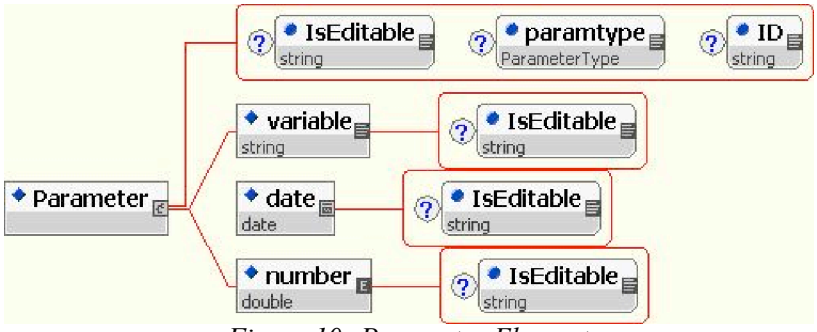


Figure 10: Parameter Element

Attributes

IsEditable: indicates whether the associated element (in this case *Parameter*) can be edited by an application

ID: the unique identifier of this parameters

paramtype: the type of this parameter which can be either a *String*, a *Date* or an *Number*

2.11 Semantics

The optional *Semantics* section of the contract allows the DBE contract model to include extra XML based contract specifications. The *Semantics* section will contain machine readable instructions detailing the various obligations of the parties to the contract. The *Semantics* section of the contract will need to be created in parallel with the rest of the contract, as no machine translation between natural language contract and machine readable is currently possible.

It may be possible for the elements of the *Semantics* section to reference the parameters stored in the *Contract/parameterList*. This would allow certain contract templates to be editable while ensuring both natural language and *Semantics* sections of the contract have the same meaning.

The semantic sections can contain BML[2] or any other XML compatible language, it will not be necessary for Contract Editors and tools to understand the *Semantics* section of the Contract.

The natural language Section of the contract (*ClauseList* and *parameterList*) will be taken to express the true meaning of the contract, the S*emantics* section is a non-binding translation of this agreement into some form of machine readable language.

Figure 11: Parameter Element

Attributes

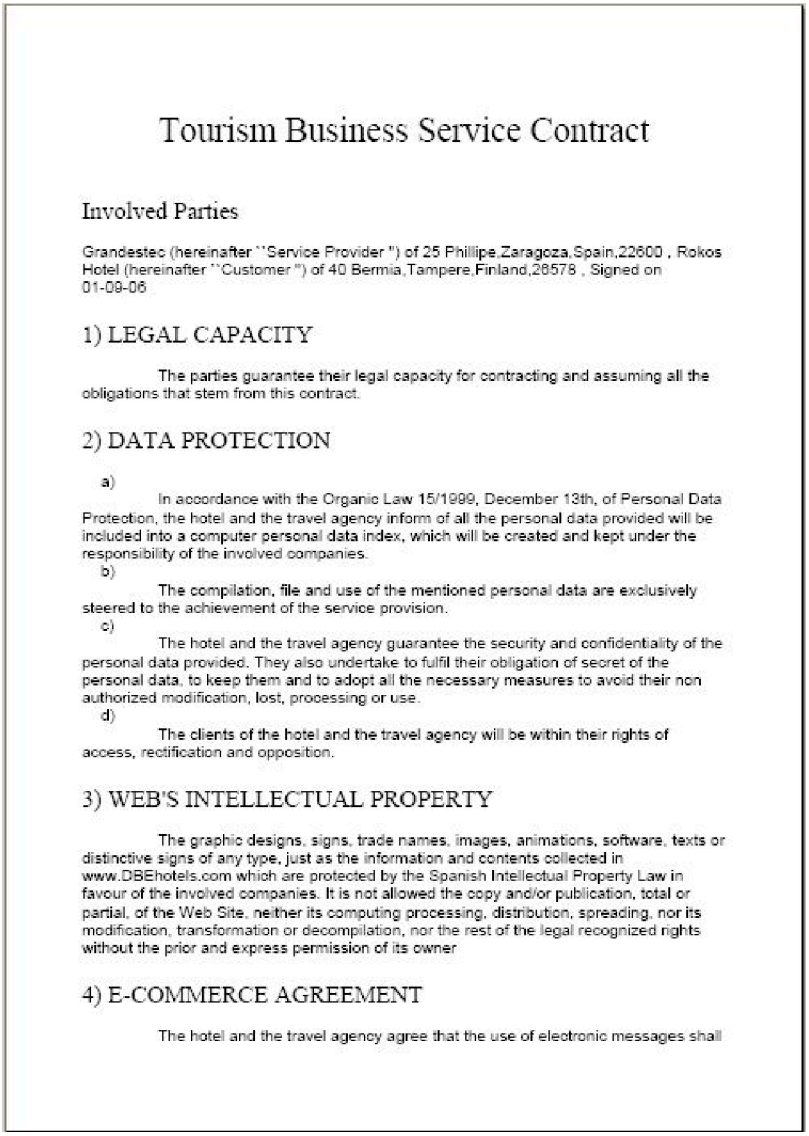
IsEditable: indicates whether the associated element (in this case *Semantics*) can be edited by an application

Semantics_Format: the format of the data held in *Semantics*

2.12 Visualising the Contract

From a user perspective the final contract will appears as follows. This contract was created with a prototype contract creation tool developed at WIT. This particular contract has 2 parties (Smart Cars Stereos and Auto-Music) and their associated roles (Supplier and Consumer respectively) as defined using the tools described above. The XML file has been digitally signed and then rendered to pdf using XSLT. It is expected that in future the visual signatures on the contract can be extracted from the identity service or the profile service. It is important to note that this is a only a visualisation of the contract and the legally agreed contract is is the digitally signed XML contract instance. It is this electronic document that has been digitally signed, that would be under consideration by the courts in the event of a dispute. The transformation from XML contract to the visualisation needs to be standardised in order to ensure consistent interpretation of a contract.

Illustration 12 shows a simplified example of how an XML based contract would be presented to the user.



Tourism Business Service Contract

Involved Parties

Grandestec (hereinafter ``Service Provider '') of 25 Phillipe,Zaragoza,Spain,22600 , Rokos Hotel (hereinafter ``Customer '') of 40 Bermia,Tampere,Finland,28578 , Signed on 01-09-06

1) LEGAL CAPACITY

The parties guarantee their legal capacity for contracting and assuming all the obligations that stem from this contract.

2) DATA PROTECTION

a)

In accordance with the Organic Law 15/1999, December 13th, of Personal Data Protection, the hotel and the travel agency inform of all the personal data provided will be included into a computer personal data index, which will be created and kept under the responsibility of the involved companies.

b)

The compilation, file and use of the mentioned personal data are exclusively steered to the achievement of the service provision.

c)

The hotel and the travel agency guarantee the security and confidentiality of the personal data provided. They also undertake to fulfil their obligation of secret of the personal data, to keep them and to adopt all the necessary measures to avoid their non authorized modification, lost, processing or use.

d)

The clients of the hotel and the travel agency will be within their rights of access, rectification and opposition.

3) WEB'S INTELLECTUAL PROPERTY

The graphic designs, signs, trade names, images, animations, software, texts or distinctive signs of any type, just as the information and contents collected in www.DBEhotels.com which are protected by the Spanish Intellectual Property Law in favour of the involved companies. It is not allowed the copy and/or publication, total or partial, of the Web Site, neither its computing processing, distribution, spreading, nor its modification, transformation or decompilation, nor the rest of the legal recognized rights without the prior and express permission of its owner

4) E-COMMERCE AGREEMENT

The hotel and the travel agency agree that the use of electronic messages shall

Illustration 12: Contract as PDF

2.13 Complete Contract Model (XML Schema)

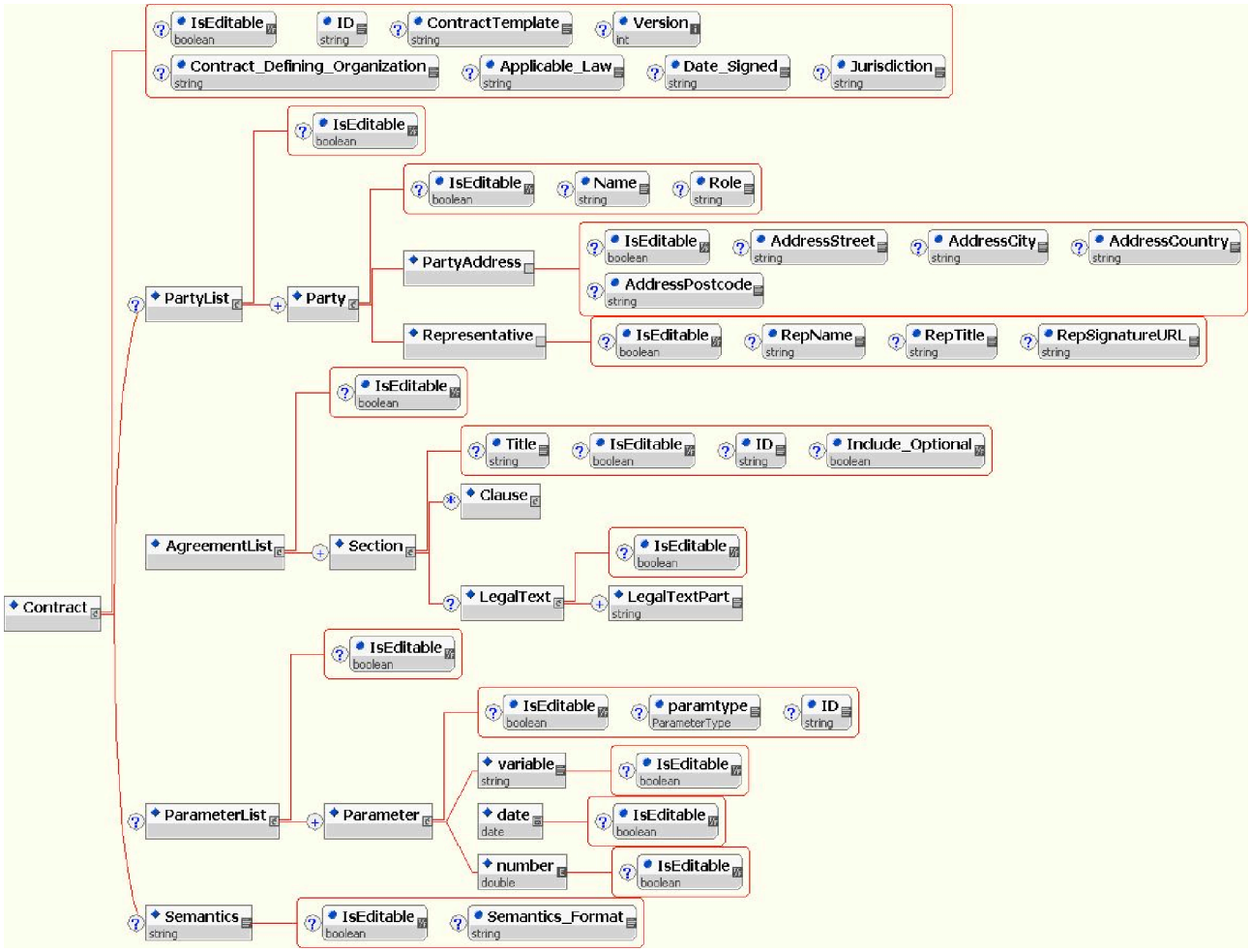


Figure 13: Full Contract Schema Model

The diagram(Figure 13: Full Contract Schema Model) gives a view of the whole contract model as described in this section.

3 Semantic Contract Modeling using BML 1.0

The *Semantics* section of the Contract model described in this document allows the for use of any XML based language to describe the meaning of the contract in a machine verifiable format. This could be a number of pre and post conditions that must be satisfied, technical operations that must be performed etc.

In this section of the document it will be shown how BML can be used to provide such information, using the tools provided in the DBE Studio.

3.1 Creating the Model

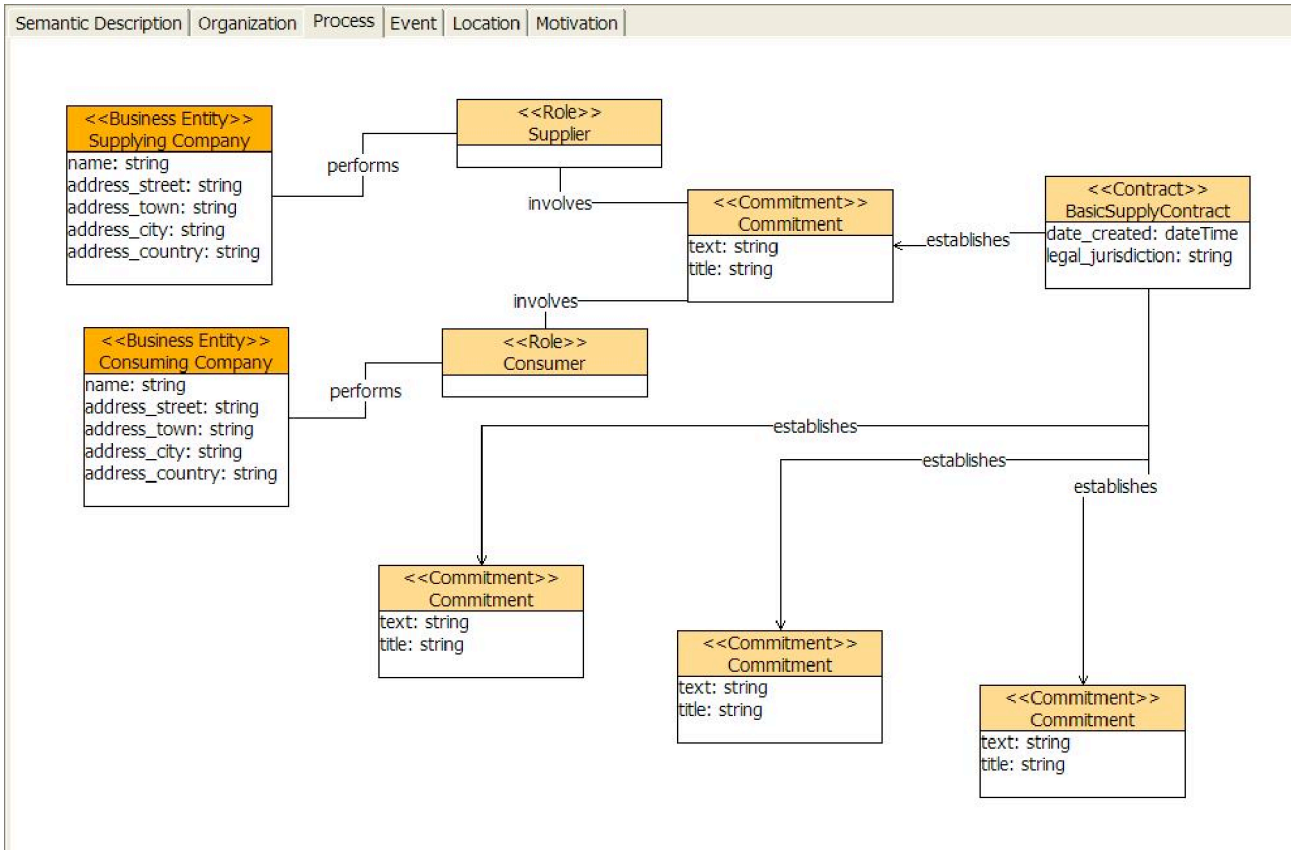


Figure 14: Generic Contract Model using BML 1.0

This M1 model was created using the BML editor of the DBE Studio. The model describes a contract between two business entities, a provider and a consumer. The Contract contains a number of commitments that both parties have agreed upon. This example just shows three commitments(with one required commitment stating that the parties agree to the contract).

3.2 Creating an Instance of the Semantic Contract

The DBE BML Data Editor is used to create an instance of the Semantic contract, and can be used to fill in all or some of the data fields to create a complete Semantic Contract. The output of the BML Data Editor is an XMI file which can be stored on the local file system. This XMI code would

be inserted into an XML contract, in the element called *Semantics.* The attribute *Semantic_Format* would be set to *BML_XMI.* It is planned that contract templates will be stored in the DBE knowledge base and can be accessed by the DBE execution environment when DBE services are executed.

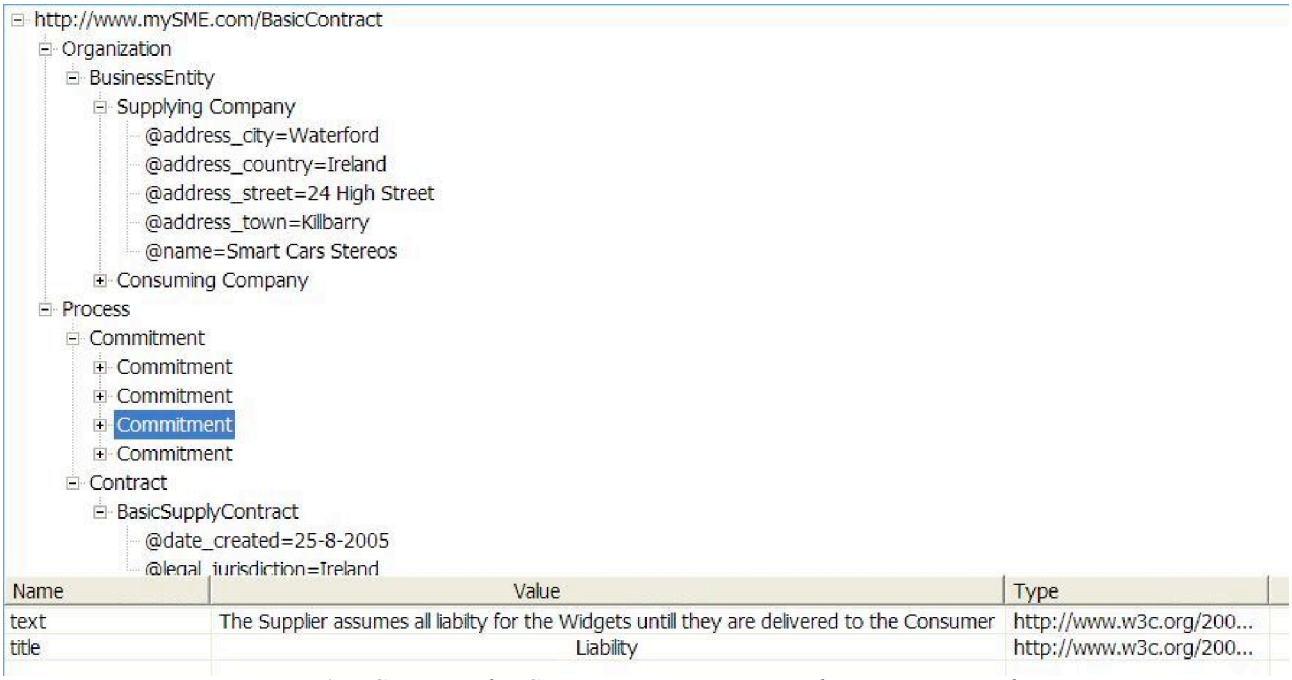


Figure 15: Creating the Contract instance using the BML Data Editor

3.3 Signing the Contract

Once an instance of a contract has been created and all data fields completed the contract can be digitally signed by both parties. The format for digitally signing the contract has not yet been confirmed and is to be integrated with the DBE Decentralised Identity system.

4 Summary and Next Steps

Future work could extend the standardisation of the Semantics section of the contract. Even without the optional Semantics section, the contract model still provides a very useful standard format for contract information.

While simple contract modelling is possible using the available version of BML and its associated tools, it could be argued that the tools are too complex for many non-technical SME business users.

However the work achieved represents a solid basis upon which the task can progress. Currently much of the functionality required to achieve more complex real world business contracts is not available with the DBE studio/execution environment. For example, there are currently no facilities to digitally sign contracts through the execution environment, nor are there authorisation facilities available to ascertain whether a service being executed is covered by a specific contract, or even if the consumer has the authority to do so. This functionality is expected to become available as the distributed identity task progresses.

5 References

[1] Digital Business Ecosystem, Deliverable 32.3, An Analysis of Legal ICTs, Jason Finnegan, Paul Malone, http://www.digital-ecosystem.org/ (checked 1st May 2006)

[2] Digital Business Ecosystem, Deliverable 15.1, Business Modelling Language 1.0, ISUFI, http://www.digital-ecosystem.org/ (checked 1st May 2006)

6 Appendix : Contract Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
        elementFormDefault = "qualified">
        <xsd:attributeGroup name = "Editable" id = "IsEditable">
                <xsd:attribute name = "IsEditable" type = "xsd:boolean"/>
        </xsd:attributeGroup>
        <xsd:attributeGroup name = "Address">
                <xsd:attribute ref = "AddressStreet"/>
                <xsd:attribute ref = "AddressCity"/>
                <xsd:attribute ref = "AddressCountry"/>
                <xsd:attribute ref = "AddressPostcode"/>
        </xsd:attributeGroup>
        <xsd:element name = "Contract">
                <xsd:complexType>
                        <xsd:all>
                                <xsd:element ref = "PartyList" minOccurs = "0"/>
                                <xsd:element ref = "AgreementList"/>
                                <xsd:element ref = "ParameterList" minOccurs = "0"/>
                                <xsd:element ref = "Semantics" minOccurs = "0"/>
                        </xsd:all>
                        <xsd:attributeGroup ref = "Editable"/>
                        <xsd:attribute name = "ID" use = "required" type = "xsd:string"/>
                        <xsd:attribute name = "ContractTemplate" type = "xsd:string"/>
                        <xsd:attribute name = "Version" type = "xsd:int"/>
                        <xsd:attribute name = "Contract_Defining_Organization" type = "xsd:string"/>
                        <xsd:attribute name = "Applicable_Law" type = "xsd:string"/>
                        <xsd:attribute name = "Date_Signed" type = "xsd:string"/>
                        <xsd:attribute name = "Jurisdiction" type = "xsd:string"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name = "Party">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element ref = "PartyAddress"/>
                                <xsd:element ref = "Representative"/>
                        </xsd:sequence>
                        <xsd:attributeGroup ref = "Editable"/>
                        <xsd:attribute name = "Name" type = "xsd:string"/>
                        <xsd:attribute name = "Role" type = "xsd:string"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:element name = "Representative">
                <xsd:complexType>
                        <xsd:attributeGroup ref = "Editable"/>
```

```
                <xsd:attribute name = "RepName" type = "xsd:string"/>
                <xsd:attribute name = "RepTitle" type = "xsd:string"/>
                <xsd:attribute name = "RepSignatureURL" type = "xsd:string"/>
        </xsd:complexType>
</xsd:element>
<xsd:element name = "PartyList">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element ref = "Party" maxOccurs = "unbounded"/>
                </xsd:sequence>
                <xsd:attributeGroup ref = "Editable"/>
        </xsd:complexType>
</xsd:element>
<xsd:element name = "AgreementList">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element ref = "Section" maxOccurs = "unbounded"/>
                </xsd:sequence>
                <xsd:attributeGroup ref = "Editable"/>
        </xsd:complexType>
</xsd:element>
<xsd:element name = "Section">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element ref = "Clause" minOccurs = "0" maxOccurs = "unbounded"/>
                        <xsd:element ref = "LegalText" minOccurs = "0"/>
                </xsd:sequence>
                <xsd:attribute name = "Title" type = "xsd:string"/>
                <xsd:attributeGroup ref = "Editable"/>
                <xsd:attribute name = "ID" type = "xsd:string"/>
                <xsd:attribute name = "Include_Optional" type = "xsd:boolean"/>
        </xsd:complexType>
</xsd:element>
<xsd:element name = "Clause">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element ref = "Clause" minOccurs = "0" maxOccurs = "unbounded"/>
                        <xsd:element ref = "LegalText" minOccurs = "0"/>
                </xsd:sequence>
                <xsd:attribute name = "Title" type = "xsd:string"/>
                <xsd:attributeGroup ref = "Editable"/>
                <xsd:attribute name = "ID" type = "xsd:string"/>
                <xsd:attribute name = "Include_Optional" type = "xsd:boolean"/>
        </xsd:complexType>
</xsd:element>
<xsd:element name = "ParameterList">
        <xsd:complexType>
                <xsd:sequence>
```

```
                    <xsd:element ref = "Parameter" maxOccurs = "unbounded"/>
                </xsd:sequence>
                <xsd:attributeGroup ref = "Editable"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name = "Parameter">
            <xsd:complexType>
                <xsd:choice>
                    <xsd:element name = "variable">
                        <xsd:complexType>
                            <xsd:simpleContent>
                                <xsd:extension base = "xsd:string">
                                    <xsd:attributeGroup ref = "Editable"/>
                                </xsd:extension>
                            </xsd:simpleContent>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name = "date">
                        <xsd:complexType>
                            <xsd:simpleContent>
                                <xsd:extension base = "xsd:date">
                                    <xsd:attributeGroup ref = "Editable"/>
                                </xsd:extension>
                            </xsd:simpleContent>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name = "number">
                        <xsd:complexType>
                            <xsd:simpleContent>
                                <xsd:extension base = "xsd:double">
                                    <xsd:attributeGroup ref = "Editable"/>
                                </xsd:extension>
                            </xsd:simpleContent>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:choice>
                <xsd:attributeGroup ref = "Editable"/>
                <xsd:attribute name = "paramtype" type = "ParameterType"/>
                <xsd:attribute name = "ID" type = "xsd:string"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name = "PartyAddress">
            <xsd:complexType>
                <xsd:attributeGroup ref = "Editable"/>
                <xsd:attributeGroup ref = "Address"/>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name = "LegalText">
```

```
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name = "LegalTextPart" maxOccurs = "unbounded">
                                        <xsd:complexType>
                                                <xsd:simpleContent>
                                                        <xsd:extension base = "xsd:string">
                                                                <xsd:attributeGroup ref = "Editable"/>
                                                                <xsd:attribute name = "type" use = "required"
type = "LegalTextPartType"/>
                                                                <xsd:attribute name = "displaytext" type =
"xsd:string"/>
                                                        </xsd:extension>
                                                </xsd:simpleContent>
                                        </xsd:complexType>
                                </xsd:element>
                        </xsd:sequence>
                        <xsd:attributeGroup ref = "Editable"/>
                </xsd:complexType>
        </xsd:element>
        <xsd:attribute name = "name" type = "xsd:string"/>
        <xsd:attribute name = "role" type = "xsd:string"/>
        <xsd:attribute name = "repName" type = "xsd:string"/>
        <xsd:attribute name = "repTitle" type = "xsd:string"/>
        <xsd:attribute name = "locallaw" type = "xsd:string"/>
        <xsd:attribute name = "value" type = "xsd:string"/>
        <xsd:attribute name = "repSignatureURL" type = "xsd:string"/>
        <xsd:attribute name = "AddressStreet" type = "xsd:string"/>
        <xsd:attribute name = "AddressCity" type = "xsd:string"/>
        <xsd:attribute name = "AddressCountry" type = "xsd:string"/>
        <xsd:attribute name = "AddressPostcode" type = "xsd:string"/>
        <xsd:attribute name = "Text" type = "xsd:string"/>
        <xsd:attribute name = "Date" type = "xsd:date"/>
        <xsd:element name = "Semantics">
                <xsd:complexType>
                        <xsd:simpleContent>
                                <xsd:extension base = "xsd:string">
                                        <xsd:attributeGroup ref = "Editable"/>
                                        <xsd:attribute name = "Semantics_Format" type = "xsd:string"/>
                                </xsd:extension>
                        </xsd:simpleContent>
                </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "LegalTextPartType">
                <xsd:restriction base = "xsd:string">
                        <xsd:enumeration value = "Text"/>
                        <xsd:enumeration value = "ParameterReference"/>
                        <xsd:enumeration value = "InternalReference"/>
                        <xsd:enumeration value = "ExternalReference"/>
```

```
                    <xsd:enumeration value = "RegulatoryReference"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "ParameterType">
                <xsd:restriction base = "xsd:string">
                    <xsd:enumeration value = "Integer"/>
                    <xsd:enumeration value = "String"/>
                    <xsd:enumeration value = "Date"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:attribute name = "court" type = "xsd:string"/>
</xsd:schema>
```