MIRACLE

Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution

Specific Targeted Research Project: 248195

D2.1 State of the art on data specifications

Work package 2

Leading partner: TNO

June, 2010

Version 1.0

DOCUMENT INFORMATION	
ID	D2.1 State of the art on data specifications
Work Package(s)	Work package 2 Data Specification
Type	Report
Dissemination	Public
Version	Final
Date	June 30th 2010
Author(s)	M.J. Konsman (TNO), F.J. Rumph (TNO)
Reviewer(s)	Alexandr Savinov (SAP)

Table of Contents

1 Management summary

This document provides an overview of the state of the art that is relevant for WP2 of the Miracle project; Data Specification. There are two subjects that form the main focus of this state of the art report; modeling approaches and existing models.

Three modeling approaches are described; Unified Modeling Language (UML), UN/CEFACT's Modeling Methodology (UMM) and Object Role Modeling (ORM).

For the existing models, two international standard organizations are relevant; ebiX and IEC.

ebiX has developed models for Customer Switching Process and for the Exchange of Metered Data. These models describe the business processes and the corresponding message definitions and do so by using the aforementioned UN/CEFACT's Modeling Methodology.

The Common Information Model (CIM) is a data model by the IEC that aims to describe all major objects that an electric utility enterprise is typically involved with.

The following conclusions are drawn:

- UMM is the methodology of choice for the development of the WP2 deliverables. It is an international standard that describes different viewpoints that help guide the process of modeling. The artifacts that are part of these viewpoints are UML based. UMM has also been adopted by ebiX which serves as a good example of the application of UMM for the energy area.
- The main subject of the Miracle project; shiftable consumption and/or production is not being covered by the existing models in the energy area. Therefore specific models will have to be developed in Miracle that are able to cope with these concepts.
- Part of the Common Information Model by the IEC provides a solid basis for the Miracle data model. Basic energy concepts that already have been modeled can be reused.

2 Introduction

This document describes the state of the art that is relevant for work package 2 of the Miracle project; Data Specification.

For this state of the art overview the approach is twofold, it focuses on:

- Modeling approaches. When it comes to data modeling and message specifications - which are the subject of WP2 - the application of a solid methodology is of great aid. It not only encourages systematic development but it also eases the propagation and dissemination of the WP2 deliverables to the outside world.
- Existing models. One of the main ideas of Miracle is to have shiftable consumption and/or production. It is important to first look at existing models (either data models or message specifications) and see what part of these models can be utilized to realize Miracle project goals before developing specific models within the project. On an international level there are two organizations that are involved with this kind of modeling; ebiX and IEC.

Chapter 3 of this document describes several modeling approaches. The subsequent chapter 4 describes XML schema languages that can be used to implement message specifications. Chapters 5 and 6 deal with the existing models and describe those of ebiX and IEC respectively. Chapter 7 contains the WP2 conclusions with respect to the state of the art.

2.1 Scope

The scope of this state of the art document is limited to conceptual modeling. It describes the modeling approaches and existing models in relation to the objectives of WP2; the specification of message that describe flexibilities in energy consumption / production for domestic scale users and how these flexibilities are to be 'used' or allocated. This will most probably result in messages for; requesting energy consumption/production, negotiation and allocation.

2.1.1 Out of scope

Implementation issues such as the transport protocol for the exchange of messages or how messages should be stored persistently in databases are explicitly out of scope for WP2.

3 Modeling approaches

This section provides a number of approaches and (graphical) languages for data modeling.

3.1 Entity relationship modeling

Entity relationship modeling or entity relationship diagramming is common term for designing, describing and specifying data models. A plethora of specific techniques can be identified which could be viewed upon as entity relationship modeling. The approaches for data modeling in Unified Modeling Language (UML) and UN/Cefact's Modeling Methodology (UMM) as described further on can be used to model entities and their relationships. They do however support a broader offering of modeling purposes, e.g. modeling of functionality, activities, interactions, states, etc. In this section we restrict ourselves to techniques focused purely on entity relationship modeling.

Bachman introduced the data structure diagram in [Ba69]. The approach is also known as network modeling. Bachman defines four main concepts, an entity being a real world object, an entity set being a set of entities which based on their characteristics can be considered collectively, entity set denoting the subordinate relationship of a group of entities of one entity class with one entity of a different entity class and finally set class denoting the grouping of such sets based on their similarities. Currently the informal use of the word entity often refers to a class of entities instead of a single real world entity (which is often named instance or object in other approaches). A set class is often referred to as a relationship. However the set class concept is generalized to support varying multiplicities on either end of the relationship and to support not just binary but n-ary relationships. This is e.g. described in [Co70], wherein Codd describes the use of relational algebra in information systems to overcome data dependencies in information systems.

The entity relationship model was introduced by Chen in [Ch76] as generalizations of previous approaches. Also the concepts of entity and entity set are used. Relationship sets are mathematical relationships between a number of entities of a number of corresponding entity sets wherein each entity has a role in the relationship. Attributes denote information of an entity or relationship which can be observed or measured as values from a value set. Moreover a notation for entity-relationship diagrams is introduced (see Figure 1).

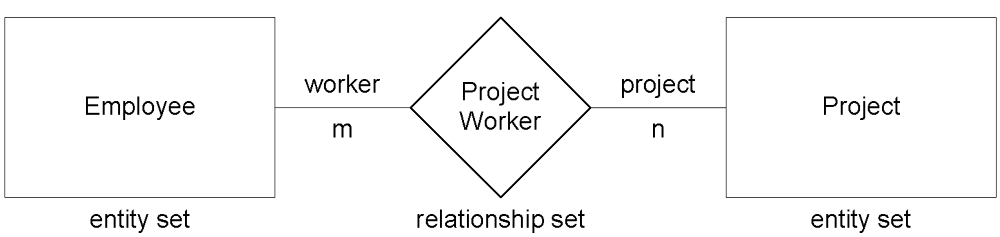


Figure 1: Entity-relationship diagram example, adopted from [Ch67]

From this work a number of approaches are created extending the entity-relationship modeling approach further. Examples of such extensions allow modeling of abstractions or generalizations and various types of constraints such as on the cardinality in relationships or on the uniqueness of entities. A design methodology for relational databases using an extended entity relationship model is described by Teorey et all in [TeYaFr86]. The use of entity-relationship models in modeling activities is described in a methodology and entity-relationship. In this paper a number of extensions of the original entity-relationship diagram are used. Saiedian provides a thorough evaluation of the extended entity-relationship model in [Sa97].

3.2 Unified Modeling Language

The Unified Modeling Language (UML, c.f. [UML]) is a general purpose software engineering with which the structure and behavior of software systems can be modeled. The UML specification, version 2.3 is the latest UML specification release at the time of writing, is maintained by the Object Management Group ([OMG]). Various organizations contribute and have contributed to this specification.

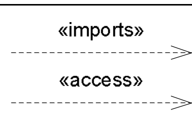
The UML is comprised of semantics and graphical notation specifications of software system modeling in 14 diagram types:

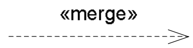
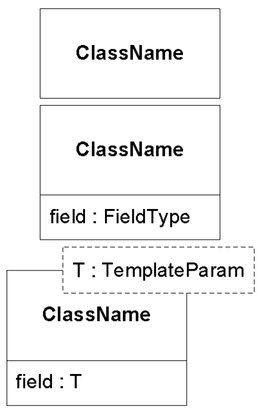
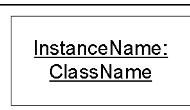
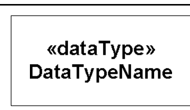
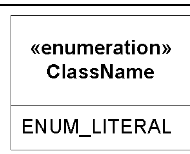
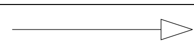
- **Structure Diagrams:** class diagram, object diagram, component diagram, composite structure diagram, package diagram, deployment diagram, and profile diagram.
- **Behavior Diagrams:** use case diagram, activity diagram, and state machine diagram.
- **Interaction Diagrams:** sequence diagram, communication diagram, timing diagram, and interaction overview diagram.

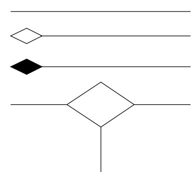
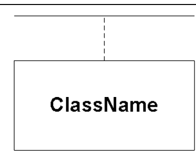
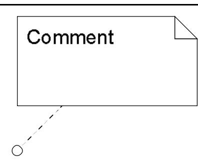
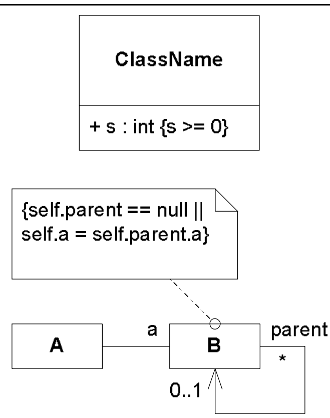
In relationship to data modeling primarily the package, class and object diagrams are relevant. Other diagram types will not be discussed in this section. Package diagrams are suited to model packages (groups of classes and other elements). Class diagrams are suited to model the types within a data model. Object diagrams can be used to model, e.g. as an example, instances of these types. Both will be discussed below. The elements of the UML notation are not bound per se to a specific diagram type, so first the elements which can occur within these aforementioned diagrams are introduced before showing their use in a diagram type.

3.2.1 UML elements for data modeling

This section provides a brief overview of the UML elements which are useable in a data modeling effort. This overview is by no means aimed to be complete; the reader is referred to the [UML] specification for the standard itself.

Element	Notation	Description
`Package`		A package groups a number of classes and their relationships.
`PackageImport`		Using the import relationship between packages it is possible to refer to elements in the imported package from the importing package with unqualified

		names. The visibility, either public or private, is denoted with the imports or access keyword respectively. With a publically visible import, other packages which import the importing package also import the imports of that package. With a privately visible import this is not the case.
`PackageMerge`		A merge relationship between packages is similar to a generalization relationship between classes. The merging package will inherit all elements from the merged package.
`Class`		A class denotes a type. Fields can be supplied for a class. A class can have fields of a specified type. For fields the cardinality, the default value, whether the field is read only or not, whether it is a derived field and whether it is composite field. Within UML classes can also have methods, but these are not used in data models. Template parameters can be provided to for instance allow variance of the type of a field.
`Instance Specification`		An instance specification is often referred to as an object. It is specified by its class name of which it is an instance and optionally a name of the instance.
`DataType`		A data type is a specific type of class which indicates a (more) primitive data type, such as string or integer. Its instances are values, not objects.
`Enumeration`		An enumeration is specific type of class of which the only instances which can exist are provided as literals in the enumeration itself. They are used to specify e.g. a finite set of options.
`Generalization`		A generation relationship between two classes indicates that the class from which the relationship originates inherits all fields and other aspects from the terminating class. E.g. a vehicle is a

		generalization of a car.
`Association`		An association indicates a relationship between two classes where the classes have each other as a property. The cardinality of and the navigability from either end of the association can be provided. The association is indicated as a derived association. The aggregation kind of the association can be none (normal line), shared (open diamond) and composite (normal line). A shared aggregation indicates that the instances in the aggregation are shared with other associations. In a composite aggregation the aggregating class is responsible for the existence of instances in the aggregation.
`Association Class`		A class can be associated with an association. This class indicates the features (parameters and when modeling functionality also methods) of the association.
`Comment`		A comment can be used to add remarks to the user of the model. A comment carries no semantic force.
`Constraint`		A constraint constraints the use of the model. E.g. indicating the allowed range of class parameters. Constraints are denoted within curly braces, either in a comment element, in textual form in e.g. the field definition or as dotted line across relationships with the constraint as textual attachment to the line (not presented here).

3.2.2 Package diagram

Figure 2 presents a UML package diagram which includes the commonly used elements. It presents 4 packages, A to E. Package A contains packages B to E. Package B publically imports package C and privately imports package D. Thus within package B all references to elements in packages C and D can be unqualified. Any package which would import package B will automatically import package C but not package D. Package E is a merger of its own elements and those of package D.

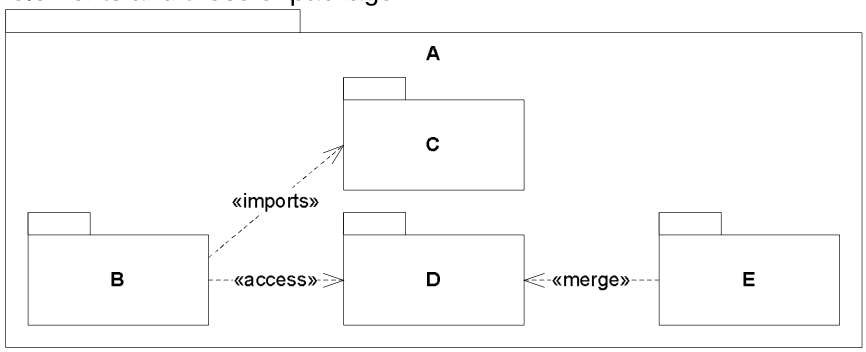


Figure 2: Example UML package diagram

3.2.3 Class diagram

Figure 3 shows an example UML class diagram. From left to right and top to bottom the diagram can be read as follows. Please note that names for association ends are omitted for reasons of brevity.

A class A is related to B of which 0 to positive infinity number of instances can exist. The lifetime of these instances is not bound to the lifetime of the aggregating instance of class A. B in turn has a property which is of the data type C. A class A is related to another class D. The instances of D in this relationship are bound to the instance of A which aggregates them. Minimally 2 and maximally 5 instances of D can exist in this relationship.

Class D has a property named field of type C and a property of the enumeration type E. E has three enumeration literals X, Y and Z. These enumeration literals are the only instances of E which can exist without extension of E.

D is associated with F. This association is type by class G. H is a specialization, or extension, of F and visa versa, F is a generalization of H. This means that an instance of D can also have this association with an instance of H.

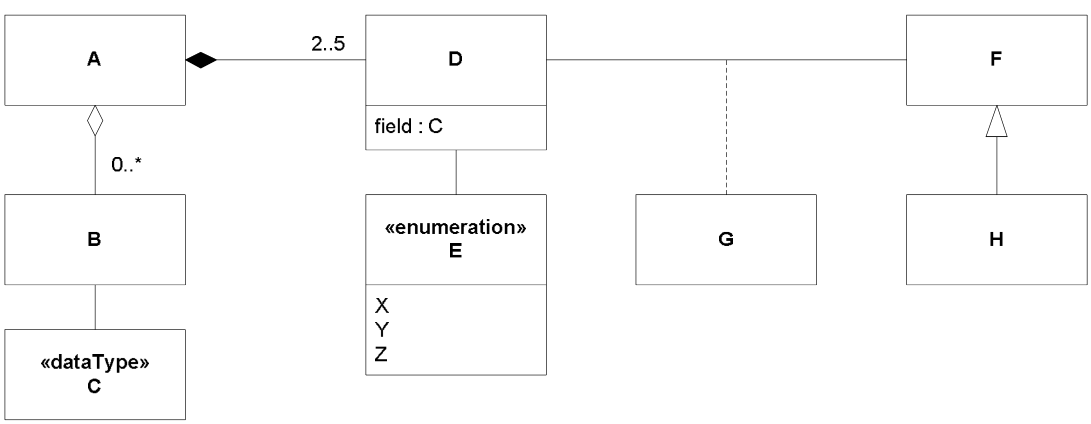


Figure 3: Example UML class diagram

3.2.4 Object diagram

Figure 4 presents a UML object diagram. It consists of an instance of class `A` named `a`. This object is associated with three instances of class `B` named `b1`, `b2` and `b3`. Instance `b3` has a value 3 set for the parameter named field.

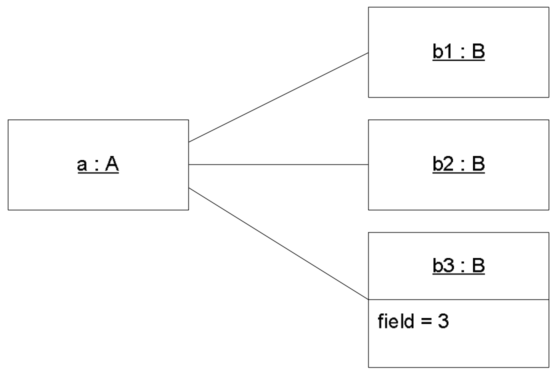


Figure 4: Example UML object diagram

3.3 UN/CEFACT's Modeling Methodology

The UN/CEFACT's Modeling Methodology (UMM) is managed by Techniques and Methodologies Group (UNTMG) of the Centre for Trade Facilitations and Electronic Business of the United Nations (UN/CEFACT), see further [UNTMG] for specifications. The UMM is a UML based approach for improving interoperability in trans-organizational business interactions.

The UMM meta model consists of four functional levels as depicted in Figure 5. The base module is a specification concerned with fundamental principles of the UMM on which other UMM specifications are based. It is related to e.g. the packaging and registration of specification elements. The foundation module specifies the views and their elements by which business collaboration models are to be constructed when using the UMM. Specialization and extension modules are extensions to the foundation module, e.g. to allow complementary types of analysis. The specialization modules are managed by UN/CEFACT and are potential candidates for inclusion into the foundation module, whereas the extension modules serve the same purpose but are maintained by other organizations.

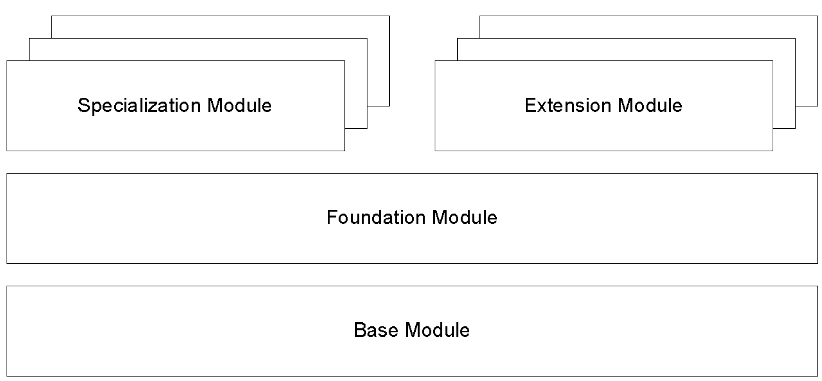


Figure 5: UMM modules, adopted from [UMMFMv1]

The UMM foundation module consists of three main views:

1. `BusinessDomainView`, focused on the analysis, not design, of business partners and their participation in business processes.
2. `BusinessRequirementsView`, focused on the further analysis of business interactions and the definition of requirements for these interactions. The sub-views in this view are: `BusinessProcessView`, `BusinessEntityView`, `CollaborationRequirementsView`, `TransactionRequirementsView` and the `CollaborationRealizationView`.
3. `BusinessTransactionView`, focused on the design of business transactions according to the requirements defined in the `BusinessRequirementsView`. The sub-views are: `BusinessChoreographyView`, `BusinessInteractionView` and `BusinessInformationView`.

Many of the views and sub-views within the UMM foundation module are related to business processes and behavior. In this review of the UMM with the focus on data modeling only the `BusinessEntityView` is discussed below.

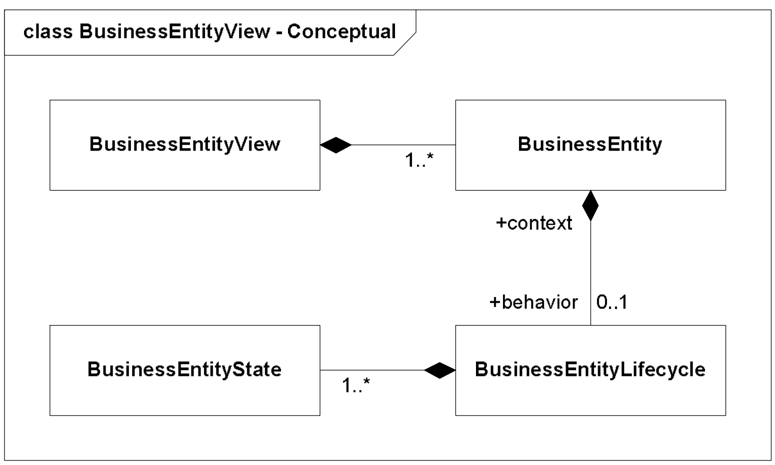


Figure 6: BusinessEntityView, adopted from [UMMFMv1]

Figure 6 shows the classes which are part of the `BusinessEntityView`. A `BusinessEntity` is a conceptualization of something within the real world, e.g. an account or order. I.e. no technology and implementation related entities are identified within this view. The `BusinessEntityLifecycle` (i.e. a class' state machine) describes the `BusinessEntityStates` of the `BusinessEntity` if this is deemed relevant by the modeler.

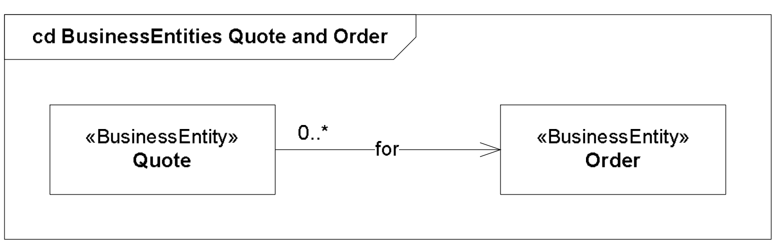


Figure 7: UMM class diagram example, adopted from [UMMFMv1]

Figure 7 shows a class diagram as an example which specifies the classes `Quote` and `Order` and their association.

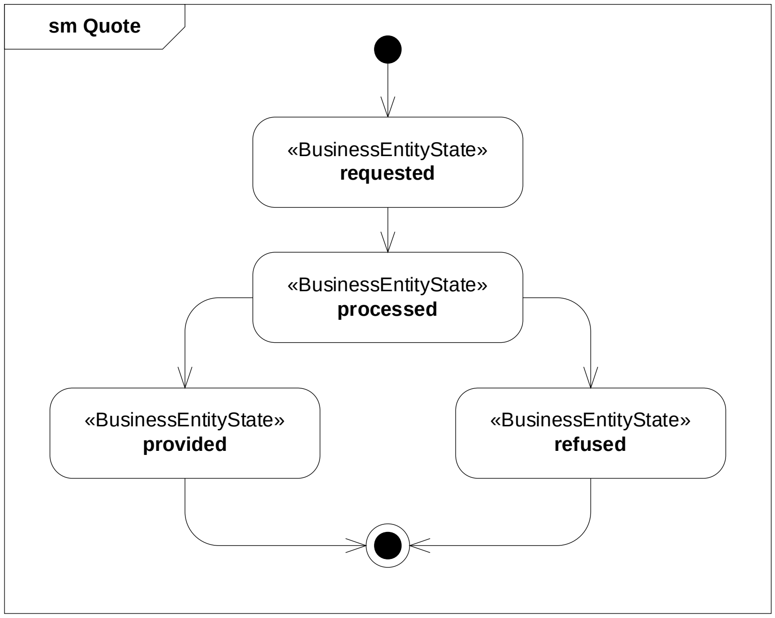


Figure 8: UMM state diagram example, adopted from [UMMFMv1]

Figure 8 shows a state diagram which by example shows the specification of possible states of the `Quote BusinessEntity`.

3.4 Object Role Modeling

Object Role Modeling (ORM) is a modeling notation on a conceptual level, which is based on modeling all facts on a conceptual level as relationships. The concept of an attribute and thus the grouping of information elements which can be seen in for instance UML and traditional ER (entity relationship) modeling are considered relevant on only an implementation level, not on a conceptual level. This approach is dubbed fact-oriented. The approach was first formalized by Dr. T Halpin in his PhD thesis [Ha89]. He also wrote numerous papers on ORM, such as [Ha99] on the comparison between entity relationship modeling approaches and ORM. A second version of the ORM notation is described in [Ha05] and [Ha09].

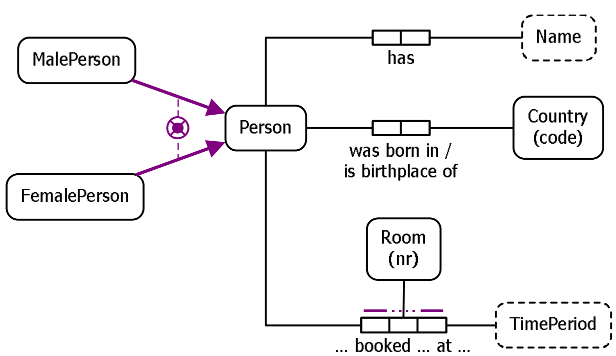


Figure 9: Example ORM diagram

The main elements within the ORM are the entity and value elements which can have n-ary relationships to express facts; of course within the universe of discourse dealt with in the conceptual model. For instance in Figure 9 the fact that a person has a name is expressed; here `Person` is an entity and `Name` is a value. This shows the ease with which facts expressed in ORM can be formulated in a natural language.

The figure also shows the describe the relationship between e.g. entities in both directions as with the birthplace of a person: a person was born in a country and a country is the birthplace of a person. A fact with a ternary relationship between entities is also shown; a person booked a room at a time period. The use of the horizontal ellipsis indicates the role which the entities or values have in the fact; they correspond with the boxes which are each connected with the entities or values. The uniqueness constraint indicated by the solid, then the dashed and then the solid line above the three boxes are to be read as: only one person can book a room for a certain time period.

Finally the figure shows the subtype relationship. In this example the gender of a person is modeled as the fact that a Person is either a MalePerson or a FemalePerson. Please note the exclusive-or constraint expressed with the dotted line with the circular icon. Many other constraint types can be expressed which are not detailed further in this document; the reader is referred to the references indicated earlier.

4 XML schema languages

Although not strictly part of the data modeling or message specification itself, which can be performed using the modeling approaches described in chapter **Error! Reference source not found.**, the implementation of these message specifications is a closely related subject. This chapter will touch upon the most common XML schema languages that are relevant to Miracle.

4.1 XML schema languages

Over the years several standards have been developed to structure the syntax and content of XML documents. The most important standards are: Document Type Definition (DTD), RELAX NG and XML Schema.

4.1.1 Document Type Definition (DTD) [DTD]

A DTD is associated with an XML file. This association can be internal (part of the XML file it is associated with) as well as external (in a separate file that is being referred to by the XML file).

Several XML building blocks can be defined by a DTD: Elements, Attributes, Entities, PCDATA (Parsed Character data) and CDATA (Character data).

DTD lacks several important features. I.e. it is not possible to define data types, there is no support for regular expressions and namespaces, only limited possibilities for adding constraints. Also a DTD is described in a separate language and not in XML.

These disadvantages lead to the development of new XML schema languages.

4.1.2 REgular LAnguage for XML Next Generation (RELAX NG) [RELAXNG]

RELAX NG addresses a lot of the issues associated with DTD. The first version of RELAX NG was published in 2003 and was developed by OASIS.

An important feature of RELAX NG is that there are actually two types of describing a RELAX NG schema. It can be described using XML, but there is also a notation that is very similar to DTD called RELAX NG compact syntax. Both formats are completely interchangeable.

4.1.3 XML Schema [XMLSchema]

XML Schema was published as a W3C recommendation in 2001. Its features are very similar to RELAX NG. XML Schema's can only be expressed as XML.

One of the main advantages of XML Schema over RELAX NG is the amount of built in data types such as; string, boolean, dateTime, etc.

XML Schema is the most popular XML schema language and is also being used by organizations such as ebiX [ebiX] to implement their message specifications.

4.2 Concluding remarks

Given its expressive power and its widespread use XML Schema will be the schema language of choice for the Miracle project.

5 ebIX Models

5.1 About ebiX

ebiX [ebiX] stands for "European forum for energy business information exchange". The main focus of ebiX is to promote the use of electronic information exchange in the energy industry. More specifically interchanging administrative data for the internal European markets for electricity and gas.

5.1.1 ebiX members

The following companies are represented in ebiX: A&B, BDEW, CapGemini, Danisch Energy, EdiSys, Energinet.dk, E.ON. Energie, E WIE EINFACH Strom & Gas GmbH, Indexis, Interelectra, KEMA, NordPool, RWE, Statnett, SvK, swissgrid, TenneT.

5.2 Organisation

The following diagram depicts the ebiX organization:

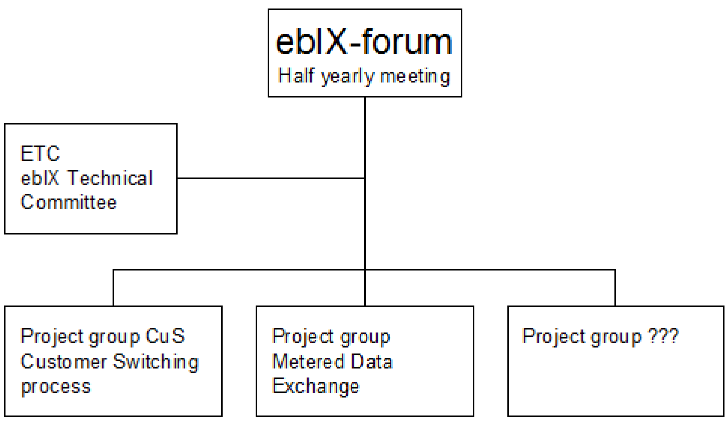


Figure 10: ebiX organogram, adopted from [ebiXorg]

As can be seen from the diagram ebiX has a flat organizational structure. The ebiX-forum is a half yearly meeting that approves of products, project plans and budgets. All members are part of the ebiX-forum with the exception of NordPool (observing member) and EdiSys (Secretary/Consultancy for ebiX).

The ETC is a permanent working group and responsible for the technical part of the standards. More specifically its tasks include: maintenance of ebiX methodology that is based on standards such as UMM [UMMFMv1], providing the project groups with tools and guidelines, maintaining contact with other standardization bodies. Another important task of the ETC is to organize implementation support. ebiX states that do not impose by legal position but by common sense and practical and acceptable solutions.

Currently there are two project groups active within ebiX; Customer Switching process and Exchange Metered Data. The activities of these project groups will be the subject for the remainder of this chapter.

5.3 Activities

The two main activities of ebiX at this moment are:

- **Customer Switching process (CuS).** This process is very relevant for the liberalizing energy market. It describes the processes and the electronic data exchange involved in switching a customer from one supplier to another.

  The CuS members are: EDSN, EdiSys, Energinet.dk, Energie Ag, RWE, Statnett, Swissgrid, TenneT, UMIX, WM data and Vattenfall. The following companies are observing members: AMT-Sybex, SAP, Kisters and Vychodoslovenska energetika a.s.
- **Exchange Metered Data (EMD).** This model describes the exchange of metered data in the energy domain.

  The EMD members are: A&B, Energinet.dk, RWE, SvK, Swissgrid, TenneT and UMIX. The following companies are observing members: EDF, Eesti Energia, SAP and VSE.

The models will be described in more detail in the following paragraphs.

5.4 Customer Switching (CuS) process [ebiXCuS]

The figure below shows all the use cases that the ebiX CuS model consists of.

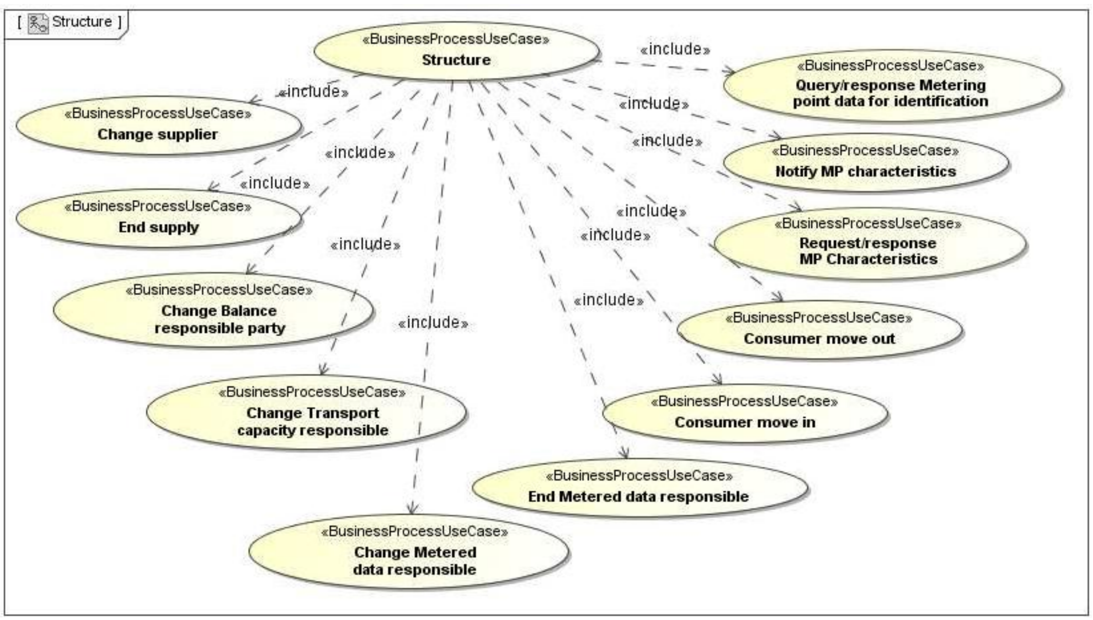


Figure 11: CuS Use case overview, adopted from [ebiXCuS]

Each of the use cases described here is a process area in UMM terminology.

- **Change supplier.** A Consumer will ask for a new contract with a Balance supplier. The Balance supplier verifies all the data needed for the contract, such as name, address, etc.
- **End supply.** During this process the Metering point administrator is informed by the Balance supplier that the supply ends. Both the Balance supplier and the Balance responsible party are removed from the metering point, the customer remains associated with the metering point.
- **Change Balance responsible party.** The Metering point administrator is informed by the Balance supplier that the Balance responsible party is changed for a specific metering point.
- **Change Transport capacity responsible.** The Metering point administrator is informed by the Balance supplier that the Transport capacity responsible is changed for a specific metering point.
- **Change Metered data responsible.** The Metering point administrator is informed by the Balance supplier that the Metered data responsible is changed for a specific metering point.
- **End Metered data responsible.** The Metering point administrator is informed by the Balance Supplier that the Metered data responsible is no longer associated with a specific metering point.
- **Consumer move in.** This process area involves other process areas that have been described here. The main one being “Change supplier”. The roles involved in the “Consumer move in” are: Balance supplier, Consumer and Metering point administrator.
- **Consumer move out.** In this case there are also other process area involved, most notably “End supply”. The roles are the same as for “Consumer move in”; Balance supplier, Consumer and Metering point administrator.
- **Request/response MP Characteristics.** Each relevant role can ask for the characteristics of a metering point. These will be provided by the Metering point administrator. The requesting role can be any of the following; Balance responsible party, Balance supplier, Grid access provider, Metered data aggregator, Metered data collector, Metered data responsible, Party connected to grid, Reconciliation responsible, Transport capacity responsible.
- **Notify MP characteristics.** This process area is similar to the one described above except that in this case the initiating role is played by the Metering point administrator. The receiving role can be any of the following; Balance responsible party, Balance supplier, Grid access provider, Metered data aggregator, Metered data collector, Metered data responsible, Party connected to grid, Reconciliation responsible, Transport capacity responsible.
- **Query/response Metering point data for identification.** The initiating role verifies whether the identification data for this metering point is valid. It does so by sending a request to the Metering point administrator. The request is initiated by the Balance supplier or the Metered data collector.

For all process areas business processes are provided in combination with the definition of the data elements that are part of the messages that need to be exchanged in the context of a business process.

5.5 Exchange Metered Data

The EMD information model is quite extensive and spans several documents [ebiXEMDDocs] including the following subjects: billing, determine switch stand, imbalance settlement, reconciliation. At the heart of these documents is an introduction document [ebiXEMDIntro] that describes the use case that is central to all subjects; measure. This use case is depicted below:

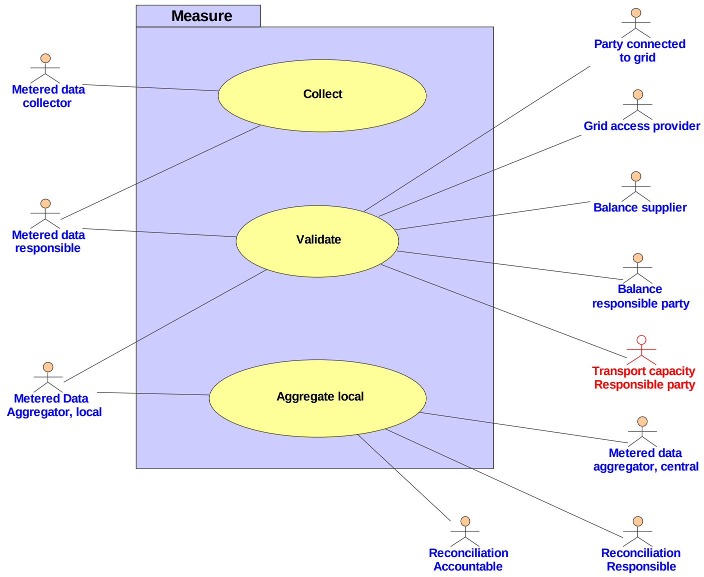


Figure 12: Measure use case, adopted from [ebiXEMDIntro]

As can be seen the measure use case consists of three lower level use cases; collect, validation and Aggregation local.

- **Collect.** This use case has involves only one role; the Metered data collector. This role uses the Master Data Register to gather all relevant information on the metering point so that it can collect the metering data. After the use case is finished the collected data is available for exchange with other roles.
- **Validate.** The collected data needs to be validated. This task is performed by the Metered Data Responsible role by comparing the collected data with historical data. Should data be missing historical data and/or control readings can be used to fill in the gaps.
- **Aggregate local.** The Metered Data Aggregator local role aggregates the validated data for one of two purposes; settlement or reconciliation. Depending on the purpose and the local regulation the rules for the aggregation process may differ. An example would be the rules for precision and rounding of aggregated data.

Similar to the CuS model all use cases are further elaborated on; each use case has a business process and accompanying message definitions.

As aforementioned the measure use case is a generic use case that is central to all other use cases. The more specific use cases are described in separate documents.

5.6 Relevance for the Miracle project

With regard to the relevance for Miracle the following observations can be made:

- ebiX makes use of the UMM methodology. This matches very well with what should be delivered by WP2. The ebiX documents take use cases as their starting point and subsequently design a business process and the corresponding messages. Thus providing the messages with the context they'll be used in.
- At this moment the ebiX models do not cover the area that is targeted by Miracle. However, the EMD project will have to be taken into account for Miracle processes that touch on billing.
- ebiX actively aligns its work with the Harmonized Electricy Market Role Model by ENTSO-E [entsoeMod]. This model has also been adopted by the Miracle project.

Despite the fact that there does not seem to be much overlap with the application area that Miracle is targeting, ebiX is still relevant for WP2. The application of the UMM methodology by ebiX can be of guidance for the WP2 deliverables.

6 IEC Common Information Model

6.1 About IEC

The International Electrotechnical Commission is a non-profit, non-governmental standards organization. The IEC has a very broad scope, this becomes apparent from the fact that IEC "prepares and publishes International Standards for all electrical, electronic and related technologies — collectively known as 'electrotechnology'" [IEC].

6.2 IEC members

The members of the IEC consist of similar organizations on the national level. At this moment 59 countries have full membership and 21 are associate members.

6.3 Organization

The figure below shows the organizational structure of the IEC

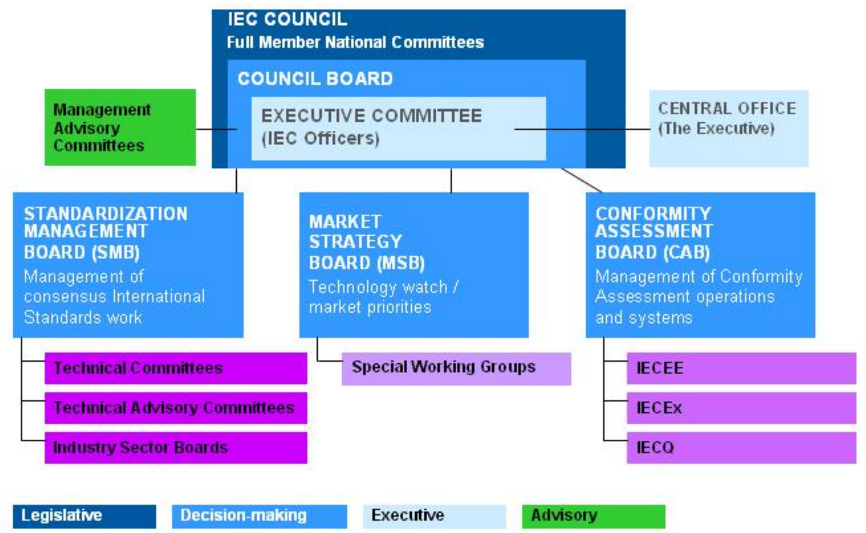


Figure 13: IEC organogram, adopted from [IECOrg]

Only the outlines of the structure will be sketched here. A more detailed description can be found on the IEC website [IECOrg].

The highest authority within the IEC is the IEC council. It consists of the presidents of all full member organizations and the IEC Officers (IEC president, treasurer, etc.). The management tasks are delegated to the Council board that implements IEC Council policies and comes up with recommendations for new policies. It also oversees the standardization work as being performed by the Standardization Management Board.

The Standardization Management Board is responsible for creating Technical Committees and deciding on the scope they should have. It is within these committees that the actual standards are being developed.

6.4 Common Information Model

The IEC's Common Information Model (CIM) [IECCIM]is a data model that focuses on describing all major objects that an electric utility enterprise is typically involved with. The model is described in UML class diagrams. The main overview of the model is shown below; it shows the top level packages of the model.

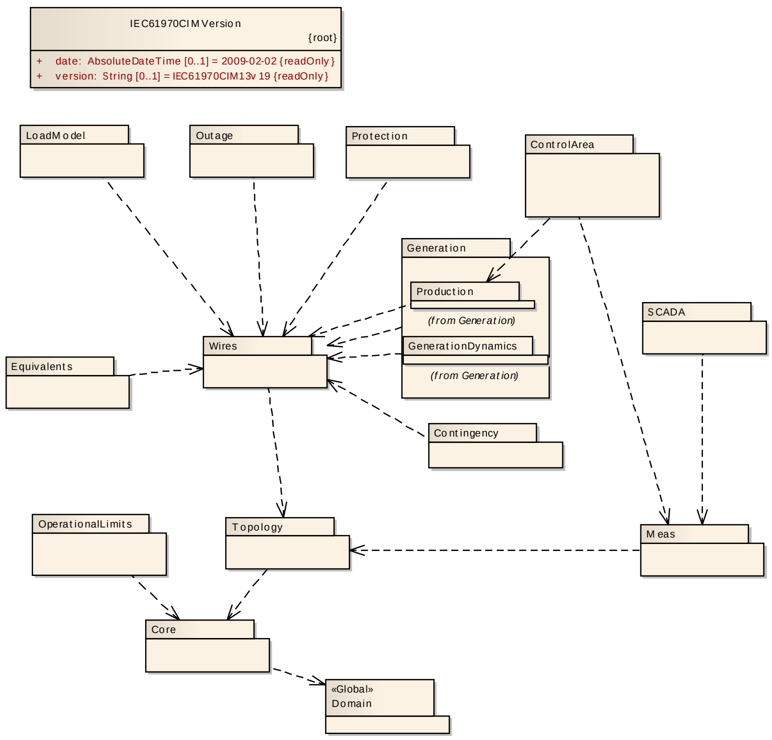


Figure 14: CIM package diagram, adopted from [IECCIM]

The packages are hierarchically structured. The package that all other packages are extended from (either directly or indirectly) is the Domain package. This package defines datatypes that may be used as attributes by classes in other packages.

As can be seen by the packages in the overview, the CIM's main focus is on the technical electrical infrastructure and not so much on aspects such as trading.

For a more detailed description please refer to [IECCIM].

6.5 Relevance for the Miracle project

With regard to the relevance for Miracle the following observations can be made:

- The International Electrotechnical Commission is an important player in the electricity world. The organization combines the efforts of organizations on a national level, as such the IEC is very influential.
- Although the focus of Miracle differs from the CIM focus, there will still be considerable overlap with the data modeling that needs to be performed within Miracle. E.g. the LoadModel package of CIM might be reusable to a large extent for forecasting activities that are part of the Miracle project.
- The CIM is a data model; it does not describe message exchanges. However it could be used as a basis for message definitions.

These observations lead to the conclusion that part of the CIM is relevant for Miracle. Classes described in CIM should be the basis for the data modeling within Miracle when applicable.

7 Conclusions

The following conclusions can be drawn from this WP2 state of the art overview.

- UMM is the methodology of choice for the development of the WP2 deliverables. It is an international standard that describes different viewpoints that help guide the process of modeling. The artifacts that are part of these viewpoints are UML based. UMM has also been adopted by ebiX which serves as a good example of the application of UMM for the energy area.
- The main subject of the Miracle project; shiftable consumption and/or production is not being covered by the existing models in the energy area. Therefore specific models will have to be developed in Miracle that are able to cope with these concepts.
- Part of the Common Information Model by the IEC provides a solid basis for the Miracle data model. Basic energy concepts that already have been modeled can reused.

References

[UML] OMG Modeling and Metadata Specifications, http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML, retrieved on June 18, 2010

[OMG] Object Management Group, http://www.omg.org/, retrieved on June 18, 2010

[UNTMG] United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) – Techniques and Methodologies Group (TMG), http://www.untmg.org/, retrieved on June 18, 2010

[UMMFMv1] UMM Meta Model – Foundation Module Version 1.0 Technical Specification, 2008, http://www.untmg.org/wp-content/uploads/2008/06/umm_foundation_module.pdf, retrieved on June 18, 2010

[Ha89] Halpin T. A Logical Analysis of Information Systems: static aspects of the data-oriented perspective, 1989.

[Ha99a] Halpin T. Entity Relationship modeling from an ORM perspective: Part 1. Journal of Conceptual Modeling, no. 11, 1999.

[Ha99b] Halpin T. Entity Relationship modeling from an ORM perspective: Part 2. Journal of Conceptual Modeling, no. 12, 2000.

[Ha99c] Halpin T. Entity Relationship modeling from an ORM perspective: Part 3. Journal of Conceptual Modeling, no. 13, 2000.

[Ha05] Halpin T. 'ORM 2', On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, Cyprus. Springer LNCS 3762, pp 676-87, 2005.

[Ha09] Halpin T. 'Object-Role modeling', Encyclopedia of Database Systems, Springer, 2009.

[Ba69] Bachman C. 'Data structure diagrams', ACM SIGMIS Database vol. 1 issue 2, pp 4-10, 1969.

[Ch76] Chen P.P. 'The entity-relationship model: Toward a unified view of data' ACM TODS 1, pp 9-36, 1976.

[TeYaFr86] Teorey T. J., Yang D., Fry J.P. 'A logical design methodology for relational databases using the extended entity-relationship model', ACM Computing Surveys, vol. 18 issue 2, pp 197-222, 1986.

[Sa97] Saiedian H. 'An evaluation of extended entity-relationship model', Elsevier Information and Software Technology 39, pp 449-462, 1997.

[DTD] DTD specification, http://www.w3.org/TR/REC-xml/#dt-doctype, retrieved on June 29 2010

[RELAXNG] RELAX NG home page, http://relaxng.org/, retrieved on June 29 2010

[XMLSchema] Schema W3C, http://www.w3.org/standards/xml/schema.html, retreived on June 29 2010

[ebiX] European forum for energy Business Information eXchange, http://www.ebix.org/, retrieved on June 21 2010

[ebiXorg] European forum for energy Business Information eXchange, http://www.ebix.org/content.aspx?ContentId=991&SelectedMenu=2, retrieved on June 21 2010

[ebiXCuS]	ebiX CuS Model, http://www.ebix.org/Documents/CuS%20Draft%20v2r1A%2020100409.pdf, retreived on June 21 2010
[ebiXEMDDocs]	European forum for energy Business Information eXchange, http://www.ebix.org/content.aspx?ContentId=1004&SelectedMenu=52, retrieved on June 21 2010
[ebiXEMDIntro]	ebIX Business information model for the exchange of metered data in the energy domain, http://www.ebix.org/documents/emd/model%20documents/ebix%20model%20measure%20introduction%201.0.d.doc, retreived on June 21 2010
[entsoeMod]	The Harmonised Electricty Market Role Model, http://www.entsoe.eu/fileadmin/user_upload/edi/library/role/role-model-v2009-01.pdf, retrieved on June 21 2010
[IEC]	International Electrotechnical Commission, http://www.iec.ch/, retrieved on June 21 2010
[IECOrg]	About the IEC > Structure and management, http://www.iec.ch/about/struct-e.htm, retrieved on June 21 2010
[IECCIM]	IEC61970-301 Ed. 3, Internation Electrotechnical Commission, 2009