MIRABEL

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

Specific Targeted Research Project: 248195

D7.5 MIRABEL-ONE:
Initial draft of the MIRABEL Standard
Work package – 7

Leading partner: TNO

December 22, 2011

Version 1.0

DOCUMENT INFORMATION	
ID	D7.5 MIRABEL-ONE
Work Package(s)	WP7 Standardization, Dissemination and Exploitation
Type	Report
Dissemination	Public
Version	1.0
Date	December 22, 2011
Author(s)	Jack Verhoosel, Diederik Rothengatter, Frens-Jan Rumph, Mente Konsman (TNO)
Reviewer(s)	Zoran Marinšek, (INEA)

Table of Contents

1 Summary

Use of renewable energy sources is enforced by national and international regulations. Drivers for such policies include mitigation of climate change due to emission of greenhouse gasses and reducing dependency on fossil fuel reserves. Due to the intermittent character of renewable energy sources such as photovoltaic or wind power, integration of such sources creates a challenge in maintaining balance between demand and supply. Indications of such challenges in countries with e.g. a high penetration of wind power are already showing in prices on power exchanges reaching zero or negative energy prices. In general, without mitigation measures, an increase in the use of intermittent renewable energy sources leads to a diminished ability to guarantee security of supply.

One of the possibilities to deal with this uncertainty of renewable energy sources is the exploitation of flexibility in electricity demand and supply. Consumers and producers own devices in which flexibility in electricity demand and supply is possible, such as washing machines, dishwashers, photovoltaic cells, micro combined heat and power units, electric heat pumps, and electric vehicles. These flexibilities include temporal shifts of activities (e.g. delay operation), temporary reductions of load comparable to existing demand response scheme's and adjustments in load profiles of charging of electric vehicles.

These flexibilities can be taken into consideration when scheduling the load and distributed generation. Thereby, such a system enables electricity suppliers to balance energy demand and supply in near real-time and thus, allows the integration of more renewable energy sources whose availability cannot be influenced. The use of flexibility is scheduled and is negotiated with the party offering the flexibility.

The goal of this document is to define a specification for modeling of such flexibility and the exchange of flexibility information between stakeholders in the energy domain, especially between consumers and electricity suppliers. The specification is described in terms of a generic data model for energy flexibility and messages for information exchange on flexibility offerings. The intention of this specification is to use it as input for formal European standardization and acceptance in the electricity market. More specifically, the document will be used as input to a CEN Workshop in 2012 that produces a CEN Workshop Agreement (CWA) in a relatively short and constructive time period.

By setting such a standard and the adoption of it in the electricity market, the consumer gets more and more involved in energy management within the local household, building or premise and with the connecting smart grid. Moreover, the European Commission, in the context of the 2020 initiative for an Energy Strategy, established at the end of 2009 a Smart Grid Task Force which identified in its work program that "Consumer empowerment includes capabilities of consumers to have sufficient and timely information on their actual energy consumption, to learn and act upon their energy savings potential through energy usage optimisation and more energy efficient technologies, to have access to competitive offers for energy services and to develop energy efficient consumption practices." The specification and the intended standard based on it is intended to add to this objective. The specification assumes is that the energy management system is a black box that produces offerings of flexibility. How these offerings are being generated, whether they are aggregations of multiple devices at the

premises or which devices are involved is of less importance for this specification. Moreover, it can be used for each of these situations.

The target audience of this document is all kinds of stakeholders in the energy domain that are directly or indirectly involved in the intended information exchange. This includes energy service providers, energy system/network operators, grid users, but also ICT service suppliers, European branch organizations, regulators and public authorities, research institutes and non-governmental organisations acting on environment matters.

The document is structured as follows. First, the background and scope of the specifications in this document is described in more detail. It includes the current situation in the energy domain, the need for utilizing flexibility in energy demand/supply and the stakeholders involved in this demand-response management issue. Then, the flexibility concept and the flex-offer is introduced and various examples of it are described. Also, the business advantages of its usage and the relation with existing approaches on demand-response management are briefly discussed. After that, the context of the intended information exchange is described in more detail. It includes the various roles and processes involved in demand-response management based on the ENTSO-E harmonized role model. It concludes with presenting the exact scope of the data model and information exchange in terms of these roles and processes.

After this introduction of the background, scope, concept and context, the core of the flexibility specification to be standardized is described. This includes the data model which is centered on the exploitation of flexibility in energy consumption and supply; the `FlexEnergy` concept. The concepts are formulated as generic as possible because of the focus on standardization. After that, the messages which use the data model for conveying information are described based on the specific business process that is associated with offering, accepting and assigning flexibility in consumption and production of energy. In order to form the bridge towards implementation of the specification, XML Schema Definitions are described based on the data model and messages specified. Finally, some example messages are described for the exchange of flex-offer information for various devices such as an electric vehicle, a heat pump and a combined heat and power system.

2 Introduction

One of the goals of the MIRABEL project is to disseminate the results of the project and provide a good basis for the acceptance of the results in the electricity market. One of the actions to be taken here is to preserve the specifications developed after the lifetime of the project itself by submitting them to a formal European standardisation body. As concluded in the previous MIRABEL deliverable D7.1, a good way to achieve this goal is to use the instrument of a CEN Workshop that produces a CEN Workshop Agreement (CWA) in a relatively short and constructive time period [VerhStap10].

The goal of this document is to define the specific results of the project that will be submitted to the standardization body for standardization purposes. Thereby the focus is on the specifications for modeling of flexibility and exchange of flexibility information between consumers and electricity suppliers. The type of specification that is described in this document is a generic data model for energy flexibility and messages for information exchange on flexibility offerings.

The target audience of this document is therefore all kinds of stakeholders in the energy domain that are directly or indirectly involved in such information exchange. This includes:

- Energy service providers: balance responsible parties, balance suppliers, metered data responsible, energy service companies, aggregators, applications and services providers, power exchange platform operators (market operators),
- Energy system and network operators (system operators): transmission and distribution system/network operators (TSOs and DSOs/DNOs),
- Grid users (party connected to the grid): producers including producers using renewable energy sources;, consumers (including mobile consumers), storage owners,
- ICT suppliers, software engineering and IT services suppliers, integrators,
- European associations: branch and trade associations such as ENTSO-E, ebIX and EFET,
- Regulators and public authorities, academics and research, laboratories,
- Non-governmental organisations acting on environment matters.

2.1 Document structure

Chapter 3 describes the background and scope of the specifications in this document in more detail. It includes the current situation in the energy domain, the need for utilizing flexibility in energy demand/supply and the stakeholders involved in this demand-response management issue. Chapter 4 introduces the flexibility concept and defines the flex-offer, various examples of it, the business advantages of its usage and the relation with existing approaches on demand-response management. Chapter 5 briefly introduces the approach we have used to model the flexibility information and the messages for information exchange. Chapter 6 describes the context of this information exchange in more detail by describing the various roles and processes involved in demand-response management based on the ENTSO-E harmonized role model. It concludes with presenting the exact scope of the data model and information exchange in terms of these roles and processes.

Chapter 7, 8, 9, and 10 contain the core of this document in terms of the specifications to be standardized. Chapter 7 contains the data model. The model itself is centered on the exploitation of flexibility in energy consumption and supply; the `FlexEnergy` concept. The concepts are formulated as generic as possible because of the focus on standardization. Chapter 8 discusses the messages which use the data model for conveying information. It describes the specific business process that is associated with offering, accepting and assigning flexibility in consumption and production of energy. Chapter 9 contains the XML Schema Definitions that have been defined based on the data model and messages specified for implementation purposes. Finally, chapter 10 contains some example messages for the exchange of flex-offer information to present to the reader in concrete steps how the specification can be used.

The Common Information Model (CIM) [CIM09] by the International Electrotechnical Committee (IEC) is being reused where possible in our data model. In Appendix A the elements adopted from CIM are listed.

3 Background and scope

Use of renewable energy sources is enforced by national and international regulations, c.f. [Europe2020] and [GREA]. Drivers for such policies include mitigation of climate change due to emission of greenhouse gasses and reducing dependency on fossil fuel reserves. Due to the intermittent character of renewable energy sources such as photovoltaic or wind power, integration of such sources creates a challenge in maintaining balance between demand and supply. Indications of such challenges in countries with e.g. a high penetration of wind power are already showing in prices on power exchanges reaching zero or negative energy prices [Energinet.dk]. In general, without mitigation measures, an increase in the use of intermittent renewable energy sources leads to a diminished ability to guarantee security of supply.

3.1 Flexibility and demand-response with scheduling

One of the possibilities to deal with this uncertainty of renewable energy sources is the exploitation of flexibility in electricity demand and supply. Consumers and producers own devices in which flexibility in electricity demand and supply is possible, such as washing machines, dishwashers, photovoltaic cells, micro combined heat and power units, electric heat pumps, and electric vehicles. These flexibilities include temporal shifts of activities (e.g. delay operation), temporary reductions of load comparable to existing demand response scheme's and adjustments in load profiles of charging of electric vehicles.

These flexibilities should then be taken into consideration when scheduling the load and distributed generation. Thereby, such a system enables electricity suppliers, or balance responsible parties in terms of the ENTSO-E Harmonized Electricity Role Model [ENTSO-E09], to balance energy demand and supply in near real-time and thus, allows the integration of more renewable energy sources whose availability cannot be influenced. The use of flexibility is scheduled and is negotiated with the party offering the flexibility. A hierarchical approach is needed for aggregation in order to cope with vast amounts of participants in the energy system. In the ENTSO-E Harmonized Electricity Role Model this is largely warranted by the structuring of the electricity market system.

3.2 Flexibility and energy management systems

From a consumer point of view, flexibility in energy demand is only one of many aspects to be considered when managing energy in a household, a larger residence building or industrial premises. Such an environment usually contains many devices that require or produce electricity. An Energy Management System (EMS) is able to take care of monitoring and control of the entire use of energy in the household, building or industrial premise. An EMS takes input from various sensors in the area and controls devices in order to achieve an optimum between various objectives, such as:

- usage of resources to minimize the import of electricity from the smart grid,
- maximize the use of the premises own energy generation,
- maintaining the comfort level within the desired limits,
- reducing the cost of energy consumption.

When considering flexibility to be applied in an EMS, i.e. using flexibility for matching demand and supply in the smart grid, a new objective is added. This objective is to further reduce cost of energy consumption through offering and negotiation of flexibility to be utilized by the smart grid. This has to be done such that smart grid balance is improved and penetration of intermittent renewable energy sources can be increased. Obviously, offering flexibility in return for a cost-reduction has to be weighted by the EMS with the other objectives and constraints for energy management.

In order to leverage this flexibility, the user has to set his preferences to be used for each of these devices in the EMS. An example element of such a preference could be a minimum and maximum level of an environment variable such as temperature that can be used as set-points for the control of a heating device. Another example preference could indicate whether or not a washing machine can be interrupted at certain fixed points within the washing programme, e.g. between washing and centrifuging. Obviously, it is the producer of the washing machine that has to make it possible that the washing machine can be interrupted at the various steps of these programmes. Based on the entire set of devices in or around the premises and the various flexibility profiles, the EMS should then be able to generate offerings of flexibility.

For the specification described in this document, the assumption is that the EMS is a black box that produces offerings of flexibility. How these offerings are being generated, whether they are aggregations of multiple devices at the premises or which devices are involved is of less importance for this specification. Moreover, it can be used for each of these situations.

3.3 Increasing consumer involvement

As a consequence, the consumer gets more and more involved in energy management within the local household and with the connecting smart grid. Moreover, the European Commission, in the context of the 2020 initiative for an Energy Strategy, established at the end of 2009 a Smart Grid Task Force which identified in its work program that "Consumer empowerment includes capabilities of consumers to have sufficient and timely information on their actual energy consumption, to learn and act upon their energy savings potential through energy usage optimisation and more energy efficient technologies, to have access to competitive offers for energy services and to develop energy efficient consumption practices."

Furthermore, the task force pointed out that the expected advanced communications capabilities of smart grids would enable consumers to exploit real-time electricity pricing and become more active players in the internal market for energy. Smart grid is a rather innovative concept and the market is under development and growing. Specification and standardization of data exchange for smart grids and specifically on the subject of information exchange for end-consumer services will contribute to the efficiency of the market by reducing implementation and maintenance cost. Also, it will reduce customer switching barriers between energy service providers which provide services for active customers. This enables a better position for market players that offer such energy efficiency services to their customers. Furthermore, such a standard will lower the risk of a multiplication of de-facto specifications from different origins resulting in a lack of interoperability and portability of technologies used.

Based on this discussion, we have selected the main scope of the specifications in this document to be the information exchange around flexibility between stakeholders in the electricity domain, specifically between consumer and energy provider, but in principle between all pairs of stakeholders that exchange offerings of flexibility. In Chapter 6, we will describe these roles and their information exchange in more detail. First, we will describe the flexibility concept in the next chapter.

4 Flexibility concept

The flexibility concept assumes that parties connected to the grid produce offerings of flexibility in load and (distributed) generation. Thereby, so-called *flex-offers* are issued indicating these power profile flexibilities, e.g. shifting in time or changing the energy amount. In the flex-offer approach, consumers and producers directly specify their demand and supply power profile flexibility in a fine-grained manner (household and SME level). Flex-offers are dynamically scheduled in near real-time, e.g. in case when the energy production from renewable energy sources, such as wind turbines, deviates from the forecasted production of the energy system.

4.1 The Flex-offer in the energy system

The central concept of our approach is the flex-offer specification. Essentially, a flex-offer is a request for demand or supply of energy with specified flexibilities as shown in Figure 1. The bars represent an electricity profile which is split into six time intervals. The flexibility in time is represented by the minimal and the maximal start time. The white, light grey and dark grey sections of the bars visualize the flexibility of the amount. The given flexibilities enable the scheduling of requests on higher hierarchy levels.

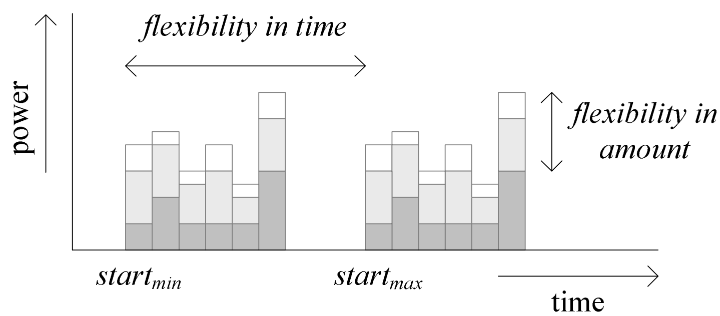

Figure 1: General example of a flex-offer specification.

On the prosumer level and within its EMS, a flex-offer is bound to a device consuming or producing electricity, e.g. a dishwasher, dryer, washing machine, swimming pool pump, electrical heating, heat pump device, charging of an electric vehicle, and combined generation of heat and power. The profile of the flex-offer corresponds to the profile of the device (and its flexibility).

Energy management for consumers and producers is realized at the lowest level of the hierarchy, and it uses functionality either provided by a smart meter or a separate energy management system. From the perspective of metering and data management, we distinguish between demand and supply. The system stores historic data and uses it to forecast demand and supply for the near future in prosumer profiles (i.e., day ahead and intra-day). Prosumers can issue flex-offers usually one day ahead or intra-day, i.e. near real-time.

The Balance Responsible Party (BRP) further aggregates the flex-offers, schedules them depending on several factors like the current market situation, the availability of renewable energy and the energy prices, and negotiates the price, the use and timing of flex-offers with the prosumers. By using schedulable flex-offers, a BRP is able to use more renewable energy, because rescheduling can be performed in near real-time in reaction to the availability of renewable energy.

Our approach allows a BRP to re-schedule requests in a way that (1) the plan is met within the day, i.e., that no imbalances are caused, and (2) options for the shift of demand or supply can be sold to the system operator or traded on a wholesale market. A system operator could use options to shift demand or supply provided by a BRP to stabilize the electricity grid. The benefits for a prosumer could be better prices for electricity (lower price for demand and higher price for supply) and an environmentally conscious behaviour.

4.2 Examples of FlexEnergy Use

This section describes some concrete examples of FlexOffers for various devices. It shows how flexibility can be expressed for electrical vehicle charging, heat pumps and combined heat power systems.

4.2.1 Electric Vehicle Charging

Electric vehicles will typically be used during the day to commute and charged during the night. The charging process is lengthy and consumes a large amount of energy. It would be beneficial for the system operator to have the flexibility to schedule some of the charging load in order to prevent capacity problems on the grid when all vehicles load simultaneously.

An electric vehicle owner can offer this flexibility by expressing the minimal constraints that have to be met. The vehicle is for instance available for charging from 6pm in the evening till 7.30am in the morning. An additional constraint is that the vehicle should be charged to 30% of its capacity as soon as possible (e.g. before 11pm in this example). The reason for this constraint is that it gives the owner the possibility to use his car for emergency situations; a 30% charge would be sufficient to get for instance to the local hospital. The rest of the charging can be allocated at any time during the night as long as the vehicle is fully charged by 7.30am. Figure 2 shows a graphical representation of this FlexOffer.

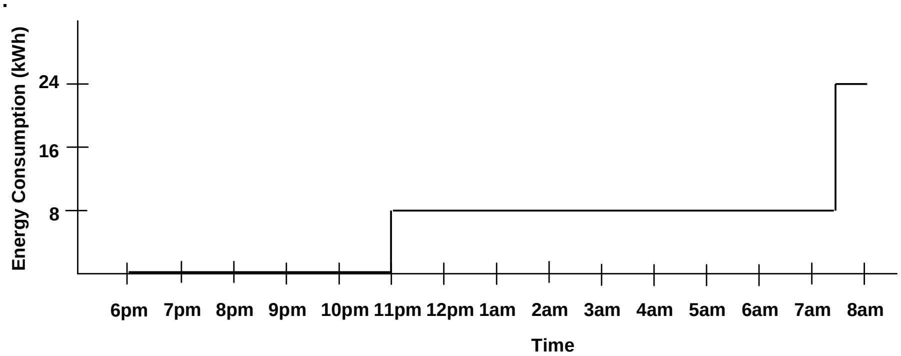

Figure 2: Flexibility constraints in a charging pattern for an electric vehicle.

The figure shows the minimal required energy level for each point in time. Each assignment profile that is above or equal to the minimum level is valid. When the owner of the electric vehicle has to use his car during this time interval, the flex-offer is not met. In that case, there should be (financial) penalty for the owner, e.g. the reduction due the flexibility offered is not granted.

4.2.2 Heat Pump

Heat pumps draw a lot of power from the grid, up to several kilowatts. The power consumption is usually controlled by a thermostat. By intervening between the thermostat and the heat pump it is possible to shift the heat pump's demand. Postponing the operation of a heat pump for a small amount of time (0-15 minutes) can already create a considerable amount of flexibility given the large energy consumption.

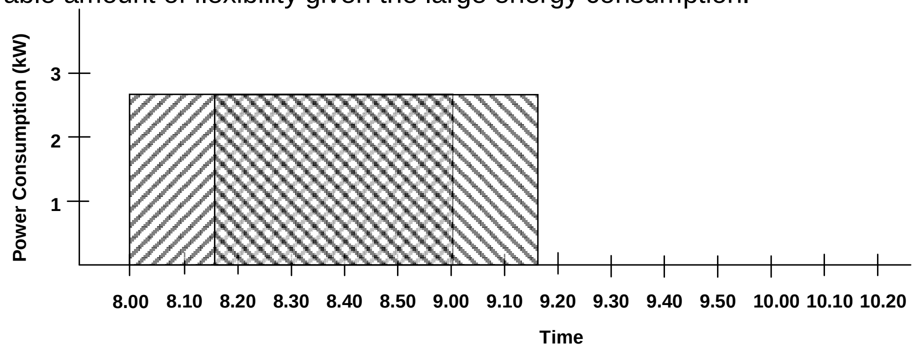

Figure 3: Example of the time shifting potential of a heat pump

Figure 3 shows an example of a flex-offer that makes this flexibility explicit. Please note that the y-axis of this figure represents power and not energy as in the previous electric vehicle charging figure. This figure shows the two extremes; an energy block that starts at 8.00 and ends at 9.00 and another block that runs from 8.15 till 9.15. All other options that start between 8.00 and 8.15 are also valid. The example is a simplification of a real situation in that all blocks have the same duration. In reality a later start time means that the heat pump has to operate for a slightly longer period of time because the start temperature will have dropped a bit further.

4.2.3 Combined Heat and Power System

A Combined Heat and Power (CHP) system consumes gas and produces heat and electricity. Like the heat pump the CHP is also usually temperature driven. The operation of a CHP can be postponed or advanced in the same manner as the heat pump. In addition to the shifting operation in time it is also possible for nearly all CHP's to operate at partial (e.g. 70%) or full power (100%). This can be exploited by offering two flexibility options: (1) to run at 100% producing 1kW for a short amount of time and (2) to run at 70% producing 0,7kW of power running for a longer period of time (in order to reach the target temperature). This is depicted in Figure 4.

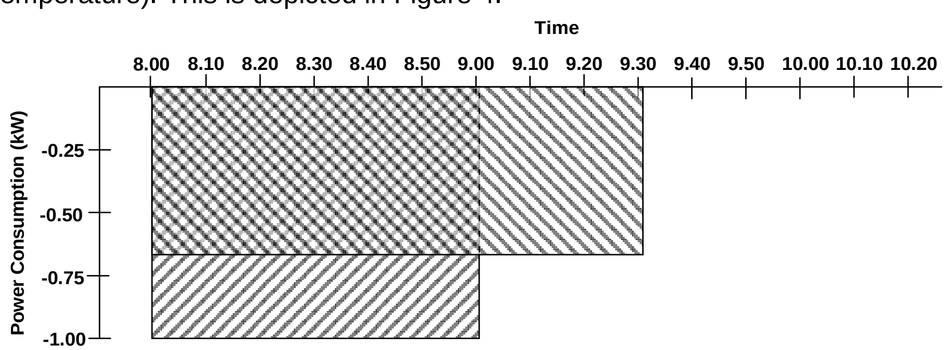

Figure 4: Example of the flexibility of a combined heat and power system.

The CHP produces electricity hence the negative energy consumption. The CHP can also be shifted in time but this figure focuses only on the different power output levels. The surface of the two blocks are more or less equal depicting the fact that the same amount of energy is required to raise the temperature to the required level.

4.3 Business advantages and global incentives

The conceptual and infrastructural approach offers advantages throughout the energy domain. In general, the flex-offer concept increases the ability to balance consumption and production in the electricity system. Currently, the power output of most renewable energy sources (RES, e.g. windmills, photovoltaic sources) is intermittent since it depends on external factors, e.g. wind speed, the amount of sunlight, etc. Hence, available power from RES can be predicted, but not planned. This makes it difficult for energy distributors to include RES into their daily schedules exactly. As an unfortunate consequence, power from RES sometimes has to be traded against very low prices due to a lack of demand.

The flex-offer mechanism provides the ability to adjust the power profile of load and/or distributed generation in order to maintain balance within the system. Forecasting of e.g. weather conditions can be used to predict the production of renewable energy more accurately. As a consequence, the flexibility in energy demand can be used to deal with this uncertainty in renewable energy production by scheduling predicted supply of RES with time intervals of flexible energy demand.

In the end, the net effect of matching demand and supply by scheduling with flexibilities is that the necessity for usage of reserve power due to imbalances, and thus the level of financial consequences, is decreased. This should be a first financial incentive for the electricity suppliers to include flexibilities into the matching of demand and supply. This financial advantage for the electricity suppliers can be partially passed on to the energy consumers to give them the incentive to make use of their flexibility against a lower energy price. On the other hand of the energy chain, the producers of energy have an incentive to produce more RES-based energy, because it can be used more effectively and waste of RES-energy can be avoided more often. Finally, the “business case” or incentive for government to support such a flexibility mechanism is that, as a consequence of the incentives of energy producers, the amount of RES-based energy will increase and (inter)national treaties on energy and promises to decrease the level of greenhouse gasses can be met.

4.4 Existing approaches and comparison

The flex-offer approach is one of many approaches towards demand response management. There are several ways of implementing demand side management. Four different approaches are presented below.

The first approach is direct control by a third party. In this case one or more devices - such as CHP systems or air-conditioning - in a household can be controlled directly by a third party. The customer usually receives a discount for handing over some of the control. An example of this approach is the SmartRate project by PG&E in California. In this project air-conditioners are controlled by energy company PG&E. When deemed necessary air-conditioners are instructed to run in a limited mode that restricts their energy consumption considerably for a period of 15 minutes. This enables PG&E to actively manage the load on the network when faced with capacity problems. In exchange for providing this ability consumers pay less for their energy.

The advantage of this approach is the ability for the energy supplier to exercise a fine grained control over energy demand due to the large amount of devices that it can influence directly. The disadvantage is apparent. Gain of control for the energy supplier is loss of control for the consumer. The consumer no longer has complete freedom to make his own decisions about energy consumption and production in the household. Of course the consumer voluntarily gave up some of his autonomy for a financial reward.

A second (very popular) approach towards demand side management is to use some form of dynamic pricing as an incentive for certain demand-response behaviour. Pricing information is sent to consumers to influence their behaviour in an indirect manner. Consumption is being stimulated with low energy prices and discouraged when prices are high. An example of such an approach is the Bidirectional Energy Management Interface (BEMI) [NesBenRing], where a price profile (consisting of 15 minutes time slots) is sent to the consumer. The price profile is either determined by an energy supplier or generated by a market place. Although the price profile can be interpreted manually by consumers it is much more convenient to use an Energy Management System (EMS) to do this. The advantage of this approach is its simplicity; all it requires is sending a price profile to a consumer, there is no need for two-way communication. The disadvantage is that the consumers are quite passive; they can only react to a price profile but they cannot offer their own production to others. Another disadvantage is that the use of price profiles by energy suppliers makes it difficult to transparently compare the offers of several suppliers.

A third option is to copy the approach taken by energy exchanges. Traditionally energy exchanges trade large volumes of energy and are not accessible to smaller consumers or producers. By enabling trading of energy for smaller volumes as well, an exchange can be an effective means to adapt demand to intermittent supply. An example of this approach is the PowerMatcher [Power]. The PowerMatcher couples intelligent agents with devices such as washing machines, heat pumps, CHP systems, etc. The agents are responsible for buying energy for consuming devices and selling it in case of production. Agents representing a consuming device are willing to pay more for energy when a device really needs to run, e.g. a heat pump that has to operate because of the temperature hitting its lower threshold. The willingness to pay for energy is expressed in a bid curve. The opposite holds for producing devices.

The big advantage of an exchange is its simplicity. It is relatively easy to process all the biddings and to extract an equilibrium price that should balance supply and demand. The exchange approach works best when the “needs” of devices are evenly spread; some devices just consumed energy and can do without for a while, some are willing to consume energy when the price is right and some devices find themselves in a must-run situation and are willing to pay any price. However there are cases where such a spread of device states is not very likely. Consider the charging of electric vehicles that return home in the evening and need to be recharged at 7 am in the morning. Their agents will all behave in similar ways. They will first wait and see how prices on the exchange develop, then at some point they will reach a must-run state because of the lengthy load process. This may very well give rise to capacity problems because all the loading is concentrated at the second half of the night, while the first half was spent waiting. These situations are disadvantageous for the exchange approach.

Finally, the flex-offer approach differs from the other three approaches in that participants explicitly specify how much flexibility (both consumption and production wise) they are willing to offer to other parties in the market. These other parties may operate intermittent energy sources and could exploit flexible demand to direct energy consumption to those moments in time their sources produce energy in order to maintain a better balance. In this case one is willing to pay for the ability to shift energy (in addition to volume based prices). The advantage of this approach is that it combines the possibility for fine grained control (as with the third party control approach) by the party that buys flexibility with the full autonomy (that is also maintained by the price profile and exchange approaches) for the party that sells it. Revisiting the electric vehicle charging example; owners of these vehicles offer the consumption of the energy needed for a recharge to be flexibly shifted over the entire night. A network operator can take advantage of these flex-offers by assigning part of the charges to the first half of the night and the other part to the second half thereby actively preventing capacity bottlenecks. A disadvantage of the flex-offer solution is that it can be a quite complex problem to satisfy all flexibility constraints. Especially when large numbers of flex-offers need to be processed and all flex-offers contain all types of flexibility constraints.

The common denominator among all approaches is that they try to exploit flexibility that is present in consumption and/or generation devices. This flexibility is used to counter the intermittent character of renewable energy sources.

5 Modeling approach

Before setting up our specification, we have elaborated on various data modeling techniques that can be used for the purpose of modeling actors, roles, processes, information and messages [KonRum10]. In addition, existing data models in the energy sector have been investigated to determine which models can be reused. We concluded that the Common Information Model [CIM09] of the International Electro-technical Commission and ebIX models [EMD09, ebiXCuS] form a good basis for our flexibility model. In addition, we decided to adopt UN/CEFACT's Modeling Methodology (UMM) as the modeling technique of choice. Table 1 depicts an overview of the models, modeling techniques that are applied in this project.

Type	Purpose	Technique	Chapter
Behavior	Roles	UML Actor diagram	6
	Process	UML Message exchange diagram	6
Structure	Data model	UML Class diagram	7
Message	Message model	UMM Message model	8
	XML mapping	XML	9

Table 1: Modelling techniques used in the flexibility specification.

UML Use case diagramming is used to describe the actors in the domain and their functional relationships. UML Message exchange diagram is used to define the processes that are executed between the various roles. UML Class diagramming is used for the description of the data model. Although a data/information model is not part of UN/CEFACT's Modeling Methodology (UMM) it is a requirement for a good modeling practice to develop this first.

The data model serves as a solid basis from which UMM artifacts such as Business Entity Views can be derived. By having the data model as common ground the consistency between the various Business Entity Views is ensured.

In some cases it is not possible to describe certain aspects of the data models with diagramming techniques alone. This is especially true for constraints. These are expressed using Object Constraint Language (OCL), which is part of UML.

Please note that within the data model specified, the derived attribute construct is used. An attribute name preceded by a slash denotes that this is a derived attribute that is determined by combining other fixed attributes. The choice for a derived attribute is made when it is deemed convenient for the user of the data model to have the information explicitly available.

In order to keep the data model readable, associations between different classes are labeled. The UML navigability construct is used to indicate the direction of the labels; it does not impose any limits on the structure of the data model.

The naming conventions used in the data model are as follows:
- package names must be in lower case;
- class names must be in camel case[1], first letter in upper case;
- field and method names must be in camel case, first letter in lower case.

When names from the data model are used, they are represented as follows: `EnergyAmount`.

UMM is used as the methodology to describe the message model. This entails describing a business process first followed by a description of the messages that are exchanged. These messages are called “`BusinessEntity`”.

6 Role and process model

This chapter describes the roles in the energy domain that are part of the scope of the specification in this document. Besides that, the processes that are of interest between these roles as well as the messages exchanged between the roles involved are described. Thereby, the focus is on the producer/consumer (prosumer), the balance supplier and the balance responsible party. The basis for this roles description is formed by the ENTSO-E harmonized role model, that is used to introduce the various roles in the energy domain that are most interesting for our specification. In addition, we introduce a role abstraction towards the level of providers and acquirers of flexibility. This abstraction can be mapped onto various pairs of roles, such as consumer and balance responsible party, but also balance responsible party and balance supplier. Finally, we describe on which subset of the messages this specification focuses, i.e. on the flex-offer, flex-offer acceptance and flex-offer assignment messages.

6.1 ENTSO-E model

The three major standardization or sector organizations on the European market are ENTSO-E (European Transmission System Operators), ebIX (European forum for energy Business Information eXchange), and EFET (European Federation of Energy Traders) [EFET]. One of them, ENTSO-E, started preparing a coherent model of the electricity market in Europe, as a prerequisite to a feasible concept which could be gradually put

[1] Camel case is the practice of writing compound words or phrases in which the elements are joined without spaces, with each element's initial letter capitalized within the compound and the first letter is either upper or lower case (source: Wikipedia).

into practice. This model has been termed the “Harmonized role model”, but it comprehensively addresses on the one hand organization and structuring of players and on the other hand the processes which (should) constitute the electricity market and the processes which are necessary to assure the operational capability of the electricity grids in the new circumstances.

Prompted by European directives, the evolution of the Harmonized role model has been accompanied by gradual diffusion into national role models and national regulations covering the organization of national electricity markets. In 2009, the year of the current issue of the model, the model has been harmonized on lower levels but not yet on top level(s). The work is being continued and further evolution of the model is envisaged. Since other stakeholders from the major industry players and other configurations of stakeholders are also active, e.g. European Electricity Grid Initiative [EEGI10], we may foresee additional incentives towards further modifications of the model.

The model is conceptually inclusive, i.e. a union of all accepted national/regional models. For this reason there are two consequences:

1. On the lower levels the structuring of roles exceeds the actual structuring of roles in individual countries. Effectively this means that the roles in individual national/regional models may integrate several roles of the Harmonized role model.
2. On the upper level(s), specifics of the markets in major countries (Germany, etc.) and regions (e.g. Nordic) are recognized and not harmonized; also the market or trading aspect of the market is not fully defined on this level as yet.

The basic representation of the Harmonized role model focuses on organizational aspects of the market (roles and domains). Figure 5 shows a condensed view on the ENTSO-E model; the relevant actors for the flex-offer concept in the network are depicted.

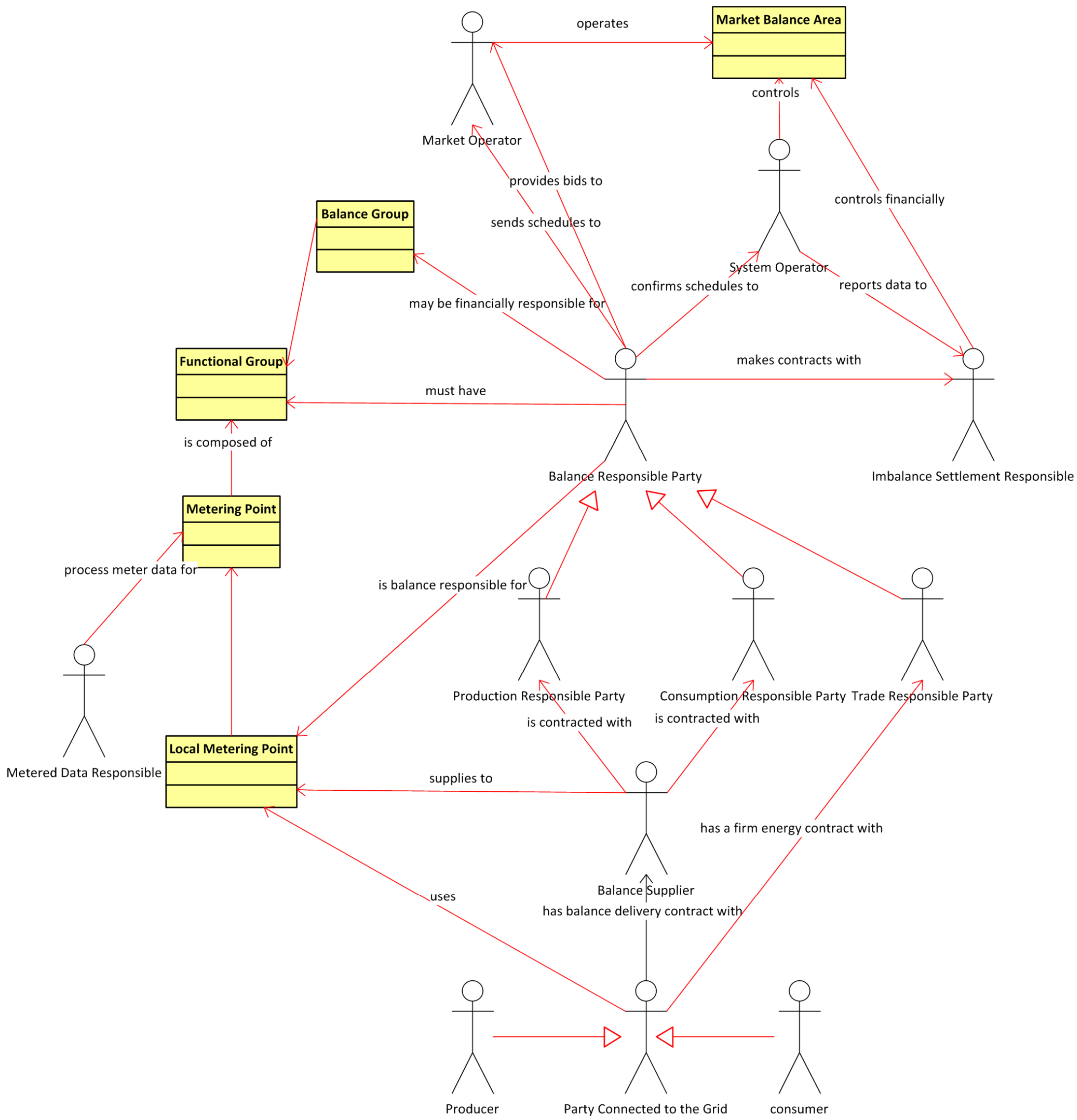

Figure 5: Condensed view on ENTSO-E Harmonized Role model.

In Figure 6, a message exchange diagram between the roles identified in Figure 5 is shown. This figure is deducted from the ENTSO-E harmonized role model and forms the basic roles of the specifications defined in this document. The roles send messages (outgoing arrows) and receive messages (incoming arrows). The specification is focused on two main roles: Balance Responsible Party (BRP) and Balance Supplier. Four other roles surround these main roles and have interaction with them: Prosumer (Party connected to the grid), System Operator, Market Operator and Metered Data Responsible. Between each pair of roles, a specific process is being executed together with the exchange of various messages.

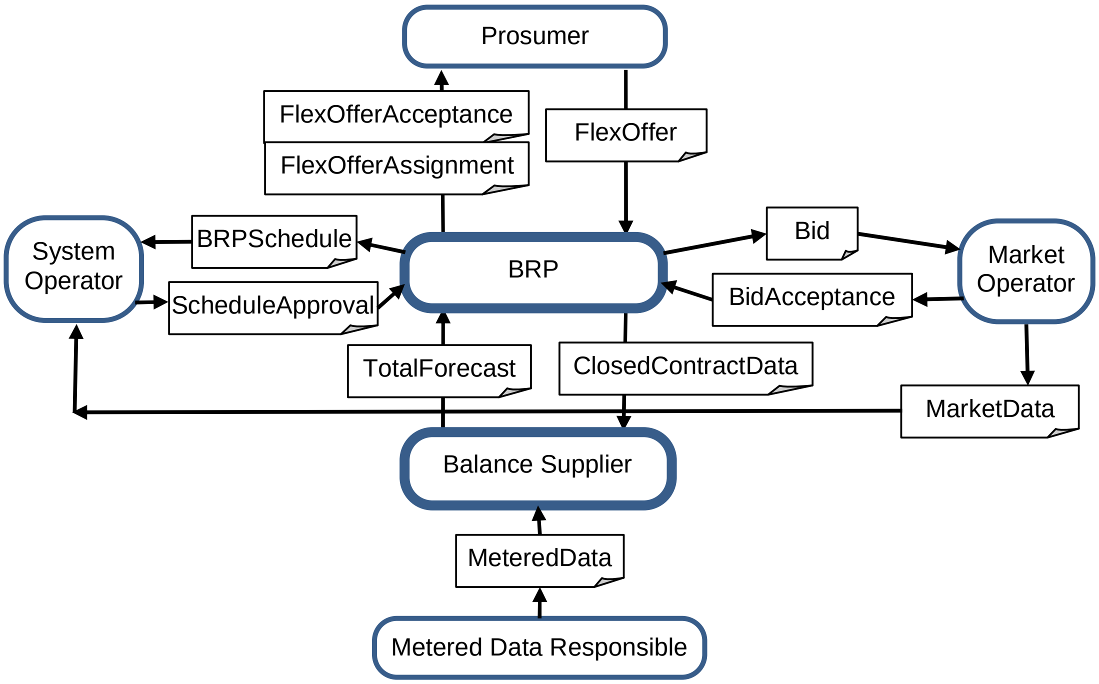

Figure 6: Messages exchange diagram between the main roles.

This specification focuses on the process and messages between the Prosumer and the BRP. The main reason for this focus is that all the other processes are already common practice in the energy domain. Also, most of the other messages are either already specified or even standardized by ENTSO-E, ebIX, EFET or standardization organizations. Although these processes and messages are not in the scope of this document, we shortly describe them in order to provide some more context.

6.2 Generation, acceptance and assignment of flex-offers

In this section, the main process between the Prosumer and the BRP is described. In that process, the messages FlexOffer, FlexOfferAcceptance and FlexOfferAssignment are the specific messages that form the interaction model.

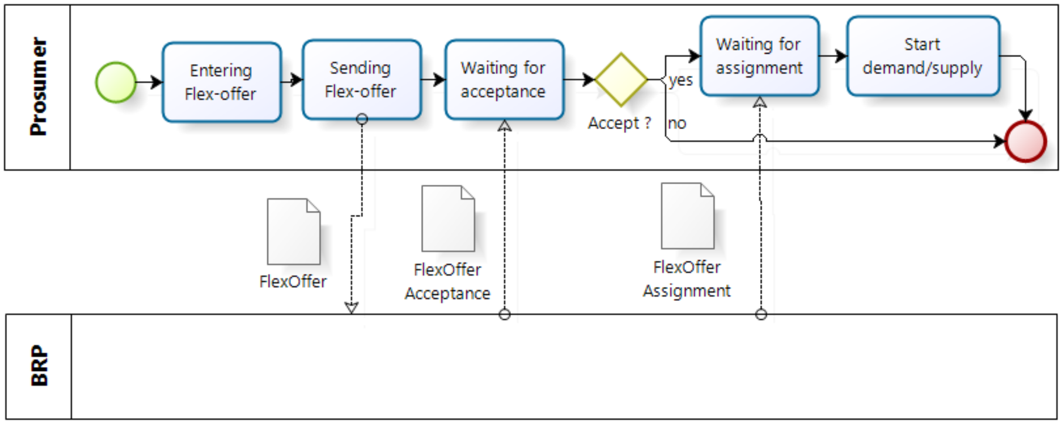

Figure 7: Generation of flex-offers.

The process starts by creating a new flex-offer and specifying its main parameters like start after time (i.e. earliest operation start time) and start before time (i.e. latest operation start time). After that the prosumer sends it to the BRP and waits for acceptance or rejection notification, which must be received not later than `acceptBefore` time specified in the flex-offer. If the flex-offer is rejected, the process finishes. If it is accepted, the prosumer waits for assignment information which must be received not later than `assignBefore` parameter of the flex-offer. After the assignment is received, the prosumer can start real consumption or production according to the schedule.

6.3 Forecast management

In the present state of the Harmonized role model, the Balance Supplier's function is to manage open contracts for the BRP. The business policy for defining the open contract price does and has to originate from the BRP, as a parameter in managing the closed contract policy. Without it, the BRP would have no degree of freedom in balancing its closed contract energy flow (preferred for influencing the dynamics and consequentially the RES handling) as opposed to open contract energy flow.

We see the following reasons for setting up the BS as a separate role:

- There are in principle several BS's per one BRP, which means considerable work but also geographical spread of physical operations of a BRP.
- Looking into a probable future, the forecast of total metered consumption is part of the BS: but it is done for the BRP and the results of the forecast are used by the BRP; and the message flow has to reflect it. We furthermore assume that there can be multiple BS reporting its forecasts to one BRP.

The most important functions of the balance supplier are collecting metered data and producing forecasts on open contract consumption. In particular, open contract price will be provided as a parameter of the BRP.

Forecast management (Figure 8) mainly includes the creation, usage and maintenance of so-called forecast models.

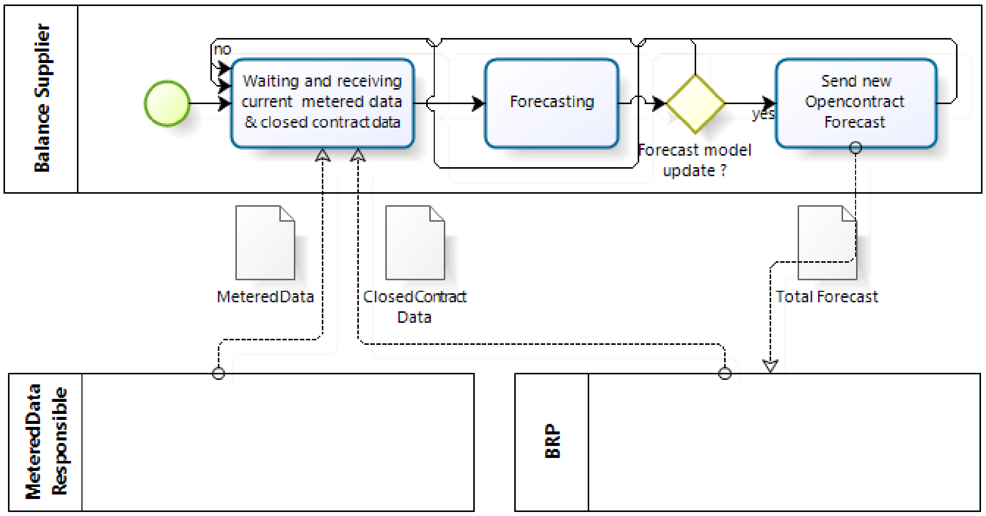

Figure 8: Forecast management.

Initially, a forecast model is created with respect to one or many time series. This model is then used in order to obtain the forecast values (for a specific forecast horizon) of the underlying time series. Furthermore, for new measurements the forecast model is updated by the control component. Internally, this forecast model update evaluates the old forecast model according to the new measurements and triggers adaptation (maintenance) of the forecast model if required. In case of an adapted forecast model, new forecast values are available with regard to the specified forecast horizon.

The forecast management process runs continuously. After a certain time interval new metered data arrives. Closed contract data resulting from accepted and/or assigned flex-offers and containing all information about contracted flex offer production/consumption energy (flex-contract) arrive periodically. Closed contract data is used for composite forecast models which allow incorporating multiple time series into one model.

There are several approaches to forecasting. Forecasting can either be done by Balance Suppliers only (Balance Supplier Forecasting) or it can be done by both Balance Suppliers and its BRP (Distributed Forecasting).

- *Balance Supplier Forecasting.* BRP receives forecasting values from BS and aggregates them to the overall forecast.
- *Distributed Forecasting.* Microscopic forecast management of BRP is essentially very similar to forecasting management of BS. However, due to the higher level within the market hierarchy, several alternative approaches could be applied with regard to the macroscopic system architecture. Examples are (1) Naive Exchange All, (2) Exchange Forecast, (3) Autonomous Forecasting, and (4) Autonomous Forecasting (Synchronized). In the case of Autonomous Forecasting, BRP receives aggregated metered data from all its Balance Suppliers and does its own forecasting. In this case the forecasting models are updated in a distributed fashion.

6.4 Flex-offer aggregation, scheduling and usage

All accepted flex-offers are stored at the BRP (Figure 9). A continuously running procedure aggregates these flex-offers, then schedules them and finally disaggregates them. This process results in individual flex-offer assignments which are sent to prosumers. Aggregated flex-offers can be rescheduled as long as they have not reached the deadline until when they have to be assigned (`assignmentBefore`). When this time is reached, they are classified as “not re-assignable”. In case of flexible energy products, aggregated flex-offers or flex-bids can get the status “not internally re-assignable” if they have been traded successfully. In this case, the Market Operator schedules them.

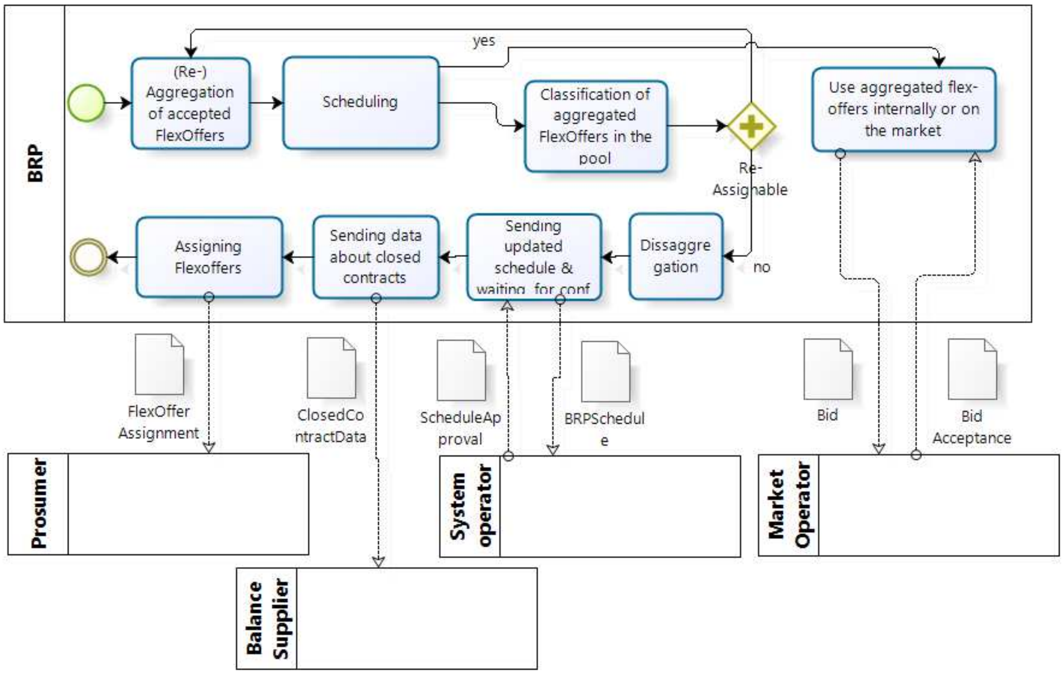

Figure 9: Flex-offer aggregation, scheduling and usage.

6.5 Issuers and acquirers of flexibility

As mentioned before, the specification in this document focusses on the process of generation, acceptance and assignment of flex-offers and the messages exchanged within this process between the involved roles. This process is defined in section 6.2 specifically between the Prosumer and the Balance Responsible Party. Nevertheless, this process can in principle also be executed between other pairs of roles.

The flex-offer approach is aimed to be applied within a multi-actor context. Any actor which has the ability to control load or (distributed) generation resources is capable of issuing an offer for the flexibility in these resources to other actors. These actors acquiring the offered flexibility may provide compensation for such offerings.

Both the functionalities of *issuer* and *acquirer* of flexibility can be mapped on the ENTSO-E harmonized role model, which is depicted in Figure 5 and Figure 6. In Table **2** the roles from the ENTSO-E harmonized model are presented as actors either issuing or acquiring flexibility. A *party connected to the grid* from the ENTSO-E model can issue flexibility in the electricity demand. In this case the BRP acquires the flexibility. On the other hand, a *party connected to the grid* can also acquire flexibility to get a good match its electricity supply. In that case, the BRP issues flexibility and the party connected to the grid acquires flexibility. Also other pairs of roles are possible in the view of issuer-acquirer of flexibility as shown in Table **2**.

Issuer of flexibility	Acquirer of flexibility
Party connected to the grid: consumer	BRP
BRP	Party connected to the grid: producer
BRP	Balance supplier
BRP	Market operator
BRP	System operator

Table 2: Issuers and acquirer of flexibility; mapping on ENTSO-E role model.

Therefore, in the remainder of this document we will use the general terms issuer and acquirer of flexibility. Figure 10 provides a schematic view on the functionalities of acquirers and issuers of flexibility in load and/or generation. In general, any number of issuers of flexibility can interact with any number of acquirers of flexibility. However, both the technical as well as the commercial setting may limit the number of issuers and/or acquirers. In this context, an arbitrary number of parties connected to the grid (many issuers) offer flexibilities to their balance responsible party (the single acquirer).

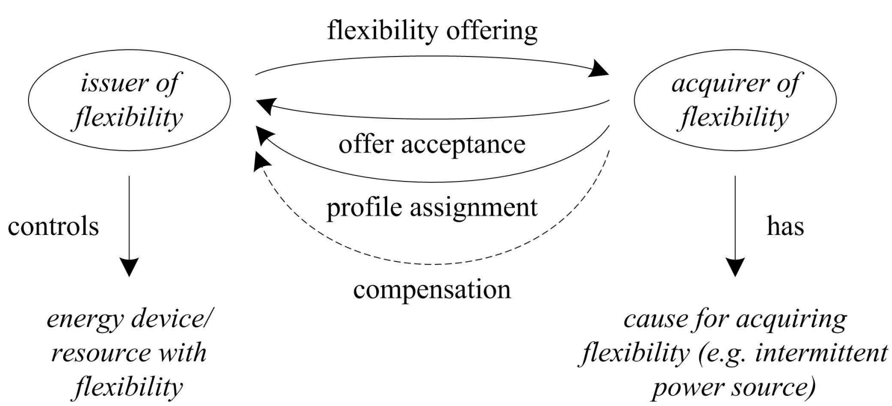

Figure 10: Schematic view of the issuer and acquirer actors.

Issuers of flexibility control one or more devices that demand energy or resources that supply energy. This control is executed either directly or indirectly and allows control of the power profiles of these devices and resources. The issuers decide what flexibility is offered; based on e.g. technical, financial and/or comfort grounds. Thus, the flexibility-issuer remains autonomous in its decision making.

Acquirers of flexibility have a use for the ability to control the power profiles of these devices and resources as offered by the flexibility-issuers. This includes e.g. the ability for a balance responsible party to achieve the schedules submitted to a system operator as in the ENTSO-E scheduling system with a substantial amount of intermittent energy sources in their portfolio.

Offerings of flexibility need to be accepted (or rejected) by the acquiring party. When flexibility offered is accepted, a profile assignment must be provided by the acquiring party to indicate the desired behaviour. This profile assignment must comply with the limits of the flexibility offered; e.g. no higher power output then offered, no temporal shift beyond the temporal bounds offered, etc.

In the next chapter, we describe the generic data model that is used as the common ground for the messages to be exchanged in the flex-offer generation, acceptance and assignment process between issuers and acquirers of flexibility.

7 Generic data model

This section describes the data model used for expressing flexibility. First in section 7.1 an introduction to the model is provided, indicating the scope of the model. Sections 7.2 through 7.4 provide specification of the model. Finally section 7.5 provides a number of examples for use of the model.

7.1 Model introduction

The data model that is presented here describes different possibilities of shifting energy demand or supply. The model is centered on the `FlexEnergy` concept (see section 7.3). It features various options to describe such flexibility both in an energetic and a financial way. It should be noted that it is highly unlikely for specific instances to use all of the available options. In most cases only one way for expressing flexibility is used.

Throughout the model several data types are used that are derived from the Common Information Model by the IEC, such as `RealEnergy`.

7.2 Time series related supporting classes

This paragraph defines a number of utility classes which are used throughout the data model specified in this deliverable for expressing time series related concepts.

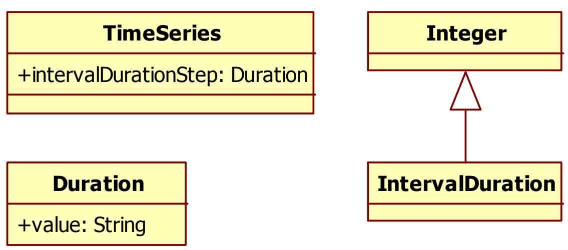

Figure 11: Supporting classes.

`TimeSeries`		
Description:	This is a parent class from which more specific `TimeSeries` classes can be derived.	
Attributes:	`interval DurationStep`	A `TimeSeries` consists of a series of intervals. The `intervalDurationStep` attribute indicates the step size of the length of such an interval. The length of an interval should be a multiple of `intervalDurationStep` (see the `EnergyConstraintInterval` explanation on page 27 for more details).

IntervalDuration	
Description:	Expresses a multitude (as an integer) of the `intervalDurationStep` attribute. This interval duration must be referred to when using this class; e.g. elements in a `TimeSeries` refer to the `intervalDurationStep` attribute of that class.

Duration		
Description:	Duration represents a period of time. The value space of duration is a six-dimensional space where the coordinates designate the Gregorian year, month, day, hour, minute, and second components defined in § 5.5.3.2 of [ISO8601].	
Attributes:	`Value`	The lexical representation for duration is the [ISO 8601] extended format `PnYnMnDTnHnMnS`, where `nY` represents the number of years, `nM` the number of months, `nD` the number of days, 'T' is the date/time separator, `nH` the number of hours, `nM` the number of minutes and `nS` the number of seconds. The number of seconds can include decimal digits to arbitrary precision.

7.3 Classes for negotiating flexibility in energy supply and demand

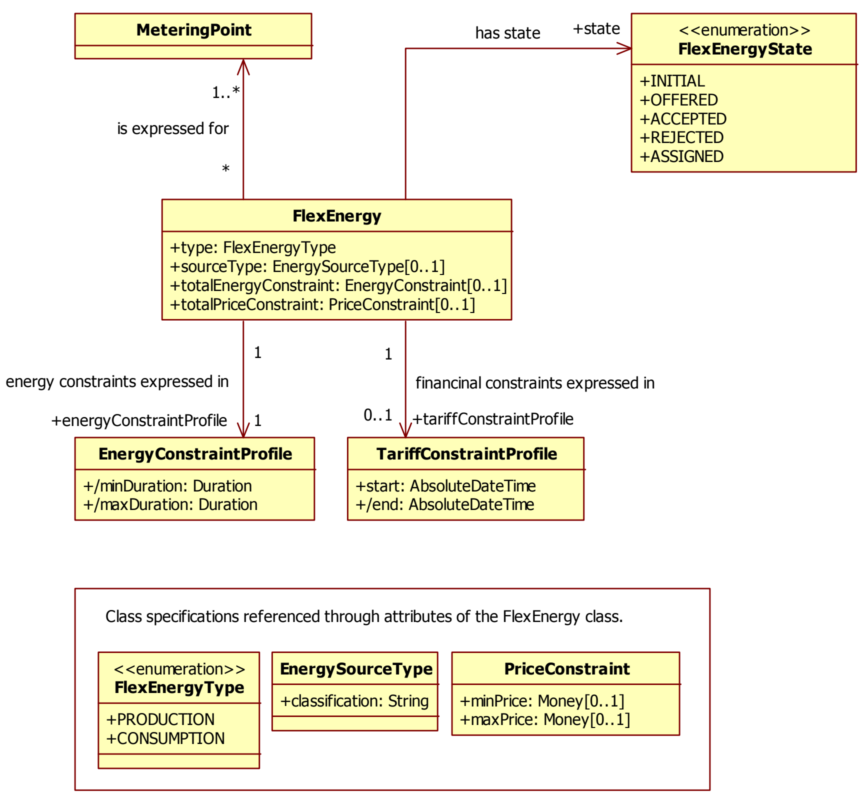

Figure 12: `FlexEnergy`.

FlexEnergy		
Description:	This class is the central class to describe the flexibility in energy. It is associated with other classes that express constraints in terms of time, energy and costs. Time shifts can be expressed through the `EnergyConstraintProfile` class. This class also serves as the basis for a flexible energy profile. Financial aspects can be expressed with the `TariffConstraintProfile` and the `PriceConstraint` classes. The flexibility expressed in `FlexEnergy` objects refers to one `MeteringPoint` instance or multiple instances if the `FlexEnergy` object reflects aggregated flexibility of these metering points.	
Attributes:	`type`	Indicates whether the `FlexEnergy` object relates to either production or consumption of electricity.
	`sourceType`	This attribute indicates, in case of production, the classification of the electricity source. The following types can be distinguished: base load, controlled, peaker, RES.
	`totalEnergy Constraint`	With this attribute limits can be put on the overall amount of energy that is expressed in a `FlexEnergy` object. The attribute is an array of the type `EnergyConstraint`, its definition can be found on page 28.
	`totalPrice Constraint`	This attribute limits the total monetary amount to be paid or earned for the electricity consumed or produced related to this `FlexEnergy` object. Note a maximum price is used in case of consumption and a minimum price is used in case of production.

FlexEnergyState		
Description:	This enumeration contains the valid states a `FlexEnergy` object can be in. Figure 13 provides a graphical view of the states of a `FlexEnergy` object.	
Enumeration literals:	`INITIAL`	The initial state of a `FlexEnergy` object. The object has been constructed but not yet offered.
	`OFFERED`	The `FlexEnergy` object is offered; i.e. a `FlexOffer` (c.f. section 8.1) object is constructed and associated with the `FlexEnergy` object and is communicated.
	`ACCEPTED`	The `FlexEnergy` object is accepted; the flexibilities expressed in this `FlexEnergy` object will be used.
	`REJECTED`	The `FlexEnergy` object is rejected; the flexibilities expressed in this `FlexEnergy` object will not be used.
	`ASSIGNED`	This state indicates that a `FlexEnergySchedule` object is associated with this `FlexEnergy` object

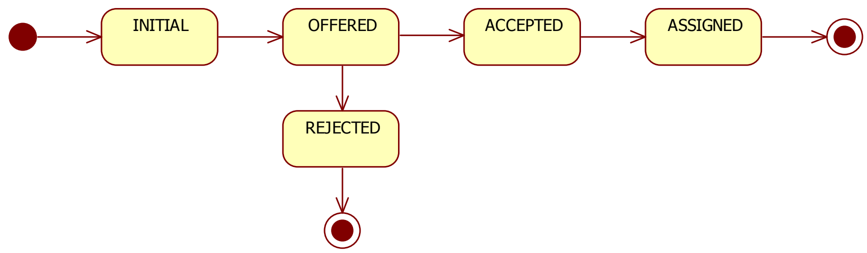

Figure 13: `FlexEnergyState`.

`FlexEnergyType`		
Description:	This class is used to indicate whether a FlexOffer is about producing or consuming electricity	
Enumeration literals:	`PRODUCTION`	This literal indicates a FlexOffer that describes flexibilities with regard to the production of electricity.
	`CONSUMPTION`	This literal indicates a FlexOffer that describes flexibilities with regard to the consumption of electricity.

`EnergySourceType`		
Description:	The EnergySourceType class can be used to indicate the source that produces electricity, e.g. photovoltaic or wind. This is useful when regulations or policies are in place that treat particular energy sources differently.	
Attributes:	`classification`	This attribute is used to indicate the energy source. It is a String, which can have one of the following values: base load, controlled, peaker, RES. It is expected that the individual RES types will need to be further specified.

`PriceConstraint`		
Description:	This class is optionally associated with the `FlexEnergy` class. It can be used to express an overall constraint to a `FlexEnergy` object on the price paid or earned for the energy exchanged in association to the `FlexEnergy` object.	
Attributes:	`minPrice`	The minimum price for the exchanged energy associated with the `FlexEnergy` object; only applicable in case of production.
	`maxPrice`	The maximum price for the exchanged energy associated with the `FlexEnergy` object; only applicable in case of consumption.

`MeteringPoint`

Description:	A point in the grid to which production and/or consumption is connected. C.f. the modeling and description of `MeteringPoint` in [ENTS09].

<table>
<tr><th colspan="3"><code>EnergyConstraintProfile</code></th></tr>
<tr><td>Description:</td><td colspan="2">The <code>EnergyConstraintProfile</code> can be used to describe an energy profile. It is associated with an ordered sequence of <code>EnergyConstraintIntervals</code>. <code>EnergyConstraintProfile</code> is derived from the <code>TimeSeries</code> class as can be seen in the diagram in Figure 14.</td></tr>
<tr><td rowspan="2">Attributes:</td><td><code>/minDuration</code></td><td>A derived attribute indicating the minimum duration of an <code>EnergyConstraintProfile</code>. This minimum duration is based on the minimum duration of the <code>EnergyConstraintInterval</code> objects referred to by the <code>EnergyConstraintProfile</code>. This duration can be calculated with the following algorithm:

<pre>minDuration = 0
for i in energyConstraintProfile.intervals {
	if i.minDuration->empty() == false then
		minDuration += i.minDuration *
			energyConstraintProfile
			.intervalDurationStep
	endif
}</pre></td></tr>
<tr><td><code>/maxDuration</code></td><td>A derived attribute indicating the maximum duration of an <code>EnergyConstraintProfile</code>. This maximum duration is based on the <code>startAfter</code> attribute of the first <code>EnergyConstraintInterval</code> and the <code>endBefore</code> of the last interval:

<pre>maxDuration =
  self.intervals->last().endBefore –
 self.intervals->first().startAfter</pre></td></tr>
<tr><td>Constraints:</td><td colspan="2"><pre>self.intervals->first().startAfter->notEmpty()
and self.intervals->last().endBefore->notEmpty()</pre>
This constraint entails that the first interval must specify <code>startAfter</code> and the last interval must specify <code>endBefore</code>.</td></tr>
</table>

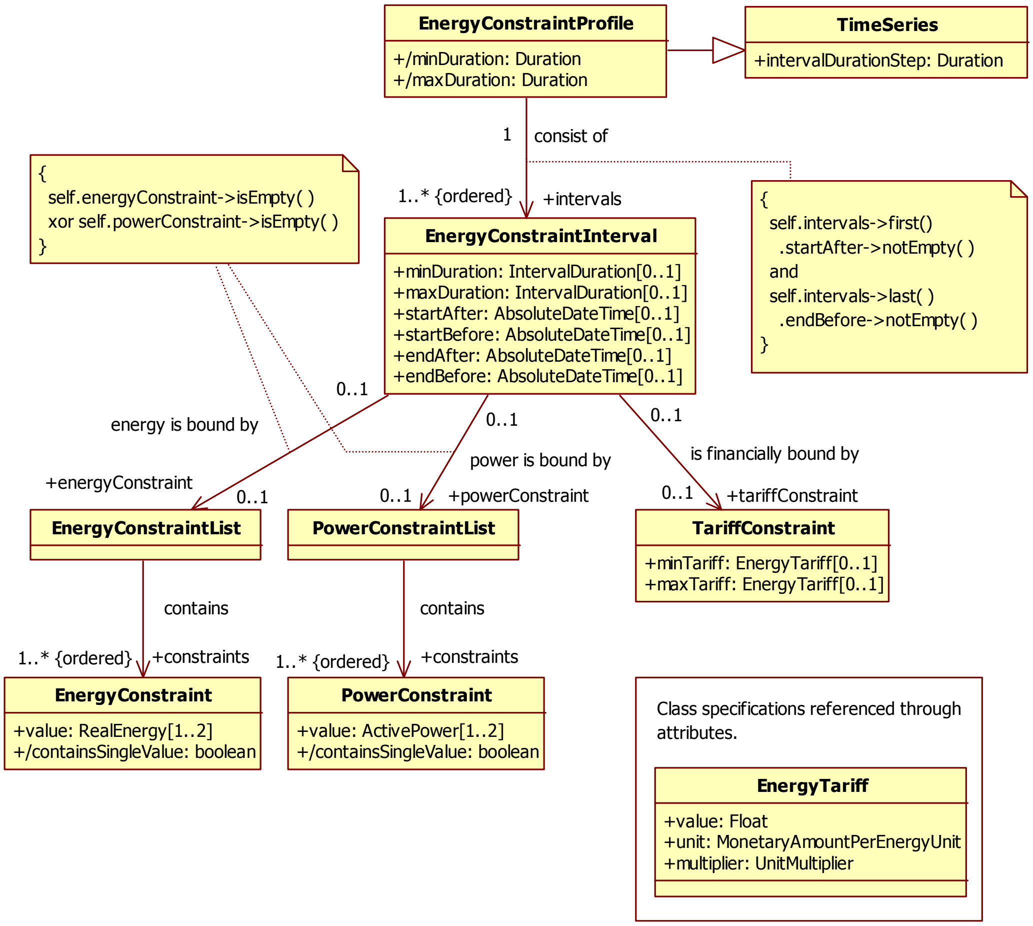

Figure 14: EnergyConstraintProfile.

EnergyConstraintInterval		
Description:	The `EnergyConstraintInterval` represents an interval of time that is part of an `EnergyConstraintProfile`. The duration of an `EnergyConstraintInterval` may be flexible and is expressed through the `minDuration` and `maxDuration` attributes.	
Attributes:	`minDuration`	This optional attribute represents the shortest possible duration for this `EnergyConstraintInterval`. The attribute is of the `IntervalDuration` type, which is based on an integer, and expresses a multitude of the `intervalDurationStep` attribute that is defined in the `TimeSeries` class.
	`maxDuration`	This optional attribute represents the longest possible duration for this `EnergyConstraintInterval`. The attribute is of the `IntervalDuration` type, which is based on

		an integer, and expresses a multitude of the `intervalDurationStep` attribute that is defined in the `TimeSeries` class.
	`startAfter`	Optionally defines the *earliest* time (inclusive) for the *start* of this `EnergyConstraintInterval`.
	`startBefore`	Optionally defines the *latest* time (inclusive) for the *start* of this `EnergyConstraintInterval`.
	`endAfter`	Optionally defines the *earliest* time (inclusive) for the *end* of this `EnergyConstraintInterval`.
	`endBefore`	Optionally defines the *latest* time (inclusive) for the *end* of this `EnergyConstraintInterval`.

Constraints:	`self.energyConstraint->isEmpty( )`
	`xor self.powerConstraint->isEmpty( )`
	
	This constraint entails that for this interval either power or energy consumed or produced in this interval is to be constrained.	

`EnergyConstraintList`	
Description:	Instances of this class aggregate a list of `EnergyConstraint` instances. An `EnergyConstraint` is either a fixed value or an upper and lower bound. With this list structure combinations can be made, e.g.: *1 kWh* *1 kWh* or *2 kWh* or *3 kWh* between *1* and *3 kWh* *1 kWh* or between *2* and *3 kWh*

`EnergyConstraint`		
Description:	Describes a constraint on an amount of energy consumed or provided. If one value is provided this value is the only allowable value (unless contained in an `EnergyConstraintList` instance with multiple elements). If two values are provided, this indicates a lower and upper bound.	
Attributes:	`value`	The values as described above.
	`/containsSingleValue`	This boolean is a derived attribute that indicates whether a single value is provided (true) or whether 2 values have been provided (false) indicating a lower and upper bound.
Constraints:	`self.value[0] < self.value[1]` In case two values have been provided (referring to a lower and an upper bound) the above mentioned constraint applies. It specifies that the first value provided should be smaller (lower bound) than the second value (upper bound).	

`PowerConstraintList`	
Description:	Instances of this class aggregate a list of `PowerConstraint` instances. A `PowerConstraint` is either a fixed value or an upper and lower bound. With this list structure combinations can be made, e.g.: *1 kW* *1 kW* or *2 kW* or *3 kW* between *1* and *3 kW* *1 kW* or *2* between *3 kW*

`PowerConstraint`		
Description:	Describes a constraint on an amount of power consumed or provided. If one value is provided this value is the only allowable value (unless contained in a `PowerConstraintList` instance with multiple elements). If two values are provided, this indicates an upper and lower bound.	
Attributes:	`value`	The values as described above.
	`/containsSingleValue`	This boolean is a derived attribute that indicates whether a single value is provided (true) or whether 2 values have been provided (false) indicating a lower and upper bound.
Constraints:	`self.value[0] < self.value[1]` In case two values have been provided (referring to a lower and an upper bound) the above mentioned constraint applies. It specifies that the first value provided should be smaller (lower bound) than the second value (upper bound).	

`TariffConstraint`		
Description:	This class expresses tariff constraints that are related to the `EnergyConstraintInterval` it is associated with. Either a minimum or maximum tariff may be given; a minimum in case of production, a maximum in case of consumption.	
Attributes:	`minTariff`	This is the minimum tariff (price per unit of energy) that one is willing to receive when selling energy during the associated `EnergyConstraintInterval`.
	`maxTariff`	This is the maximum tariff (price per unit of energy) that one is willing to pay when buying energy during the associated `EnergyConstraintInterval`.

EnergyTariff		
Description:	`EnergyTariff` expresses a monetary amount in units of currency per quantity of electrical energy to be paid in case of consumption or is earned in case of production. This class exactly follows the CIM class `CostPerEnergyUnit`, but is renamed for readability.	
Attributes:	`Value`	A float value that expresses the cost.
	`Unit`	This refers to the currency that is being used and is of the type `MonetaryAmountPerEnergyUnit`; it expresses e.g. Euros, dollars, etc.
	`multiplier`	This attribute is of the type `UnitMultiplier`. An example would be k Euros (1000 Euros) or using the none multiplier indicating that the unit should not be multiplied.

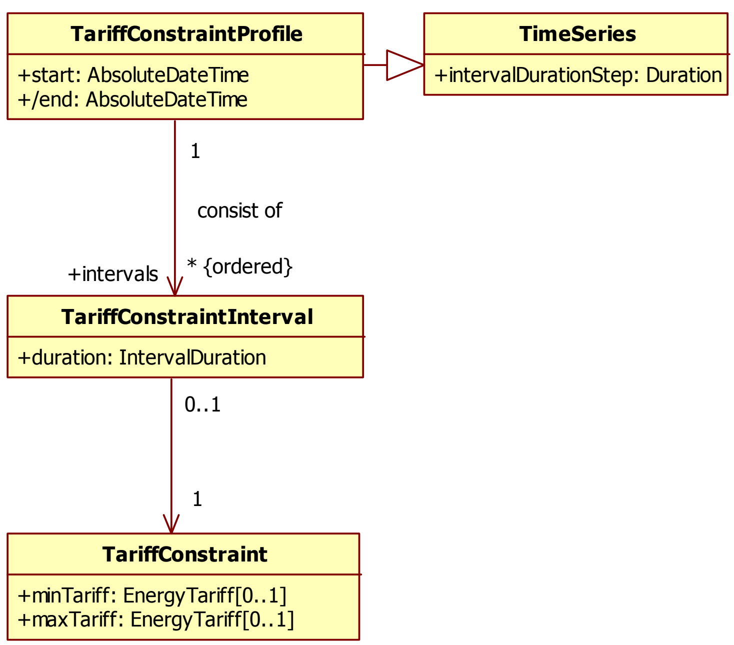

Figure 15: `TariffConstraintProfile`.

`TariffConstraintProfile`		
Description:	The `TariffConstraintProfile` is a way of expressing financial constraints for the `FlexEnergy` structure. Here the financial constraints have their own `TimeSeries` independent from the `EnergyConstraintProfile`. This way one can express the willingness to shift an instance of an `EnergyConstraintProfile` to a particular point in time. Considering the case of consumption; by specifying a low maximum price for buying energy in a certain interval it will become very unlikely that energy consumption is shifted to that interval. The duration of a `TariffConstraintProfile` instance must be equal to the maximum length of the `EnergyConstraintProfile` or longer; c.f. section 7.4. *Please note that the `TariffConstraintProfile` can never be used in combination with a `TariffConstraint` on an `EnergyConstraintInterval` (see Figure 14) as this may lead to conflicting constraints. Only one of these options is allowed in a specific instance.*	
Attributes:	`Start`	This is the starting time of the `TariffConstraintProfile`

	`/end`	This is the end time of the `TariffConstraintProfile`; this attribute is derived from the durations of the `TariffConstraintInterval` objects referred to.
		
		`end = 0;`
		`foreach(interval in this.intervals) {`
		`      end += interval.duration;`
		`}`

`TariffConstraintInterval`		
Description:	The `TariffConstraintInterval` represents an interval of time that is part of a `TariffConstraintProfile`. The duration of these intervals is of fixed length (in contrast to the `EnergyConstraintProfile`).	
Attributes:	`Duration`	This attribute expresses the duration for this `TariffConstraintInterval`. The attribute is of the `IntervalDuration` type, which is based on an integer, and expresses a multitude of the `intervalDurationStep` attribute that is defined in the `TimeSeries` class.

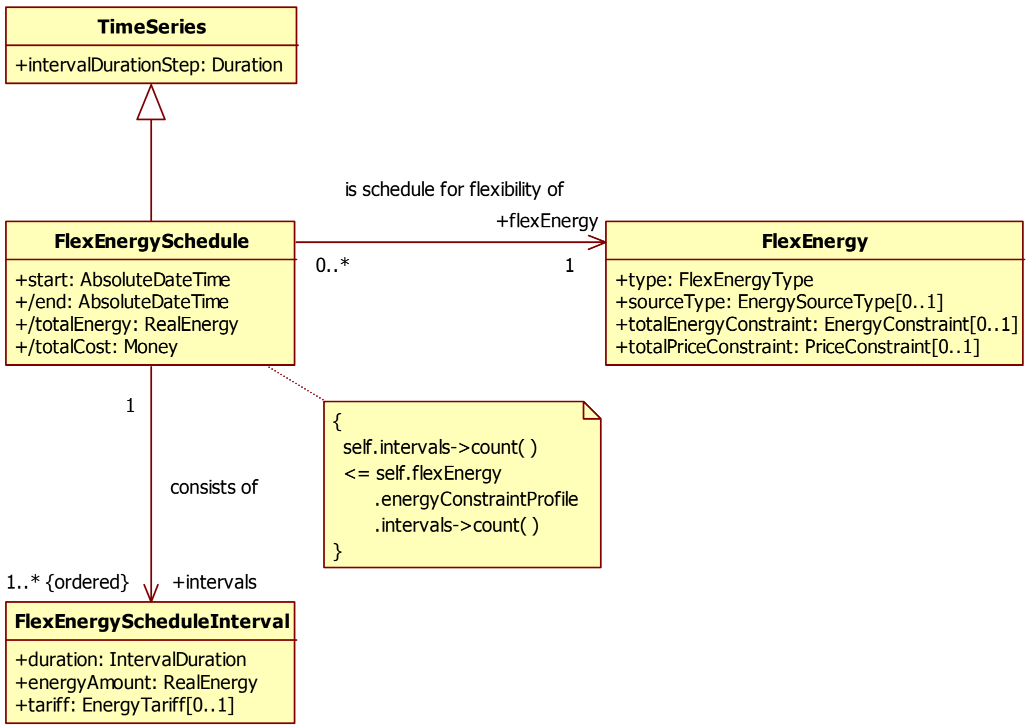

Figure 16: `FlexEnergySchedule`.

`FlexEnergySchedule`		
Description:	The `FlexEnergySchedule` class is derived from `TimeSeries` and is associated with 1 or more `ScheduleInterval` classes. This structure is used to express a fixed energy profile. Please note that the constraint in Figure 16 indicates that a `FlexEnergySchedule` must consist of equal or less intervals than in the intervals in the `EnergyConstraintInterval` of a `FlexEnergy` object.	
Attributes:	`start`	The point in time at which the flex-offer schedule starts.

	`/end`	This calculated attribute represents the end point of this flex-offer schedule.
		
		`end = self.start;`
		`foreach(interval in this.intervals) {`
		`       end += interval.duration;`
		`}`

<table>
<tr><td rowspan="2"></td><td><code>/totalEnergy</code></td><td>This is total amount of energy that is expressed through this flex-offer schedule.

<pre>totalEnergy = 0;
foreach(interval in this.intervals) {
      totalEnergy += interval.energyAmount;
}</pre></td></tr>
<tr><td><code>/totalCost</code></td><td>The total amount of money that is expressed through this flex-offer schedule.

<pre>totalCost = 0;
foreach(interval in this.intervals) {
      totalCost += (interval.energyAmount *
                   interval.tariff);
}</pre></td></tr>
<tr><td>Constraints:</td><td colspan="2"><code>self.intervals->count( )</code>
<code><= self.flexEnergy.energyConstraintProfile.intervals->count( )</code>

The number of intervals for <code>FlexEnergySchedule</code> is equal or less than the number of intervals in <code>FlexEnergy</code>. The number of intervals is less in case the period of one or more intervals that originally were in <code>FlexEnergy</code> is set to a length of zero. The zero length intervals will not be present in the <code>FlexEnergySchedule</code> since they have no meaning.</td></tr>
</table>

<table>
<tr><th colspan="3"><code>ScheduleInterval</code></th></tr>
<tr><td>Description:</td><td colspan="2">The <code>ScheduleInterval</code> represents an interval of time that is part of a <code>FlexEnergySchedule</code>.</td></tr>
<tr><td rowspan="3">Attributes:</td><td><code>Duration</code></td><td>This attribute expresses the duration for this <code>ScheduleInterval</code>. The attribute is of the <code>IntervalDuration</code> type, which is based on an integer, and expresses a multitude of the <code>intervalDurationStep</code> attribute that is defined in the <code>TimeSeries</code> class. Please note that the <code>intervalDurationStep</code> of a <code>ScheduleInterval</code> object must have the same value as <code>intervalDurationStep</code> that is used in the corresponding <code>EnergyConstraintProfile</code>.</td></tr>
<tr><td><code>energyAmount</code></td><td>The amount of energy that is associated with this <code>ScheduleInterval</code>.</td></tr>
<tr><td><code>tariff</code></td><td>The tariff that is associated with this <code>ScheduleInterval</code>.</td></tr>
</table>

7.4 Temporal constraints

The data structure that describes all the concepts that are related to `FlexEnergy` contains several distinct points in time. These points in time are interdependent. The temporal relation between `TariffConstraintProfile` and `EnergyConstraintProfile` is expressed in the formula below where `self` is an instance of `FlexEnergy`.

```
     self.tariffConstraintProfile.start
=    self.energyConstraintProfile.intervals->first( ).startAfter
≤    self.energyConstraintProfile.intervals->first( ).startBefore
≤    self.energyConstraintProfile.intervals->last( ).endAfter
≤    self.energyConstraintProfile.intervals->last( ).endBefore
=    self.tariffConstraintProfile.end( )
```

Note: this timeline expresses the fact that the `TariffConstraintProfile` should completely envelop the interval that is the maximal interval potentially covered by the `EnergyConstraintProfile`.

Note: `startBefore` of the first interval of the `energyConstraintProfile` and `endAfter` of the last interval are optional (see also Figure 14).

The `FlexEnergySchedules` expressed relate to the temporal bounds expressed in the related `EnergyConstraintInterval` as specified in the formula below where `self` is an instance of `FlexEnergySchedule`.

```
     self.flexEnergy.energyConstraintProfile.intervals->first( ).startAfter
≤    self.intervals->first( ).start
≤    self.intervals->last( ).end
≤    self.flexEnergy.energyConstraintProfile.intervals->last( ).endBefore
```

The intervals in a `FlexEnergySchedule` and an `EnergyConstraintProfile` are consecutive which means that the end time of one interval is the start time of the next interval.

7.5 Examples usage of expressions of flexibility in `FlexEnergy`

This section describes some concrete examples of `FlexEnergy` for various devices. It shows how flexibility can be expressed for heat pumps, combined heat and power systems and electrical vehicle charging.

7.5.1 Electric Vehicle charging

The `FlexEnergy.totalEnergyBounds` constraint can be used to express the upper and lower bound of the energy in a `FlexEnergy` object. E.g. a `FlexEnergy` object expressing a profile with several `EnergyConstraintIntervals` with flexibility in the amount of energy per interval results in a variable total amount of energy for the `FlexEnergy` object.

Such a construct can be used to model the bounds for charging an electric vehicle. Figure 17 shows such a profile consisting of an interval *a* of fixed length and fixed power, an interval *b* of fixed length but variable power, and an interval *c* of variable length (and thus variable amount of energy) and variable power (thus also of variable amount of energy).

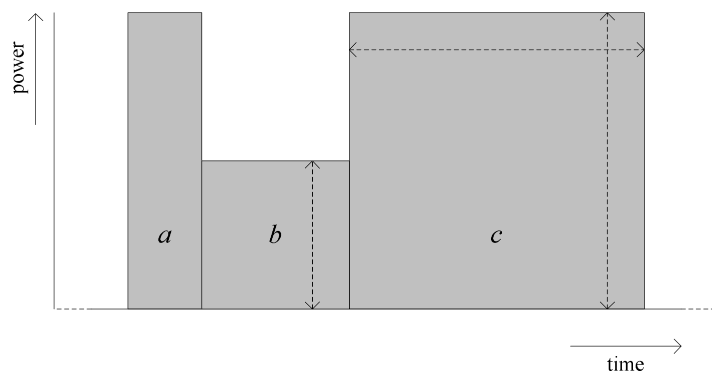

Figure 17 Flex-offer graph for electric vehicle charging.

The first interval ensures that a certain minimal state of charge is reached in an initial interval to e.g. ensure a certain level of comfort. The second interval in this example can be caused by a priority for other consumption. For the final interval this restriction is no longer of influence and the maximum amount of power can be used, and this interval has a variable duration.

Formulas 1 through 6 are example constraints on the charging pattern for an electric vehicle. They express the upper and lower bounds of the duration and power consumption, indicated with a subscript d and p respectively, per interval. The total energy which can be consumed according to these constraints lies between 1.5 kWh and 10.5 kWh. These constraints can be expressed with `EnergyConstraintInterval.energyConstraint`.

$$15 \text{ min.} \leq a_d \leq 15 \text{ min.} \quad (1)$$
$$6 \text{ kW} \leq a_p \leq 6 \text{ kW} \quad (2)$$

$$60 \text{ min.} \leq b_d \leq 60 \text{ min.} \quad (3)$$
$$0 \text{ kW} \leq b_p \leq 3 \text{ kW} \quad (4)$$

$$0 \text{ min.} \leq c_d \leq 60 \text{ min.} \quad (5)$$
$$0 \text{ kW} \leq c_p \leq 6 \text{ kW} \quad (6)$$

Equation 7 presents a constraint on the total amount of energy consumed. This is an additional constraint with respect to those formulated in expressions 1 through 6, which ensures that the desired state of charge will finally be reached. This constraint can be expressed with `FlexEnergy.totalEnergyBounds`.

$$6 \text{ kWh} \leq a_d \times a_p + b_d \times b_p + c_d \times c_p \leq 6 \text{ kWh} \quad (7)$$

7.5.2 Heat Pump

The power consumption of a heat pump is usually controlled by a thermostat. By intervening between the thermostat and the heat pump it is possible to shift the heat pump's demand. Postponing the operation of a heat pump for a small amount of time (0-15 minutes) can already create a considerable amount of flexibility given the large energy consumption. Figure 18 shows an example of a flex-offer that makes this flexibility explicit. It is the flex-offer graph for the time-shifting potential depicted in Figure 3.

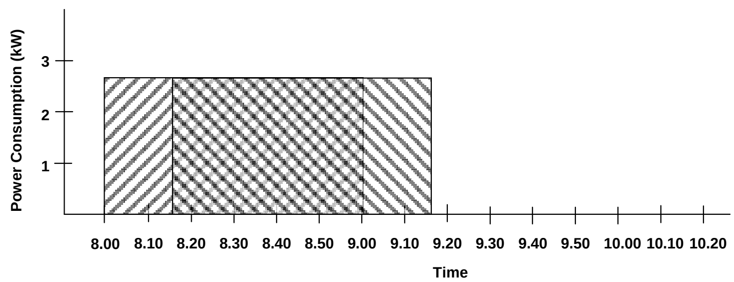

Figure 18: Flex-offer graph for a heat pump.

The constraints depicted in the figure can also expressed in formulas. For this we use variable a to denote the block of energy, subscripts d, p and s are used to express duration, power and start time respectively (all of which can be expressed through `EnergyConstraintInterval`).

$60 \text{ min.} \leq a_d \leq 60 \text{ min.}$ (8)
$2.5 \text{ kW} \leq a_p \leq 2.5 \text{ kW}$ (9)
$8.00 \text{ hr} \leq a_s \leq 8.15 \text{ hr}$ (10)

7.5.3 Combined Heat and Power System

Also for the Combined Heat and Power (CHP) example described in section 4.2.3, we can express the flexibility constraints depicted in Figure 19 in formulas.

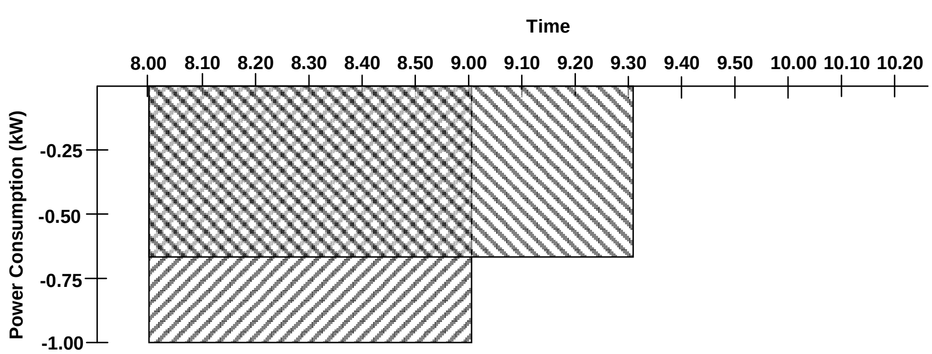

Figure 19: Flex-offer graph for a CHP.

The CHP produces electricity hence the negative energy consumption. The CHP can also be shifted in time but this figure focuses only on the different power output levels. The surface of the two options for this block are more or less equal depicting the fact that the same amount of energy is required to raise the temperature to the required level.

Again we will use variable a to denote the energy block and subscripts d for duration, p for power and s for start time.

a_p = 1 kW xor a_p = 700 W (11)
a_s = 8.00 hr (12)
$a_p \times a_d$ = 1 kWh (13)

8 Message model

This chapter builds upon the data model that is specified in Chapter 7. It focuses on the data that needs to be exchanged by players that want to offer and acquire FlexEnergy.

8.1 Role level message model introduction

There is a relation between the processes that are specified in Chapter 6 and the business process specified here. The latter only focuses on the exchange of messages between swimming lanes. The internal processes of a swimming lane are not specified. This also means that the process that is the subject of this chapter can be mapped onto multiple processes in Chapter 6 as long as these processes use the same messages (`FlexOffer`, `FlexOfferAcceptance` and `FlexOfferAssignment`). To reflect this, the actors that play a part in this business process have been labeled with generic names: ”`FlexEnergy` Issuer” and ”`FlexEnergy` Acquirer”.

8.2 FlexOffer Business Process

The main idea behind the business process that is specified here is that `FlexEnergy` can be offered to another party by means of a `FlexOffer`. This process covers the steps from issuing a `FlexOffer` to receiving an assignment. It focuses solely on the interaction of information between the `FlexEnergy` Issuer and the `FlexEnergy` Acquirer, e.g. how FlexOffers are scheduled is not part of this process description.

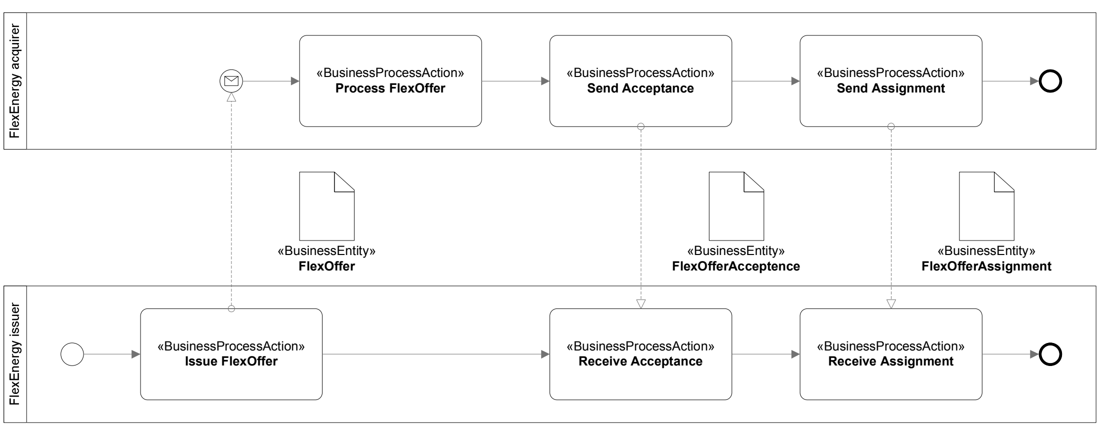

Figure 20: FlexOffer business process.

The terms `FlexEnergy` Issuer and `FlexEnergy` Acquirer where chosen to support multiple levels of aggregation. Whether the interaction takes place between a Party Connected to the grid and a BRP or between two BRP’s the process remains the same.

The first step in the business process is undertaken by the Issuer by issuing the `FlexOffer`. It is then received by the Acquirer which in turn processes the `FlexOffer`. The nature of this processing is out of scope for this business process.

During this processing a decision will be made whether the FlexOffer will be used or not. The outcome of this decision is communicated to the Issuer by the Acquirer in the form of a `FlexOfferAcceptance BusinessEntity`. This `BusinessEntity` only contains a confirmation or a rejection of the FlexOffer.

In the case that the FlexOffer is confirmed a `FlexOfferAssignment BusinessEntity` will follow later. This `BusinessEntity` contains the choices that were made by the Acquirer within the boundaries that were stated in the original FlexOffer. The `FlexOfferAcceptance` and `FlexOfferAssignment` entities may also be combined into a single message.

8.3 FlexOffer

This section specifies the message for offering flexibility as indicated in Figure 21.

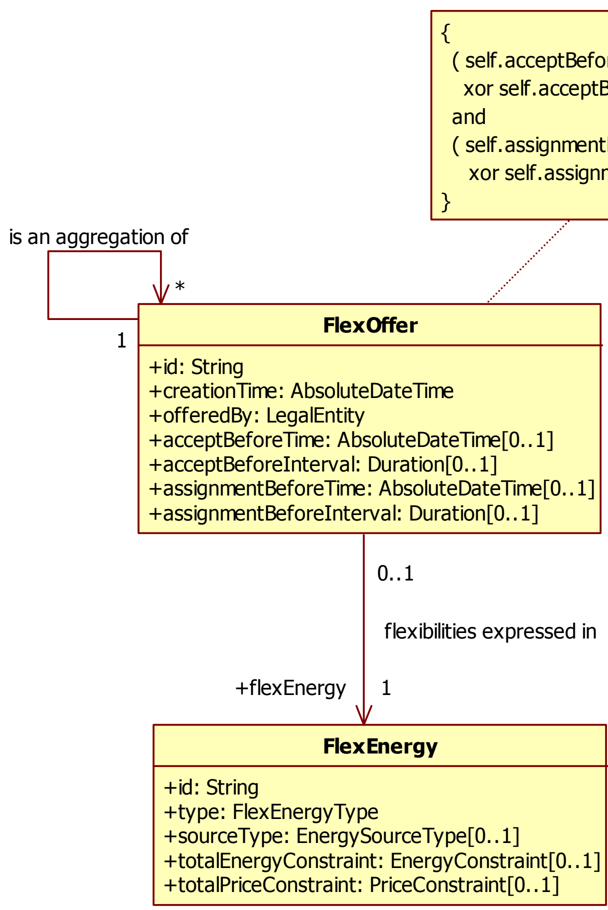

Figure 21: `FlexOffer BusinessEntityView`.

<table>
<tr><th colspan="3">FlexOffer</th></tr>
<tr><td>Description:</td><td colspan="2">Through a FlexOffer <code>FlexEnergy</code> can be offered on a marketplace. A <code>FlexOffer</code> is always associated with a <code>LegalEntity</code> that is responsible for it. <code>FlexOffers</code> can also be aggregated into a new <code>FlexOffer</code> as is shown by the self-association.</td></tr>
<tr><td rowspan="6">Attributes:</td><td><code>id</code></td><td>A unique identifier for the <code>FlexOffer</code>. The selection of an identifier scheme is beyond the scope of this document as it is up to the user of this model to make these choices when implementing the model.</td></tr>
<tr><td><code>offeredBy</code></td><td>This attribute represents the <code>LegalEntity</code> that is responsible for issuing this <code>FlexOffer</code>.</td></tr>
<tr><td><code>acceptBeforeTime</code></td><td>An absolute moment in time by which the <code>FlexOffer</code> is to be accepted, i.e. the Issuer needs to know whether the <code>FlexOffer</code> is going to be exerted or not. The Acquirer notifies the Issuer by sending a <code>FlexOfferAcceptance</code> message (see 8.5).</td></tr>
<tr><td><code>acceptBefore Interval</code></td><td>This attribute expresses the same deadline as the <code>acceptBeforeTime</code> attribute, but instead of specifying an absolute point in time, an interval is expressed. This interval is the amount of time before the start of operations (as expressed in <code>FlexEnergySchedule.start</code>) before which acceptance of the FlexOffer is to be received by the Issuer.</td></tr>
<tr><td><code>assignmentBefore Time</code></td><td>A moment in time by which the Issuer needs to know what the actual assignment is. This assignment should always respect the boundaries that were specified in the <code>FlexOffer</code>. The <code>acceptBefore</code> and <code>assignmentBefore</code> moment may coincide. The assignment cannot be changed or revoked after the <code>assignmentBefore</code> moment has passed. As with <code>acceptBeforeTime</code>, here an absolute point in time is specified.</td></tr>
<tr><td><code>assignmentBefore Interval</code></td><td>This attribute defines the same deadline as <code>assignmentBeforeTime</code> but as a deadline relative to the start of operations (as expressed in <code>FlexEnergySchedule.start</code>) instead of as an absolute point in time.</td></tr>
<tr><td>Constraints</td><td colspan="2">This constraint expresses that accept and assignment deadlines are specified as absolute or relative points in time.

<code>(self.acceptBeforeTime.isEmpty( )</code>
<code> xor self.acceptBeforeInterval.isEmpty( ))</code>
<code>and</code>
<code>(self.assignmentBeforeTime.isEmpty( )</code>
<code> xor self.assignmentBeforeInterval.isEmpty( ))</code></td></tr>
</table>

`LegalEntity`	
Description:	The `LegalEntity` class represents the entity that can issue a `FlexOffer` and accepts all resulting legal responsibilities. The same goes for `FlexOfferAcceptance` and `FlexOfferAssignment`.

8.4 FlexOfferAcceptance

The message for accepting offered flexibility is specified in this section.

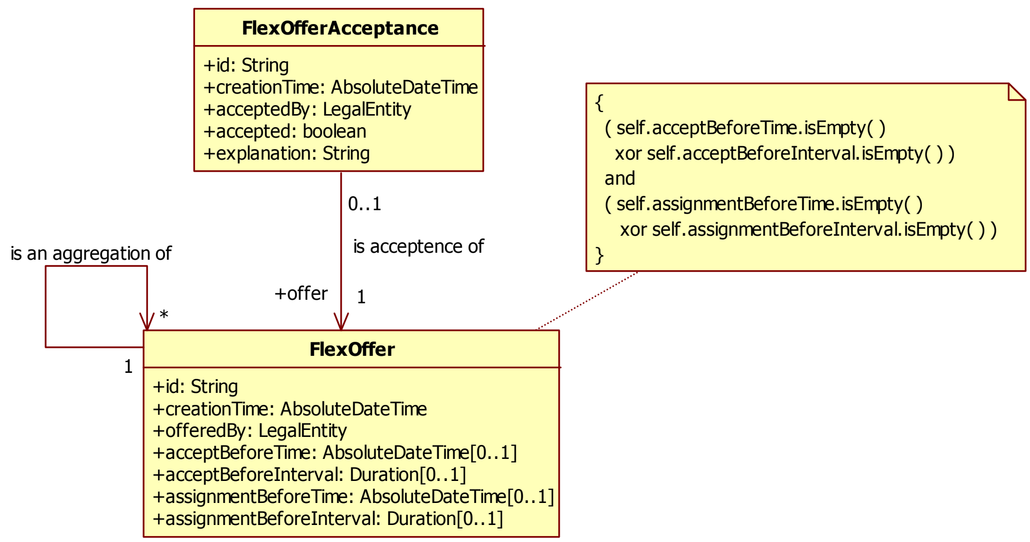

Figure 22: `FlexOfferAcceptance BusinessEntityView`.

`FlexOfferAcceptance`		
Description:	This class contains information about whether a `FlexOffer` has been accepted by the `FlexOffer` Acquirer or not. Therefore it can only be associated with a single `FlexOffer`. Acceptance of a `FlexOffer` only signifies that the Acquirer of the `FlexOffer` is going to make use of it at some point in time within the time flexibility offered. The actual assignment may be postponed to a later point in time.	
Attributes:	`id`	A unique identifier for the `FlexOfferAcceptance`, establishing a unique relation to “its” FlexOffer. The selection of an identifier scheme is beyond the scope of this document as it is up to the user of this model to make these choices when implementing the model.
	`acceptedBy`	This attribute represents the `LegalEntity` that is responsible for issuing this `FlexOfferAcceptance`.
	`accepted`	A Boolean that is true in case the FlexOffer has been

		accepted and false in case the FlexOffer has been rejected.
	`explanation`	This attribute might be used to provide some background on the acceptance or rejection of the FlexOffer.

8.5 `FlexOfferAssignment`

The assignment of `FlexEnergySchedules` for accepted `FlexOffers` is specified in this section.

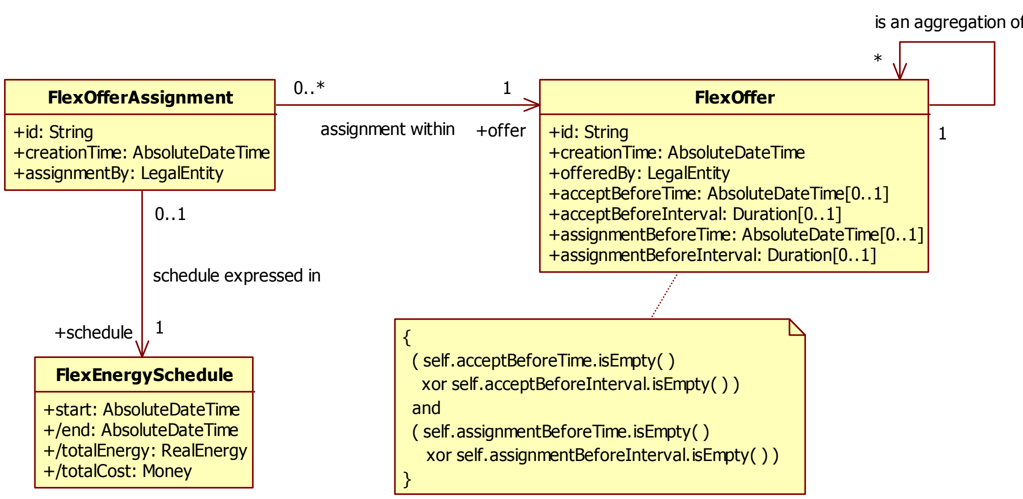

Figure 23: `FlexOfferAssignment BusinessEntityView`.

`FlexOfferAssignment`		
Description:	The `FlexOfferAssignment` is related to a FlexOffer. For each level of flexibility (time, energy, costs) that is offered through the FlexOffer a fixed choice is made (expressed via an instance of the associated `FlexEnergySchedule`) so that the Issuer of the FlexOffer knows what to do. An assignment is always associated with a single `FlexOffer`.	
Attributes:	`Id`	A unique identifier for the `FlexOfferAssignment` establishing a unique relation to “its” FlexOffer. The selection of an identifier scheme is beyond the scope of this document as it is up to the user of this model to make these choices when implementing the model.
	`creationTime`	The moment in time the `FlexOfferAssignment` was created.
	`assignedBy`	This attribute represents the `LegalEntity` that is responsible for issuing this `FlexOfferAssignment`.

8.6 Temporal constraints

This section indicates the temporal constraints which apply to `FlexOffer`, `FlexOfferAcceptance`, `FlexOfferAssignment and FlexEnergySchedule`. They are specified below; the constraints are preceded by the definitions of `absoluteAcceptBeforeTime` and `absoluteAssignmentBeforeTime` as these concepts play an important role in the constraints.

absoluteAcceptBeforeTime = FlexOffer.acceptBeforeTime
xor
absoluteAcceptBeforeTime = FlexEnergySchedule.start - FlexOffer.acceptBeforeInterval

absoluteAssignmentBeforeTime = FlexOffer.assignmentBeforeTime
xor
absoluteAssignmentBeforeTime = FlexEnergySchedule.start - FlexOffer.assignmentBeforeInterval

FlexOffer.creationTime
< FlexOfferAcceptance.creationTime
< absoluteAcceptBeforeTime
≤ FlexOfferAssignment.creationTime
< absoluteAssignmentBeforeTime
≤ FlexEnergySchedule.start
< FlexEnergySchedule.end

This timeline describes the order which all the events related to issuing, accepting and assigning a FlexOffer should adhere to.

9 XML Schema mapping

This chapter provides a mapping of the model specified in Chapter 7 and the messages specified in Chapter 8 into an XML schema. First the strategy for derivation of the XML schema is discussed; in the subsequent sections the schemas are presented.

9.1 Schema derivation strategy

The XML schema mapping consists of two separate schemas, one for the model and one for the messages which are based on the model. Each schema has http://mirabel-project.eu/schemas/ as a prefix.

Each class in the model and message specifications is translated into a XML schema type with exactly the same name. The attributes of the classes are translated into elements of sequences in complex types. In case of class attributes where exactly one of two attributes should have a value, the XML schema choice construct is applied. For ordered relations, the sequence construct is used.

Independent element specifications are only provided in the messages schema for the three messages defined in section 9.3.

9.2 Model schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://mirabel-project.eu/schemas/model"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cim="http://iec.ch/TC57/2001/CIM-schema-cim10"
  attributeFormDefault="qualified"
  elementFormDefault="qualified"
  targetNamespace="http://mirabel-project.eu/schemas/model"
>

  <!-- Schema imports -->

  <import
    namespace="http://iec.ch/TC57/2001/CIM-schema-cim10"
    schemaLocation="cim-schema-cim10.xsd"
  />


  <!-- Type definitions -->

  <complexType name="FlexEnergy">
    <sequence>
      <element name="meteringPointID" type="string" />
      <element name="type" type="tns:FlexEnergyType" />
      <element name="sourceType" type="tns:EnergySourceType" minOccurs="0" />
      <element name="totalEnergyConstraint" type="tns:EnergyConstraint" minOccurs="0" />
      <element name="totalPriceConstraint" type="tns:PriceConstraint" minOccurs="0" />
      <element name="energyConstraintProfile" type="tns:EnergyConstraintProfile" />
      <element name="tariffConstraintProfile" type="tns:TariffConstraintProfile" minOccurs="0" />
    </sequence>
  </complexType>

  <simpleType name="FlexEnergyType">
    <restriction base="string">
      <enumeration value="PRODUCTION" />
      <enumeration value="CONSUMPTION" />
    </restriction>
  </simpleType>

  <complexType name="EnergySourceType">
    <sequence>
      <element name="classification" type="string" />
    </sequence>
  </complexType>

  <complexType name="PriceConstraint">
    <choice>
      <element name="minPrice" type="cim:Money" />
      <element name="maxPrice" type="cim:Money" />
    </choice>
  </complexType>

  <complexType name="EnergyConstraintProfile">
    <complexContent>
      <extension base="tns:TimeSeries">
        <sequence>
          <element
            name="energyConstraintInterval"
            type="tns:EnergyConstraintInterval"
            maxOccurs="unbounded"
          />
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```
<complexType name="EnergyConstraintInterval">
  <sequence>
    <element name="minDuration" type="tns:IntervalDuration" minOccurs="0" />
    <element name="maxDuration" type="tns:IntervalDuration" minOccurs="0" />
    <element name="startAfter" type="dateTime" minOccurs="0" />
    <element name="startBefore" type="dateTime" minOccurs="0" />
    <element name="endAfter" type="dateTime" minOccurs="0" />
    <element name="endBefore" type="dateTime" minOccurs="0" />
    <choice>
      <element name="energyConstraintList" type="tns:EnergyConstraintList" />
      <element name="powerConstraintList" type="tns:PowerConstraintList" />
    </choice>
    <element name="tariffConstraint" type="tns:TariffConstraint" />
  </sequence>
</complexType>

<complexType name="EnergyConstraintList">
  <sequence>
    <element name="energyConstraint" type="tns:EnergyConstraint" maxOccurs="unbounded" />
  </sequence>
</complexType>

<complexType name="EnergyConstraint">
  <choice>
    <element name="value" type="cim:RealEnergy" />
    <sequence>
      <element name="lowerBound" type="cim:RealEnergy" />
      <element name="upperBound" type="cim:RealEnergy" />
    </sequence>
  </choice>
</complexType>

<complexType name="PowerConstraintList">
  <sequence>
    <element name="powerConstraint" type="tns:PowerConstraint" maxOccurs="unbounded" />
  </sequence>
</complexType>

<complexType name="PowerConstraint">
  <choice>
    <element name="value" type="cim:ActivePower" />
    <sequence>
      <element name="lowerBound" type="cim:ActivePower" />
      <element name="upperBound" type="cim:ActivePower" />
    </sequence>
  </choice>
</complexType>

<complexType name="TariffConstraint">
  <sequence>
    <element name="minTariff" type="tns:EnergyTariff" minOccurs="0" />
    <element name="maxTariff" type="tns:EnergyTariff" minOccurs="0" />
  </sequence>
</complexType>

<complexType name="EnergyTariff">
  <sequence>
    <element name="value" type="float" />
    <element name="unit" type="cim:MonetaryAmountPerEnergyUnit" />
    <element name="multiplier" type="cim:UnitMultiplier" />
  </sequence>
</complexType>

<complexType name="TariffConstraintProfile">
  <complexContent>
    <extension base="tns:TimeSeries">
      <sequence>
```

```
        <element name="start" type="dateTime" />
        <element name="tariffConstraintInterval" type="tns:TariffConstraintInterval"
          maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="TariffConstraintInterval">
  <sequence>
    <element name="duration" type="tns:IntervalDuration" />
    <element name="tariffConstraint" type="tns:TariffConstraint" />
  </sequence>
</complexType>

<complexType name="FlexEnergySchedule">
  <complexContent>
    <extension base="tns:TimeSeries">
      <sequence>
        <element name="start" type="dateTime" />
        <element name="interval" type="tns:FlexEnergyScheduleInterval" maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="FlexEnergyScheduleInterval">
  <sequence>
    <element name="duration" type="tns:IntervalDuration" />
    <element name="energyAmount" type="cim:RealEnergy" />
    <element name="tariff" type="tns:EnergyTariff" />
  </sequence>
</complexType>

<complexType name="TimeSeries">
  <sequence>
    <element name="intervalDurationStep" type="duration" />
  </sequence>
</complexType>

<simpleType name="IntervalDuration">
  <restriction base="int">
    <minInclusive value="0"></minInclusive>
  </restriction>
</simpleType>

</schema>
```

9.3 Messages schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://mirabel-project.eu/schemas/messages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:model="http://mirabel-project.eu/schemas/model"
  attributeFormDefault="qualified"
  elementFormDefault="qualified"
  targetNamespace="http://mirabel-project.eu/schemas/messages"
>

  <!-- Schema imports -->

  <import namespace="http://mirabel-project.eu/schemas/model" schemaLocation="model.xsd" />
```

```
  <!-- Element definitions -->

  <element name="flexOffer" type="tns:FlexOffer" />
  <element name="flexOfferAcceptance" type="tns:FlexOfferAcceptance" />
  <element name="flexOfferAssignment" type="tns:FlexOfferAssignment" />

  <!-- Type definitions -->

  <complexType name="FlexOffer">
    <sequence>
      <element name="id" type="string" />
      <element name="creationTime" type="dateTime" />
      <element name="offeredById" type="string" />
      <choice>
        <element name="acceptBeforeTime" type="dateTime" />
        <element name="acceptBeforeInterval" type="duration" />
      </choice>
      <choice>
        <element name="assignmentBeforeTime" type="dateTime" />
        <element name="assignmentBeforeInterval" type="duration" />
      </choice>
      <element name="flexEnergy" type="model:FlexEnergy" />
    </sequence>
  </complexType>

  <complexType name="FlexOfferAcceptance">
    <sequence>
      <element name="id" type="string" />
      <element name="creationTime" type="dateTime" />
      <element name="flexOfferId" type="string" />
      <element name="acceptedById" type="string" />
      <element name="accepted" type="boolean" />
      <element name="explanation" type="string" />
    </sequence>
  </complexType>


  <complexType name="FlexOfferAssignment">
    <sequence>
      <element name="id" type="string" />
      <element name="creationTime" type="dateTime" />
      <element name="flexOfferId" type="string" />
      <element name="acceptedById" type="string" />
      <element name="schedule" type="model:FlexEnergySchedule" />
    </sequence>
  </complexType>

</schema>
```

10 Example messages

In this chapter the examples of FlexEnergy use in section 4.2 are used as examples of the use of the information modelling and messages defined in the XML Schema mapping presented in Chapter 9.

The examples in this section are presented as listings of XML lines which are instantiations according to the messages and model XML schema definitions. In each of the listings, the `msg` and `mdl` prefixes are used for the messages and model XML namespaces respectively.

The examples are provided in three sections: the three examples (heat pump, combined heat and power and electric vehicles) of section 4.2 are used as the basis for examples of the FlexOffer message. Next, two sections are included which provide examples instantiations in XML of the acceptance and assignment messages.

10.1 FlexOffer Example Messages

This section provides examples of the FlexOffer message structure. Basically this is the FlexOffer information model with additional information about who is offering the flexibility, information for identification of the offer, before when the offer must be accepted and before when it must be assigned.

10.1.1 Electric Vehicle Charging

The Listing 10.1 provides an XML example of a FlexOffer message for the flexibility in a charging process of an electric vehicle as described in section 4.2.1; the charging must have an initial charge as quick as possible and be fully completed before a certain point in time, but in this latter part of the charging process two charging power levels may be selected.

```
<?xml version="1.0" encoding="UTF-8"?>
<msg:flexOffer
        xmlns:msg="http://mirabel-project.eu/schemas/messages"
        xmlns:mdl="http://mirabel-project.eu/schemas/model"
        xmlns:xsi="http://www.w3.org/2011/XMLSchema-instance"
>
        <!-- offer and issuer identification -->
        <msg:id>flexoffer-ev-1</msg:id>
        <msg:creationTime>2011-01-01T07:45:00</msg:creationTime>
        <msg:offeredById>offering-party-id</msg:offeredById>

        <!-- offer acceptence and assignment constraints -->
        <msg:acceptBeforeTime>2011-01-01T07:55:00</msg:acceptBeforeTime>
        <msg:assignmentBeforeTime>2011-01-01T08:00:00</msg:assignmentBeforeTime>

        <!-- offered flexibility -->
        <msg:flexEnergy>
                <mdl:meteringPointID>metering-point-id-3</mdl:meteringPointID>
                <mdl:type>CONSUMPTION</mdl:type>

                <!-- consumption of 6000Wh in total -->
                <mdl:totalEnergyConstraint>
                        <mdl:value>6000</mdl:value>
                </mdl:totalEnergyConstraint>

                <mdl:energyConstraintProfile>
                        <mdl:intervalDurationStep>PT1M</mdl:intervalDurationStep>

                        <!-- inital charge -->
                        <mdl:energyConstraintInterval>
                                <!-- duration is equal to 15 minutes -->
                                <mdl:minDuration>15</mdl:minDuration>
                                <mdl:maxDuration>15</mdl:maxDuration>

                                <!-- start at 08:00 -->
                                <mdl:startAfter>2011-01-01T08:00:00</mdl:startAfter>
                                <mdl:startBefore>2011-01-01T08:00:00</mdl:startBefore>

                                <!-- consumption at 6000W -->
                                <mdl:powerConstraintList>
                                        <mdl:powerConstraint>
                                                <mdl:value>6000</mdl:value>
                                        </mdl:powerConstraint>
                                </mdl:powerConstraintList>
                        </mdl:energyConstraintInterval>
```

```
                        <!-- low power charge -->
                        <mdl:energyConstraintInterval>
                                <!-- duration is equal to 60 minutes -->
                                <mdl:minDuration>60</mdl:minDuration>
                                <mdl:maxDuration>60</mdl:maxDuration>

                                <!-- consumption between 0W and 3000W -->
                                <mdl:powerConstraintList>
                                        <mdl:powerConstraint>
                                                <mdl:lowerBound>0</mdl:lowerBound>
                                                <mdl:upperBound>3000</mdl:upperBound>
                                        </mdl:powerConstraint>
                                </mdl:powerConstraintList>
                        </mdl:energyConstraintInterval>

                        <!-- final high power charge -->
                        <mdl:energyConstraintInterval>
                                <!-- duration is equal to 60 minutes -->
                                <mdl:minDuration>60</mdl:minDuration>
                                <mdl:maxDuration>60</mdl:maxDuration>

                                <!-- consumption between 0W and 6000W -->
                                <mdl:powerConstraintList>
                                        <mdl:powerConstraint>
                                                <mdl:lowerBound>0</mdl:lowerBound>
                                                <mdl:upperBound>6000</mdl:upperBound>
                                        </mdl:powerConstraint>
                                </mdl:powerConstraintList>
                        </mdl:energyConstraintInterval>
                </mdl:energyConstraintProfile>
        </msg:flexEnergy>

</msg:flexOffer>
```

Listing 10.1 XML message example for electric vehicle charging flexibility.

10.1.2 Heat Pump

The XML FlexOffer message example in Listing 10.2 represents the information provided in section 4.2.2; in summary: in this example the heat pump has flexibility in when the heating process is started.

```
<?xml version="1.0" encoding="UTF-8"?>
<msg:flexOffer
        xmlns:msg="http://mirabel-project.eu/schemas/messages"
        xmlns:mdl="http://mirabel-project.eu/schemas/model"
        xmlns:xsi="http://www.w3.org/2011/XMLSchema-instance"
>
        <!-- offer and issuer identification -->
        <msg:id>flexoffer-heat-pump-1</msg:id>
        <msg:creationTime>2011-01-01T07:45:00</msg:creationTime>
        <msg:offeredById>offering-party-id</msg:offeredById>

        <!-- offer acceptence and assignment constraints -->
        <msg:acceptBeforeTime>2011-01-01T07:55:00</msg:acceptBeforeTime>
        <msg:assignmentBeforeTime>2011-01-01T08:00:00</msg:assignmentBeforeTime>

        <!-- offered flexibility -->
        <msg:flexEnergy>
                <mdl:meteringPointID>metering-point-id-2</mdl:meteringPointID>
                <mdl:type>CONSUMPTION</mdl:type>

                <mdl:energyConstraintProfile>
                        <mdl:intervalDurationStep>PT1M</mdl:intervalDurationStep>
                        <mdl:energyConstraintInterval>
                                <!-- duration is equal to 15 minutes -->
                                <mdl:minDuration>15</mdl:minDuration>
```

```xml
                                <mdl:maxDuration>15</mdl:maxDuration>

                                <!-- start between 08:00 and 08:15 -->
                                <mdl:startAfter>2011-01-01T08:00:00</mdl:startAfter>
                                <mdl:startBefore>2011-01-01T08:15:00</mdl:startBefore>

                                <!-- consumption at 2500W -->
                                <mdl:powerConstraintList>
                                        <mdl:powerConstraint>
                                                <mdl:value>2500</mdl:value>
                                        </mdl:powerConstraint>
                                </mdl:powerConstraintList>
                        </mdl:energyConstraintInterval>
                </mdl:energyConstraintProfile>
        </msg:flexEnergy>

</msg:flexOffer>
```


Listing 10.2 XML message example for heat pump flexibility.

10.1.3 Combined Heat and Power

Listing 10.3 provides an xml FlexOffer message example of flexibility in a combined heat and power installation as presented in section 4.2.3. In essence the flexibility in this example is the ability to change the power at which the installation is operated.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<msg:flexOffer
        xmlns:msg="http://mirabel-project.eu/schemas/messages"
        xmlns:mdl="http://mirabel-project.eu/schemas/model"
        xmlns:xsi="http://www.w3.org/2011/XMLSchema-instance"
>
        <!-- offer and issuer identification -->
        <msg:id>flexoffer-chp-1</msg:id>
        <msg:creationTime>2011-01-01T07:45:00</msg:creationTime>
        <msg:offeredById>offering-party-id</msg:offeredById>

        <!-- offer acceptence and assignment constraints -->
        <msg:acceptBeforeTime>2011-01-01T07:55:00</msg:acceptBeforeTime>
        <msg:assignmentBeforeTime>2011-01-01T08:00:00</msg:assignmentBeforeTime>

        <!-- offered flexibility -->
        <msg:flexEnergy>
                <mdl:meteringPointID>metering-point-id-1</mdl:meteringPointID>
                <mdl:type>PRODUCTION</mdl:type>
                <mdl:sourceType>
                        <mdl:classification>CHP</mdl:classification>
                </mdl:sourceType>

                <mdl:energyConstraintProfile>
                        <mdl:intervalDurationStep>PT1M</mdl:intervalDurationStep>
                        <mdl:energyConstraintInterval>

                                <!-- start at 08:00 -->
                                <mdl:startAfter>2011-01-01T08:00:00</mdl:startAfter>
                                <mdl:startBefore>2011-01-01T08:00:00</mdl:startBefore>

                                <!-- consumption of 1000Wh -->
                                <mdl:energyConstraintList>
                                        <mdl:energyConstraint>
                                                <mdl:value>1000</mdl:value>
                                        </mdl:energyConstraint>
                                </mdl:energyConstraintList>

                                <!-- production at 700W or 1000W -->
                                <mdl:powerConstraintList>
                                        <mdl:powerConstraint>
```

```
                                                <mdl:value>700</mdl:value>
                                                <mdl:value>1000</mdl:value>
                                        </mdl:powerConstraint>
                                </mdl:powerConstraintList>
                        </mdl:energyConstraintInterval>
                </mdl:energyConstraintProfile>
        </msg:flexEnergy>

</msg:flexOffer>
```

Listing 10.3 XML message example for combined heat and power flexibility.

10.2 FlexOffer Acceptance Example Messages

An example acceptance of the FlexOffer message in section 10.1.2 (the heat pump example) is provided as XML in Listing 10.4.

```
<?xml version="1.0" encoding="UTF-8"?>
<msg:flexOfferAcceptance
        xmlns:mdl="http://mirabel-project.eu/schemas/model"
        xmlns:msg="http://mirabel-project.eu/schemas/messages"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
        <msg:id>flexoffer-heat-pump-1-acceptence</msg:id>
        <msg:creationTime>2011-01-01T07:54:00</msg:creationTime>
        <msg:flexOfferId>flexoffer-heat-pump-1</msg:flexOfferId>
        <msg:acceptedById>accepting-party-id</msg:acceptedById>
        <msg:accepted>true</msg:accepted>
</msg:flexOfferAcceptance>
```

Listing 10.4 XML message example for acceptance of heat pump flex offer.

10.3 FlexOffer Assignment Example Messages

An example assignment of the FlexOffer message in section 10.1.2 (the heat pump example) is provided as XML in Listing 10.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<msg:flexOfferAssignment
        xmlns:mdl="http://mirabel-project.eu/schemas/model"
        xmlns:msg="http://mirabel-project.eu/schemas/messages"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
        <msg:id>msg:id</msg:id>
        <msg:creationTime>2011-01-01T07:59:00</msg:creationTime>
        <msg:flexOfferId>flexoffer-heat-pump-1</msg:flexOfferId>
        <msg:acceptedById>accepting-party-id</msg:acceptedById>

        <msg:schedule>
                <mdl:intervalDurationStep>PT1M</mdl:intervalDurationStep>
                <mdl:start>2011-01-01T08:15:00</mdl:start>
                <mdl:interval>
                        <mdl:duration>15</mdl:duration>
                        <mdl:energyAmount>625</mdl:energyAmount>
                </mdl:interval>
        </msg:schedule>
</msg:flexOfferAssignment>
```

Listing 10.5 XML message example for assignment for heat pump flex offer.

11 Conclusion

This document contains the specification of a generic data model, a message model and XML schema mapping for the information exchange between issuers and acquirers of flexible energy. This information exchange concerning flexibility can be introduced between prosumers of energy, such as households, and balance responsible parties, such as energy suppliers. The main goal of dealing with flexibility is to reduce the mismatch between energy demand and supply that is introduced by the increasing use of renewable energy sources. Reducing this mismatch will decrease the cost for imbalances in the energy network thanks to less usage of reserve power. Reduction of cost at the balance responsible parties can be partly passed to the issuers of flexibility. In principle, the specification in this document can be used by any pair of stakeholders that want to deal with or match flexibility in energy demand and supply.

The main goal of this document is to define a specification that can be used as input to a standardisation process at one of the major European standardisation bodies. Specifically, it will be used as input to a CEN Workshop to be started in the Spring of 2012 targeted towards all kinds of stakeholders in the energy domain that are directly or indirectly involved in such information exchange. This includes:

- Energy service providers: balance responsible parties, balance suppliers, metered data responsibles, energy service companies, aggregators, applications and services providers, power exchange platform operators (market operators),
- Energy system and network operators (system operators): transmission and distribution system/network operators (TSOs and DSOs/DNOs),
- Grid users (party connected to the grid): producers including producers using renewable energy sources;, consumers (including mobile consumers), storage owners,
- ICT suppliers, software engineering and IT services suppliers, integrators,
- European associations: branch and trade associations such as ENTSO-E, ebIX and EFET,
- Regulators and public authorities, academics and research, laboratories,
- Non-governmental organisations acting on environment matters.

Under the assumption that the CEN Workshop is started, it will lead to a CEN Workshop Agreement in the form of a recognized standard that can be used for further standardization towards a real European standard or norm.

References

[VerhStap10] Verhoosel, Jack P.C. and Stap, R.E., *D7.1 Standardization Roadmap*, MIRABEL project deliverable, June 2010.

[CIM09] IEC61970-301 Ed. 2, ‘Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base’, International Electrotechnical Commission, 2009.

[Europe2020] EUROPE 2020, a European strategy for smart, sustainable and inclusive growth, 2010.

[GREA] German Renewable Energy Act (EEG).

[Energinet.dk] Energinet.dk ‘download of market data’, DK-West area, 1 - 5 October 2009, ‘wind power vs. consumption’ and ‘price day ahead spot market’.

[ENTSO-E09] ENTSO-E: The Harmonized Electricity Market Role Model, version 2009-01, 2009.

[NesBenRing] D. Nestle, C. Bendel, J. Ringelstein, “Bidirectional Energy Management Interface (BEMI): integration of the low voltage level into grid communication and control”. *CIRED 19th International Conference on Electricity Distribution*.

[Power] www.powermatcher.net

[KonRum10] Konsman, M. and Rumph, F-J, *D2.1 State-of-the-art on data specifications*, MIRABEL project deliverable, June 2010.

[EMD09] ebIX, ‘Business information model for the exchange of metered data in the energy domain – Measure’, version 1.0, revision D, April 2009

[ebiXCuS] ebiX CuS Model, http://www.ebix.org/Documents/CuS%20Draft%20v2r1A%2020100409.pdf, retrieved on June 21 2010

[EFET] www.efet.org.

[EEGI10] www.smartgrids.eu/ documents/EEGI/ EEGI_Implementation_plan_May%202010.pdf

[ISO8601] ISO (International Organization for Standardization), representations of dates and times, 1988-06-15

Appendix A IEC CIM derived classes

This appendix contains descriptions from [CIM09] of the classes of this Common Information Model which are used in this deliverable. The descriptions of the classes, their attributes, enumerations and their constants are literal quotes of the Common Information Model standard document.

Figure 24 shows the currently adopted classes from [CIM09]. Please note that the relationships in this diagram are duplicates of the attributes defined (i.e. of `unit` and `multiplier` attributes).

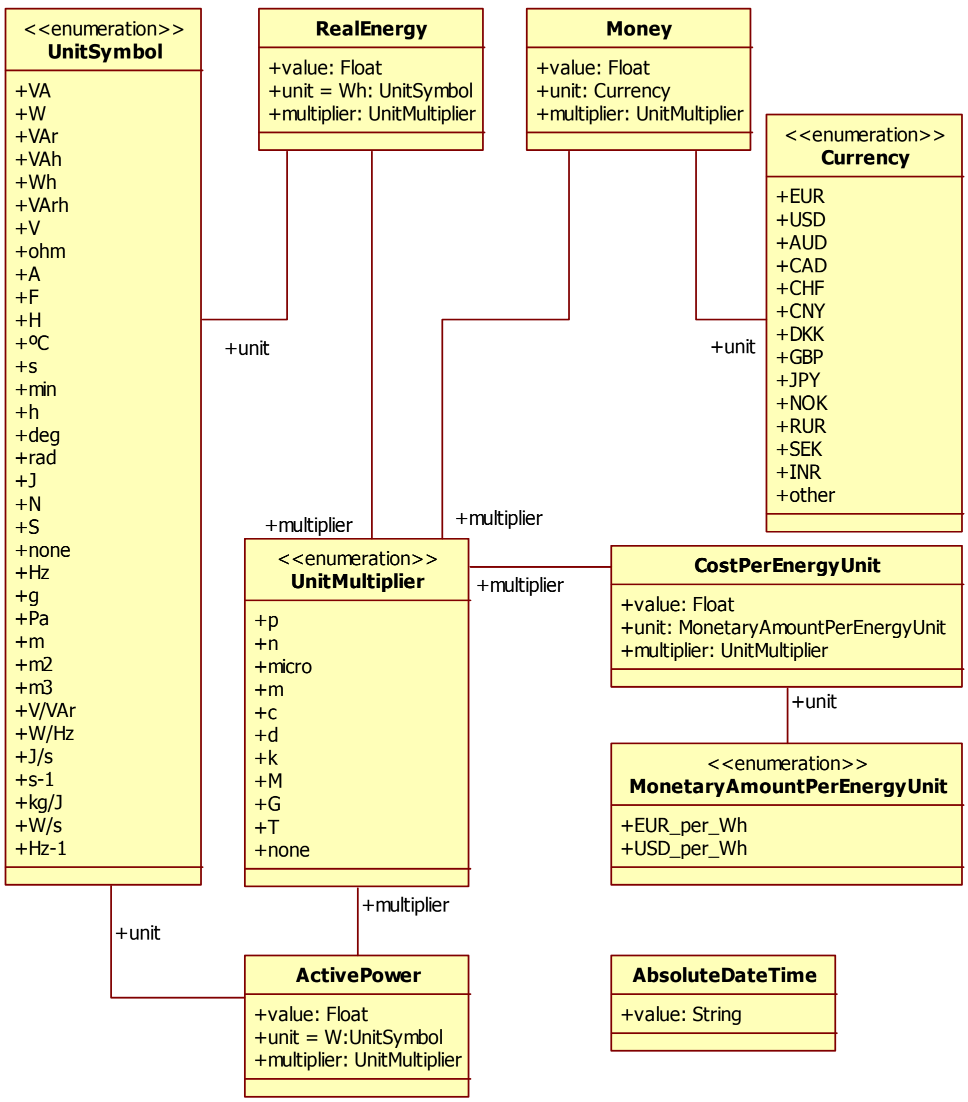

Figure 24: IEC CIM derived classes in the common package.

AbsoluteDateTime		
Description:	Date and time as "yyyy-mm-ddThh:mm:ss.sss", which conforms to ISO 8601. UTC time zone is specified as "yyyy-mm-ddThh:mm:ss.sssZ". A local time zone relative UTC is specified as "yyyy-mm-ddThh:mm:ss.sss-hh:mm". AbsoluteDateTime can be used both for calendar time, e.g. 2007-02-07T10:30, and for relative time, e.g. 10:30.	
Attributes:	value	String representation of date and time; refer to description of the class.
Source:	[CIM09]	

ActivePower		
Description:	Product of root mean square (RMS) value of the voltage and the RMS value of the in-phase component of the current.	
Attributes:	value	The amount of Watts.
	unit	The unit of the ActivePower class, which is W within [CIM09].
	multiplier	A multiplier, e.g. kilo or mega.
Source:	[CIM09]	

CostPerEnergyUnit		
Description:	Cost, in units of currency, per quantity of electrical energy produced or consumed.	
Attributes:	value	A float value that expresses the cost.
	unit	This refers to the currency that is being used and is of the type MonetaryAmountPerEnergyUnit; it expresses e.g. Euros, dollars, etc.
	multiplier	This attribute is of the type UnitMultiplier. An example would be k Euros (1000 Euros) or using the none multiplier indicating that the unit should not be multiplied.
Source:	[CIM09]	

`Currency`		
Description:	Monetary currencies. Apologies for this list not being exhaustive.	
Enum. constants:	`USD`	US dollar
	`EUR`	European Euro
	`AUD`	Australian dollar
	`CAD`	Canadian dollar
	`CHF`	Swiss francs
	`CNY`	Chinese yuan renminbi
	`DKK`	Danish crown
	`GBP`	British pound
	`JPY`	Japanese yen
	`NOK`	Norwegian crown
	`RUR`	Russian ruble
	`SEK`	Swedish crown
	`INR`	India rupees
	`other`	Another type of currency.
Source:	[CIM09]	

`MonetaryAmountPerEnergyUnit`		
Description:	Monetary amount per energy unit.	
Enum. constants:	`USD_per_Wh`	A number of USD per Watt-hour.
	`EUR_per_Wh`	A number of EUR per Watt-hour.
Source:	[CIM09]	

`Money`		
Description:	Amount of money	
Attributes:	`value`	The basic monetary amount, e.g. 20 or 0,15.
	`unit`	The monetary unit.
	`multiplier`	A multiplier of the basic value, e.g. to express k€.
Source:	[CIM09]	

`RealEnergy`		
Description:	Real electrical energy	
Attributes:	`value`	The amount of Watt-hours.
	`Unit`	The unit of the `RealEnergy` class, which is Wh within [CIM09].
	`multiplier`	A multiplier, e.g. kilo or mega.
Source:	[CIM09]	

UnitMultiplier			
Description:	The unit multipliers defined for the CIM.		
Enum. constants:	`p`	Pico	10^{-12}
	`n`	Nano	10^{-9}
	`micro`	Micro	10^{-6}
	`m`	Milli	10^{-3}
	`c`	Centi	10^{-2}
	`d`	Deci	10^{-1}
	`k`	Kilo	10^{3}
	`M`	Mega	10^{6}
	`G`	Giga	10^{9}
	`T`	Tera	10^{12}
	`none`		
Source:	[CIM09]		

UnitSymbol		
Description:	The units defined for usage in the CIM.	
Attributes:	`VA`	Apparent power in volt ampere
	`W`	Active power in watt
	`VAr`	Reactive power in volt ampere reactive
	`VAh`	Apparent energy in volt ampere hours
	`Wh`	Real energy in watt hours
	`VArh`	Reactive energy in volt ampere reactive hours
	`V`	Voltage in volt
	`ohm`	Resistance in ohm
	`A`	Current in ampere
	`F`	Capacitance in farad
	`H`	Inductance in Henry
	`ºC`	Relative temperature in degrees Celsius
	`s`	Time in seconds
	`min`	Time in minutes
	`h`	Time in hours
	`deg`	Plane angle in degrees
	`rad`	Plane angle in radians
	`J`	Energy in joule
	`N`	Force in Newton
	`S`	Conductance in Siemens
	`none`	Dimension less quantity, e.g. count, per unit, etc.
	`Hz`	Frequency in hertz
	`g`	Mass in gram
	`Pa`	Pressure in Pascal (n/m2)

	m	Length in meter
	m2	Area in square meters
	m3	Volume in cubic meters
	V/VAr	Volt per volt ampere reactive
	W/Hz	Watt per hertz
	J/s	Joule per second
	s-1	per second
	kg/J	Mass per energy
	W/s	Watt per second
	Hz-1	per Hertz
Source:	[CIM09]	