GOFLEX

An Overview of the Flex-offer Concept

V0.9

January 2018

Imprint

Author(s): **Laurynas Šikšnys (AAU), Torben Bach Pedersen (AAU)**
Participant(s):
Reviewer(s):

Project: Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX)

Version: **V0.9**

Contact: **Laurynas Šikšnys – siksnys@cs.aau.dk**
Website: **www.goflex-community.eu**

Legal disclaimer

The project Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731232. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC). INEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

Document History

Version	Date	Status	Author	Comment
0.8	11 Jan 2018	Initial document	Laurynas Šikšnys	First draft
0.9	12 Jan 2018	Initial document	Laurynas Šikšnys and Torben B. Pedersen	First document

Table of Contents

List of Acronyms and Abbreviations

Abbrevia-tion	Definition
Flex-offer	A flexible offer – a representation of a single flexibility unit
BRP	Balance Responsible Party

1 Bottom-up energy systems

The GOFLEX (Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid) project focuses on a new so-called *bottom-up* or *cellular* energy system:

- An energy system where significant energy production is occurring at lower levels of the grid, and energy market roles such as "producers" and "market operators" have new opportunities at the lower levels of the grid. Each grid level may possess most of the functions present at higher levels. Each portion of a system that contains a minimal set of roles may be called a cell. In the future energy system, such cells may very well sit within other cells in a manner similar to nesting dolls.

This cellular approach to arranging the energy system is well suited to incorporating distributed sources of renewable energy. A group of actors capable of meeting their own energy needs can trade energy and flexibility with each other and form their own cell.

The cellular system works *bottom-up*: the smallest cells (subsystems) are commercial and residential buildings, houses and industry plants (e.g. performing as microgrids). The next level typically correspond to distribution grids and the third level to transmission grids.

2 Flex-Offer Concept Overview

To be able to exchange information about energy flexibility among different actors within a *cell*, there is a need for a common representation of flexible loads. The European project MIRABEL proposed a format to encode this information, called a *flex-offer* [1, 3].

A visual representation of a (simple) *flex-offer* is shown in Figure 1. Each bar in the graph corresponds to a time slice of energy consumption, with the lower part representing the minimum amount of energy that a flexible resource needs to provide its service, and the upper part an interval within which it can adjust its consumption, while still satisfying functional constraints (e.g., comfort temperature). This is called to (energy) *amount flexibility*. Another type of flexibility is *time flexibility* as shown in Figure 1. Time flexibility is provided when an energy load can be shifted within a time interval, defined by an earliest start time at which the flexible resource can start its consumption, and a latest end time at which it should be done. When created, a flex-offer is assigned a *baseline schedule* that corresponds to the consumption pattern that the associated flexible resource prefers to follow. Updated schedules can be assigned to the flex-offers to modify the consumption behaviour of the flexible resource, utilizing its provided flexibility. More advanced forms of the flex-offer exist, and will be discussed later.

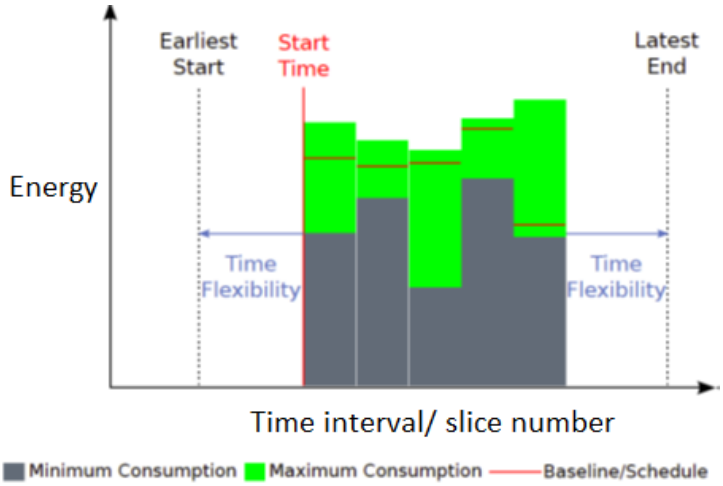

Figure 1 A visual representation of the simple Flex-Offer

3 Flex-offer Aggregation and Disaggregation

In general, the flex-offer representation makes it practical to exchange flexibility information between different entities. However, flex-offers from individual flexible resources (e.g., heat pumps, electric vehicles) typically represent *small* flexible loads. Thus, a single (small) flex-offer has low impact and is of little interest for electricity trading, peak shaving, and balancing demand and supply on the grid, where required balancing capacities are much higher. At the same time, optimising energy loads based on a large number of flex-offers is a computationally hard problem, which requires dealing with many decision variables and constraints originating from many flex-offers. By utilizing *flex-offer aggregation [2]*, flexibilities from individual appliances can be combined and thus offered in a more useful and effective aggregated form. Such aggregated flexibility can again be represented as flex-offers – but with much larger energy amounts and flexibility margins. Aggregation is typically performed by BRPs, system operators, and/or entities called *Aggregators*. They receive flex-offers from individual flexible resources and then aggregate these flex-offers. The flexibility of aggregated flex-offers tends to be lower than the joint flexibility of the flex-offers that compose them. This reduction in flexibility is, however, unavoidable in order to reduce flex-offer scheduling complexity and to increase their value (e.g., on the flexibility market). After aggregation, *schedules* are typically assigned to the aggregated flex-offers (e.g., based on energy sold on the market). By respecting all inherent aggregated flex-offer constraints, a schedule specifies exact start times and aggregated energy amounts be assigned to the underlying flexible resources. Such schedules are *disaggregated* to schedules for each individual flex-offer it is composed of. This operation is denoted flex-offer *disaggregation*. Disaggregated schedules are finally forwarded to the flexible resources which initially offered flexibility. This flex-offer aggregation, scheduling, and disaggregation process is illustrated in Figure 2.

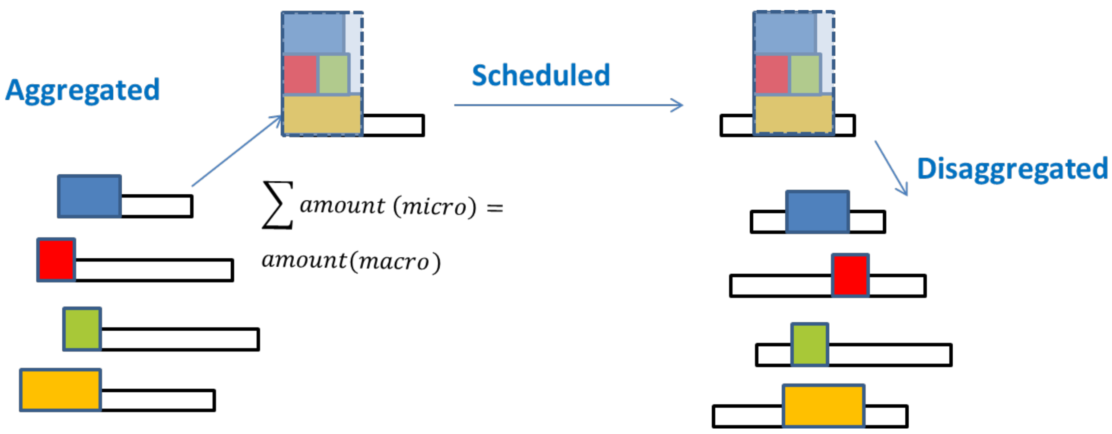

Figure 2 – Flex-offer aggregation, scheduling, and disaggregation process

4 Flex-offer Constraints

To cover a wide range of flexible resources, a flex-offer identifies different types of flexibility and uses specialized constraints to characterize each type of flexibility – see Fig. 3.

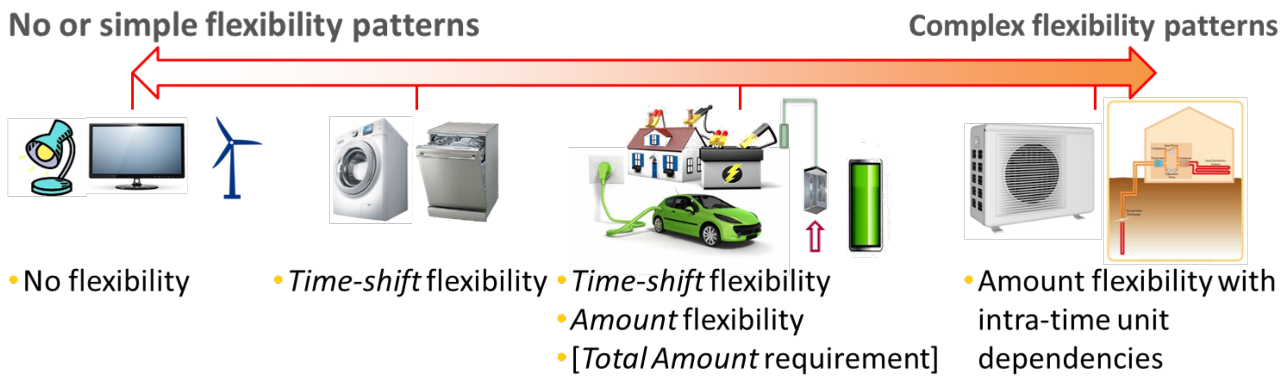

Figure 3 – Complexity of load flexibility patterns supported by flex-offers

A flex-offer in simple forms (Fig. 1) activates the following types of constraints to capture *start time* and *energy amount* flexibilities:

- **Start time constraint –** it is a range defined by the two parameters: *startAfterTime*, *startBeforeTime* - see Fig. 1.
- **Energy amount constraint -** for every discrete (e.g., 15min) interval of an active device operation, energy amount flexibility is characterized by a range with *energyConstraint.lower* and *energyConstraint.upper* as lower and upper bounds, respectively – see Fig. 1.

However, some flexible resources might require additional more advanced (flexibility) constraints to be added to a flex-offer to model its flexibility or data exchange flow more accurately:

- **Total energy constraint –** it is a range [*totalEnergyConstraint.lower*, *totalEnergyConstraint.upper*] which bounds the total energy amount requested or offered within the full active operation of a flexible resource.
- **Dependent energy amount constraint -** like the energy amount constraint, this constraint captures the minimum and maximum energy amounts at a discrete interval *t* depending on the total energy consumed at the intervals 1..*t*-1. A dependent energy amount constraint [4] for a specific discrete time interval *t* is given as 2D energy flexibility polygon – see Figure 4. This constraint should be used for the most advanced forms of flexible resources (e.g., heat-pumps), where the flexibility changes over time and is dependent on an internal system state (e.g., temperature).
- **Acceptance time constraint** - defined by the parameter *acceptanceBeforeTime*, this constraint sets the deadline on when a flex-offer receiving party (e.g., BRP) should acknowledge successful acceptance or rejection of the flex-offer. A flex-offer rejection may occur if, e. g., flex-offer constraints or other metadata (e.g., prices) are invalid or inappropriate (e.g., quantities are too small, prices are too high).
- **Assignment time constraint** – defined by the parameters *assignmentBeforeTime* and *assignmentBeforeDurationSeconds,* this constraint sets the deadlines on when flex-offer schedule update (assignment) is allowed to be sent by the flex-offer receiving party (BRP) to a flex-offer issuing party (flexible resource). A deadline can be an *absolute timestamp* (*assignmentBeforeTime*) or a *relative duration* (*assignmentBeforeDurationSeconds*) with respect to the scheduled operation activation time.

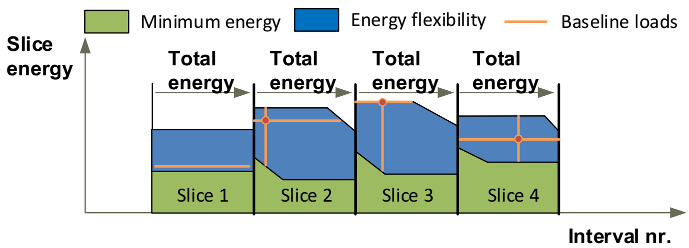

Figure 4 – An example of a flex-offer with dependent energy amount constraints

For all these flex-offer constraint types, scalable aggregation and disaggregation techniques have been developed [2, 4, 6].

5 Flex-Offer Pricing

In addition to flexibility, flex-offer may capture related information such as *price*. Each flex-offer may define, a *deviation price*, which has to be paid to *flexibility provider* (e.g., Prosumer) by a *flexibility consumer (e.g., BRP)*, for requesting a 1kWh deviation from the baseline (preferred schedule) which was initially reported by the flexibility provider – see Fig. 1. This yields a number of "V"-shaped price curves (in the linear case) associated to each discrete time interval of an active device operation – see Fig. 5.

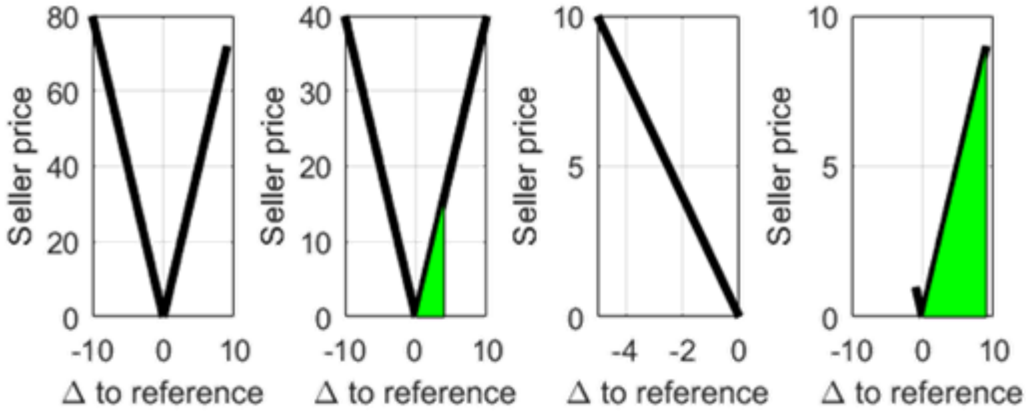

Figure 5 – An example of a flex-offer with associated deviation prices

The deviation price is paid for the successful activation of flexibility (a deviation from the baseline). Alternatively, *flexibility* may be treated as an asset and priced in the explicit form (e.g., monthly). For this, the flexibility provider (a flexible resource) may get rewarded by the flexibility consumer (e.g., BRP) based on the number of flex-offers, total *time* and *energy amount* flexibility, etc. offered – see Fig. 6.

Customer: arrowhead_lift

Item	Value	Price
Number of flexoffers	2	
Fixed reward for all flexoffers		10.00 DKK
Total Time Flexibility	0 time units (15 min)	0.00 DKK
Total Energy Flexibility	68.83 kWh	6.88 DKK
Number of default schedule deviations	0	0.00 DKK
The sum of stat time scheduling deviations with respect to the default schedule	0 time units (15 min)	0.00 DKK
The sum of energy deviations with respect to the default schedule	0.00 kWh	0.00 DKK
Total Reward		16.88 DKK

Figure 6 –Pricing of flexibility based on flex-offer contracts

6 Flex-Offer JSON representation

An example JSON representation [6] of a flex-offer used in the ARROWHEAD [1] and TOTALFLEX [5] projects with activated basic constraints for *acceptance time*, *assignment time*, *start time*, *energy amount*, and *total energy* flexibilities is as follows:


```
{
    "id": 0,
    "state": "Assigned",
    "offeredById": "SELF",
    "acceptanceBeforeTime": "2018-01-12T06:45:00+01:00",
    "assignmentBeforeDurationSeconds": 0,
    "assignmentBeforeTime": "2018-01-12T07:00:00+01:00",
    "creationTime": "2018-01-12T06:15:00+01:00",
    "durationSeconds": 9000,
    "endAfterTime": "2018-01-12T09:45:00+01:00",
    "endBeforeTime": "2018-01-12T11:45:00+01:00",
    "numSecondsPerInterval": 900,
    "startAfterTime": "2018-01-12T07:15:00+01:00",
    "startBeforeTime": "2018-01-12T09:15:00+01:00",
    "totalEnergyConstraint": {
        "lower": 18.0,
        "upper": 20.0
    },
    "slices": [{
        "durationSeconds": 900,
        "costPerEnergyUnitLimit": 1,
        "energyConstraint": {
            "lower": 2.877109715311126,
            "upper": 4.650334966274789
        }
    },
    {
        "durationSeconds": 900,
        "costPerEnergyUnitLimit": 1,
        "energyConstraint": {
            "lower": 5.424558499875854,
            "upper": 8.589466603032985
        }
    },
    {
        "durationSeconds": 900,
        "costPerEnergyUnitLimit": 1,
        "energyConstraint": {
            "lower": 6.02222779348911,
            "upper": 9.055657773803548
        }
    }],
    "flexOfferSchedule": {
        "startTime": "2018-01-12T07:15:00+01:00",
        "energyAmounts": [3.7637223407929588,
        7.0070125514544195,
        7.538942783646329]
    },
    "defaultSchedule": {
        "startTime": "2018-01-12T07:15:00+01:00",
        "energyAmounts": [3.7637223407929574,
```

```
        7.0070125514544195,
        7.538942783646329]
    }
}
```

7 References

1. Le Guilly, Thibaut; Siksnys, Laurynas; Albano, Michele; Pedersen, Per D.; Stluka, Petr; Ferreira, Luis Lino; Skou, Arne Joachim; Pedersen, Torben Bach; Olsen, Petur. An Energy Flexibility Framework on the Internet of Things. In S. Hamrioui (Ed) *The Success of European Projects using New Information and Communication Technologies*, page 17-37, 2016 http://www.scitepress.org/documents/EPS_Colmar_2015.pdf
2. Laurynas Siksnys, Emmanouil Valsomatzis, Katja Hose, Torben Bach Pedersen: Aggregating and Disaggregating Flexibility Objects. *IEEE Transactions of Knowledge and Data Engineering* 27(11): 2893-2906 (2015) http://ieeexplore.ieee.org/document/7123643/
3. Emmanouil Valsomatzis, Katja Hose, Torben. Bach Pedersen and Laurynas Šikšnys. *Measuring and Comparing Energy Flexibilities*. In *Proceedings of the EBDT/ICDT Workshops, 2015, CEUR WS proceedings* http://ceur-ws.org/Vol-1330/paper-14.pdf
4. Laurynas Siksnys, Torben Bach Pedersen. *Dependency-based FlexOffers: scalable management of flexible loads with dependencies.* In *Proceedings of ACM e-Energy* 2016: 11:1-11:13, https://dl.acm.org/citation.cfm?doid=2934328.2934339
5. TOTALFLEX project team. *TotalFlex Final Report.* Final reporting from ForskEL project 10774. 2016. http://www.ciss.dk/wp-content/uploads/sites/3/2017/06/TotalFlex-final-report.pdf
6. The Virtual Market of Energy (VME) implementation developed in the Arrowhead project. https://github.com/lawrizs/ARROWHEAD_VME