

‘ISSN APPLICATION FRAME’ – A FLEXIBLE AND PERFORMANT FRAMEWORK HOSTING SMART GRID APPLICATIONS

Mario FASCHANG, Mark STEFAN, Friederich KUPZOG
 AIT Austrian Institute of Technology GmbH, Vienna – Austria
 [mario.faschang; mark.stefan; friederich.kupzog]@ait.ac.at

Alfred EINFALT, Stephan CEJKA
 Siemens AG Österreich, Vienna – Austria
 [alfred.einfalt; stephan.cejka]@siemens.com

ABSTRACT

Resilient operation of novel networked smart grid systems requires for active distribution grid monitoring, management, and control. In order to reduce system complexity and harness fast control interaction, grid-related software applications (e.g., voltage control) are destined to operate directly on a controller in the respective intelligent secondary substation (iSSN). In this paper, we propose a flexible and performant software framework for hosting of smart grid applications – the so called iSSN application frame. It provides means for efficient field measurement access, data preprocessing and representation. As such, it supports simple development and field deployment for such applications. For the iSSN application frame, we developed a set of independent software components for the representation of smart buildings and the distribution grid, as well as a data storage. The components are loosely coupled by a cloud-based middleware, which allows for flexible extension. Thus, the iSSN application frame forms the foundation for the efficient operation of novel distribution grid applications.

1 INTRODUCTION

The implementation of the smart grid approach introduces modern information and communication technology (ICT) components to the distribution grid domain (i.e., the secondary substations). This ICT-based ‘upgrade’ of secondary substations to *intelligent secondary substation nodes* (iSSN) enables distribution grid operators (DSO) to make better use of the available grid infrastructure, aim for novel business objectives, and ultimately save on investments, increase their grids’ reliability, and gain new revenue.

Classically, grid related applications for network planning and operation of low voltage (LV) networks has been operated offline e.g. as functionality of DSO’s GIS System. Network planning is based on classical network maps and a few remote slave pointer measurements at transformer bus bar in combination with standardized load profiles. LV grid operation is passive and mainly event triggered. Customers call the service center if and where failures or disturbances have occurred.

The idea of the iSSN-approach is the local usage of the dense set of distribution grid measurements in the secondary substation to perform analysis, to compute set-points for distributed actors (switches, transformers’ tap-changers, inverters, heat

pumps, EVs, etc.), and to perform other distribution grid-related functions.

In order to perform these functions in a distributed manner, software applications have to be operated on an automation device (e.g., industry grade computer) in the secondary substation. Fig. 1 shows the location of such an automation device in the Smart Grid Architecture Model (SGAM – cf. [3]). It is located in the component layer’s distribution domain, in the substation zone.

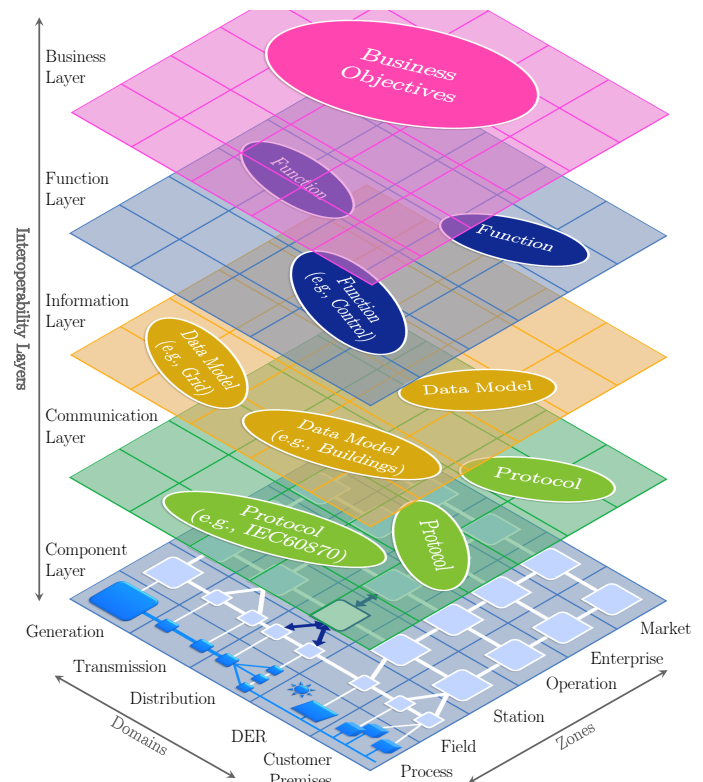


Fig. 1: In the Smart Grid Architecture Model, the iSSN application frame is operated on the component layer’s distribution domain in its (sub)station zone. It serves as a hosting framework for distribution grid-related software application.

The applications, which are operated on the automation device in the iSSN, use various protocols to interact with other devices or systems (e.g., smart meters, SCADA systems, etc.). Data models are required to organize and abstract the underlay information. This abstraction allows for convenient development of functions (e.g., voltage controller – as proposed in [4]),

which in turn serve the DSO's business objectives (cf. Fig. 1).

In order to allow for a convenient development and efficient operation of the applications on the automation device, a set of requirements have to be fulfilled by the iSSN application frame. These requirements are listed in the next chapter. Subsequently, we present the realization of the application frame in Chapter 3 with detailed information about the implemented data models. In the last chapter (Chapter 4), we conclude the current state of our work and give an outlook to the ongoing and further application development and operation in the scope of the Austrian smart city project *SCDA – Smart City Demo Aspern* [1].

2 REQUIREMENTS

Requirements for the iSSN application frame are given below. They are derived from the grid application developer's perspective. It should be mentioned here, that these are a small subset of the requirements that are relevant for the operation of a complete smart grid automation component in an iSSN.

R1 – Modular, resilient, and scalable software system The modularity, resilience, and scalability are the non-functional requirements that are relevant for the application frame. The modularity is required for an easy integration of the developed smart grid application into the application frame and the extension of the frame's functionality. Resilience is required due to the operation of critical infrastructure. It should avoid single points of failure, and allow for hot standby of software components and for transparent fail-over in case one of the software modules stops its operation.

R2 – Local data availability and preprocessing In order to allow for performant application operation, field measurements need to be stored locally on substation level. Depending on the application, both, live field measurements as well as historical data of at least the last twelve months should be directly available to the developed application. Data preprocessing by the application frame is required in order to guarantee a basic data quality to the grid-related applications.

R3 – Model representation of field components Field measurements, the grid structure, and its current system state need to be abstracted in the application frame by adequate data models. This is required to hide complexity of field situation and to allow a simple and structured access to field sensors and actuators for the developed applications.

3 APPLICATION FRAME

In order to cover the previously mentioned requirements for efficient grid application operation, the iSSN application frame has been developed. General architectural decisions and implemented components are described in the next subsections.

3.1 General Architecture

The general architecture of the iSSN application frame is depicted in Fig. 2. It consists of the following stand-alone software modules.

- **Grid Information Module (GIM)** – represents the distribution grid with its topology and all relevant components (e.g., grid topology, lines, switches and states, loads), including preprocessing and provisioning of power profiles from grid sensors.
- **Building Information Module (BIM)** – represents the smart buildings that are connected to the respective distribution grid, including preprocessing and provisioning of nominal power profiles delivered by Building Energy Management Systems (BEMS).
- **Storage Module** – acts as local data storage for field measurement time-series – including grid and building related measurements and meta-data.

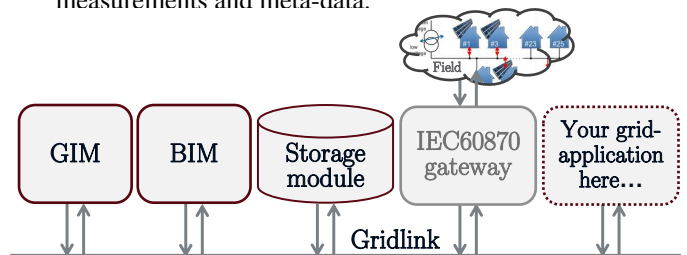


Fig. 2: The general structure of the iSSN application frame consisting of GIM, BIM, and the Storage module. The IEC60870 gateway component is the interface to the field devices.

Communication between several application modules is achieved by use of Gridlink, a decentralized distributed message bus developed in Java and based on vert.x¹ and Hazelcast². Gridlink provides distributed execution of an application by using modules, each of which providing certain functionality and dynamically forming a cluster of known instances during execution. By design, Gridlink does not contain any central component, thus no single point of failure is introduced (according to requirement R1). Modules may join or leave the bus at any time during execution without influencing other modules' behavior. Therefore, they can replace each other to enable transparent fail-over by claiming the same identification name on the bus.

3.2 Storage Module

It is vital that modules have access to historical and current data measured by smart meters and other sensors in the field. For this purpose, the storage module is responsible for permanent storage of time-series and meta data of sensor data. It utilizes a Java-based embedded data store, shown to be superior to state-of-the-art off-the-shelf SQL and NoSQL solutions w.r.t. data

¹ <http://vertx.io>

² <http://hazelcast.com>

retrieval time and required storage size (cf. [2]). Immutable values, frequent readouts, and the generation of statistical indicators are optimally supported.

3.3 Data Preparation

Both the GIM and the BIM request power profiles from the storage module and process them for further usage by applications. Caused by misbehavior of the field sensors or lack of data, gaps might occur within the recorded profiles. As a consequence, data preprocessing is necessary – in particular, such gaps must be identified.

Each power profile consists of a list of succeeding time stamps and corresponding measurement values. A gap is an interval of time, defined by

$$t_{gap} \geq 2 \cdot t_{min} \quad (1)$$

where t_{min} is the minimum time difference between any two successive time stamps within the profile. To provide useful data for applications, all gaps are identified and so called time-series are created. A time-series contains at least two time stamps, which indicate the start \underline{ts}_i and the end \overline{ts}_i of seamless profile data. If a given profile has been analyzed and no gaps were found, then there exists only one time-series. If one gap was identified somewhere in the profile, then two time-series are created.

Example Assume the following list of time stamps t_i and measures values P_i .

i	1	2	3	4	5	6	7	8
t_i [s]	1.0	2.0	3.0	4.0	7.0	8.0	9.0	10.0
P_i [kW]	2.64	3.91	4.11	3.04	4.23	6.66	5.12	7.65

Based on these time stamps $t_{min} = 1.0$ s and thus $t_{gap} \geq 2.0$ s. It can be seen, that there exists a gap between the fourth and fifth time stamp. Therefore two time series with the following time stamps are created:

$$\begin{aligned} \underline{ts}_1 &= 1.0 \text{ s} & \overline{ts}_1 &= 4.0 \text{ s} \\ \underline{ts}_2 &= 7.0 \text{ s} & \overline{ts}_2 &= 10.0 \text{ s} \end{aligned}$$

3.4 Grid Information Module

The Grid Information Module (GIM) collects grid-data, pre-processes this data and provides this information about the distribution grid and its elements to other modules in a well structured form.

The grid topology is provided as XML-based *Common Information Model (CIM)* and imported by the GIM building up a topological model using the information about nodes, lines, transformers, connections, etc. For each device, an entry in a table is created including device specific information and a unique identifier.

To store the topology information, a matrix of size $n \times n$ is created where n is the number of grid elements. The connection information between element i and element j is stored in line i and column j and contains $e_{i,j} = \langle s_{(i,j)}, l_{(i,j)} \rangle$, where $s_{(i,j)} \in \{0, 1\}$ is the switch state and $l_{(i,j)} \in \mathbb{Z}$ the length (as number of elements) of the connection between the two elements.

Initially, the connection information is set to $\langle 0, 0 \rangle$ for each pair of elements and based on the imported topology information it is updated to $\langle 1, 1 \rangle$ if two elements are directly connected. As a result, the matrix can be used to identify all directly connected elements within the grid.

To provide more details about the grid to other modules, the topological information is extended. Therefore, well-defined starting points of the grid are identified (e.g., MV/LV connection point). Based on these elements, a recursive algorithm iterates through the elements of the matrix until some also predefined final nodes (e.g., energy consumers) are reached. Based on the traversed path within the network topology, the connection information is updated (connection length of indirectly connected elements is changed). By using positive and negative connection lengths, parent-child-relations in terms of network topology are shown. In case of topology changes (e.g., change of a switch position), the information is updated in the Grid Information Module. Thus, the GIM can be seen as an up-to-date representation of the grid.

To sum up, this matrix can be interpreted as extended adjacency matrix and it can be used to identify the relation between two elements of the network, e.g., if they are connected, the path length in terms of hops between two elements, and if it is possible to reach an element when changing switch positions.

Example Assume the distribution grid in Fig. 3a. Each element gets a unique id and then the topology matrix A is generated as explained before.

$$\begin{pmatrix} \langle 0, 0 \rangle & \langle 1, 1 \rangle & \langle 1, 2 \rangle & \langle 1, 3 \rangle & \langle 1, 4 \rangle & \langle 0, 3 \rangle & \langle 0, 4 \rangle \\ \langle 1, -1 \rangle & \langle 0, 0 \rangle & \langle 1, 1 \rangle & \langle 1, 2 \rangle & \langle 1, 3 \rangle & \langle 0, 2 \rangle & \langle 0, 3 \rangle \\ \langle 1, -2 \rangle & \langle 1, -1 \rangle & \langle 0, 0 \rangle & \langle 1, 1 \rangle & \langle 1, 2 \rangle & \langle 0, 1 \rangle & \langle 0, 2 \rangle \\ \langle 1, -3 \rangle & \langle 1, -2 \rangle & \langle 0, 0 \rangle & \langle 1, 1 \rangle & \langle 1, 1 \rangle & \langle 0, 0 \rangle & \langle 0, 0 \rangle \\ \langle 1, -4 \rangle & \langle 1, -3 \rangle & \langle 1, -2 \rangle & \langle 1, -1 \rangle & \langle 0, 0 \rangle & \langle 0, 0 \rangle & \langle 0, 0 \rangle \\ \langle 0, -3 \rangle & \langle 0, -2 \rangle & \langle 0, -1 \rangle & \langle 0, 0 \rangle & \langle 0, 0 \rangle & \langle 0, 1 \rangle & \langle 0, 0 \rangle \\ \langle 0, -4 \rangle & \langle 0, -3 \rangle & \langle 0, -2 \rangle & \langle 0, 0 \rangle & \langle 0, 0 \rangle & \langle 0, -1 \rangle & \langle 0, 0 \rangle \end{pmatrix}$$

The generated information in matrix A can be used to illustrate the grid topology including their dependencies as directed graph [6] (Fig. 3b).

Directly connected topology elements (with closed switches) are connected by a solid arrow in the graph. The dashed arrow indicates, that a connection between nodes exists but the switch is open. The graph representation is useful for a wide range of different networks to illustrate the network-elements and the relation between them.

Another function of the GIM is to model grid monitoring devices, in particular their position within the topology and

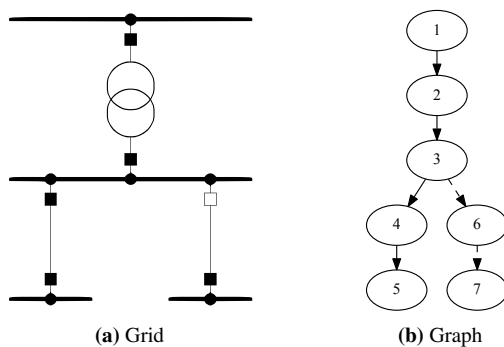


Fig. 3: Grid (a) and graph (b) representation of matrix A (solid lines represent closed switches and dashed lines connections with open switches).

to provide their corresponding power profiles. Therefore the monitoring devices are identified and their power profiles are requested from the storage module by the GIM.

In general, grid monitoring devices are sensing active and reactive power flow for each phase of a line. Thus, measured power is the sum of all sub-branches and all directly connected nodes. GIM analyzes the determined positions of the monitoring devices and – in case that a parent-child-relation exists to a sub-branch – so called residual profiles are calculated. Assume that d_i and d_j are monitoring devices where d_j is a child of d_i and $P(d_i)$ and $P(d_j)$ are the corresponding power profiles.

As a consequence, the residual profile of d_i is calculated by $P_R(d_i) = P(d_i) - P(d_j)$ by subtracting each time stamp of the profiles. Due to this data preparation, other modules and applications can directly use the power profiles for further calculations.

3.5 Building Information Module

The functionality of the Building Information Module (BIM) is similar to the GIM – collect and preprocess data and provide it to other modules and applications.

In almost the same manner as the GIM, the BIM requests profiles from the storage module, in particular profiles of buildings instead of monitoring devices within the distribution grid. Also, gaps within the profiles are identified and seamless time-series are created. As a result, suitable data can be provided for other modules of the application frame.

At the moment building representation by their power profiles and data preparation are the only implemented functions of BIM. Further, functions like in-building actuator modeling and energy market participation-related functions are envisioned.

4 CONCLUSION AND OUTLOOK

Recent work (e.g., [5]) has shown that decentralized operation of distribution grid-related software applications require spe-

cific requirements. In order to enable resilient operation of networked smart distribution grids we developed the iSSN application frame consisting of data models for the distribution grid and the buildings and a data storage module.

First applications have proven the suitability of the presented frame and its modules. The modularization of the raw field data into a grid and a building related model has led to a clear, modular and extensible frame, which is highly convenient for application developers.

In the course of the research project *Smart City Demo Aspern (SCDA)* [1] we are going to use the iSSN application frame to develop and deploy a set of applications for decentralized low voltage grid management. Further extension are foreseen especially for the BIM functionality (modeling of in-building actuators) and the data preprocessing (synchronization of asynchronous field measurements).

ACKNOWLEDGMENTS

The research project *SCDA – Smart City Demo Aspern* is funded within the program *Smart Cities* by the Austrian *Climate and Energy Fund* under project number 846141.

REFERENCES

- [1] Aspern Smart City Research GmbH & Co KG (ASCR). SCDA – Smart City Demo Aspern. <http://www.ascr.at/foerderungen/scda/>, 2016. [Online; accessed 02-02-2016].
- [2] S. Cejka, R. Mosshammer, and A. Einfalt. Java embedded storage for time series and meta data in Smart Grids. In *Proceedings of 2015 IEEE Smart Grid Comm, 2-5 Nov 2015, Miami, Florida*, pages 439–444, Nov 2015.
- [3] CEN-CENELEC-ETSI. Smart Grid Reference Architecture. Technical Report, CEN-CENELEC-ETSI Smart Grid Coordination Group, Nov 2012.
- [4] A. Einfalt, F. Zeilinger, R. Schwalbe, B. Bletterie, and S. Kadam. Controlling active low voltage distribution grids with minimum efforts on costs and engineering. In *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 7456–7461, Nov 2013.
- [5] M. Faschang. *Rapid Control Prototyping for Networked Smart Grid Systems Based on an Agile Development Process*. PhD thesis, Vienna University of Technology, Vienna, 2015.
- [6] P. Hines, S. Blumsack, E. C. Sanchez, and C. Barrows. The topological and electrical structure of power grids. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10, Jan 2010.