GRIND: An Generic Interface for Coupling Power Grid Simulators with Traffic, Communication and Application Simulation Tools

David Chuang
Shanghai Jiao Tong University
SEIEE
Shanghai, China
unknownmight@sjtu.edu.cn

Björn Schünemann, David Rieck
Fraunhofer FOKUS
ASCT
Berlin, Germany
{bjoern.schuenemann, david.rieck}@fokus.fraunhofer.de

Ilja Radusch
Technische Universität Berlin
OKS / DCAITI
Berlin, Germany
ilja.radusch@dcaiti.com

***Abstract*—The prospective penetration of the electric vehicle fleet will bring about certain repercussions due to their high demand in power. Simulations are, therefore, of importance for making estimations for assessing the impact of the incoming electric vehicle fleet on the power systems and to predict some cost specific values. To analyse the working conditions of power grids, power system simulators are not to be dispensed with. However, to have a complete picture of a charging scenario involving electric vehicles, further aspects should be preferably observed, e.g., traffic, communication and application aspects. So far, no sophisticated tool exists that incorporates the further simulation aspects for a comprehensive investigation of electric mobility. To address this issue, this paper proposes a concept for enabling the coupling of power system simulators with simulators of other domains. The concept is described in form of a specification called Grid Analysis Interface Definitions (GRIND). As a proof of concept, the V2X Simulation Runtime Infrastructure (VSimRTI) and the electrical power system simulator OpenDSS are coupled following the proposed GRIND specification.**

Keywords*—*VSimRTI; Simulation Tools; Electric Mobility.

I. Introduction

Recent research and development is continuously striving to create innovation that improves the standard in driving and minimize hazardous situations on the roads. To address the issues about the local CO_2 emissions, the vehicular industry is shifting toward focusing on manufacturing electric vehicles [1]. However, the upcoming plug-in electric vehicle (PHEV) fleet might have certain negative impact on the power grid due to their high power demand. In order to reduce risks and repercussions of a prospective penetration of a fleet, a good foresight must be obtained before the roll-outs take place. Especially testing is crucial for making accurate estimations to assess the impact of the incoming PHEV fleet on the power systems and to predict some cost specific values. One option to do field testing vehicular set-ups might require non trivial budgets, and they are rigid and non-flexible. The other option is to resort to simulations. There exists several simulation tools for power system analysis. Power system simulations alone, however, do not suffice in order to conduct analysis on charging patterns of PHEV's. To have a complete picture of the elaborate happenings, a traffic simulator for modelling vehicular traffic, a communication simulator to facilitate an information exchange among traffic participants and infrastructure units, and an application simulator for emulating in-vehicle and mobile applications should be incorporated into the simulation environment. So far, no simulation environment is available, which provides a sophisticated modelling of all these aspects.

To amend the described shortage, this paper proposes a concept for interconnecting power system simulators with simulators from other domains. The work is inspired by TraCI[2], "a technique for interlinking road traffic and network simulators" to facilitate research on the VANET domain. As a proof of concept, a concrete implementation will be done by coupling the open source load flow simulator OpenDSS [3] with the powerful simulation framework VSimRTI [4] that enables the coupling of simulators of different research domains. VSimRTI is a promising candidate since it already couples existing traffic, communication, and application simulators.

This paper is structured as follows: In Section II, relevant work will be presented including the simulation architecture VSimRTI. Concepts for realizing the coupling process and made design decisions follow in Section III. Moreover, implementation details are given. Finally, the proof of concept is introduced in Section IV, and a conclusion is given in Section V.

II. Background

A. Simulation Couplings

A notable work of high relevance is the **Tra**ffic **C**ontrol **I**nterface (TraCI) [2]. TraCI is an API designed to "interlink road traffic and network simulators", it is a generic protocol specification that allows external programs to control the microscopic and macroscopic vehicle behaviour in a traffic simulation from outside. To design the concept, the authors recognized the fact that vehicular behaviour can be broken down into atomic operations called "mobility primitives". Each one of those mobility primitives were used to set a basis for constructing a message. An important feature of the TraCI interface is that it was made generic and is, therefore, neutral to simulation specific details. This feature allows any vehicular simulation tool to become a TraCI server and any program to be the client. TraCI also adheres to a server and client architecture, which allows it to be platform-independent and

 ISBN: 978-1-61208-308-7

it also allows the communication to take place over different machines.

Another work of relevance is presented by Andersson, Elofsson, Galus et al., who conducted a wide range of research in the PHEV domain [5], [6]. They proposed a framework that couples the energy hub concept with an extended version of MATSim in order to investigate their ideas. However, the coupling does not allow trivial replacement of the particular tools since the framework was not designed in a generic manner.

B. VSimRTI

VSimRTI [4] is a generalized framework for the coupling of different simulators, each for a particular domain, following an ambassador concept inspired by some fundamental concepts of the High Level Architecture (HLA). All management tasks, such as synchronization, interaction and lifecycle management are handled completely by VSimRTI. Several optimization techniques, such as optimistic synchronization, are implemented. The generic VSimRTI interfaces allow an easy integration and exchange of simulators. Consequently, the deployment of simulators is enabled for each particular domain. For immediate use, a set of simulators is already coupled with VSimRTI: the traffic simulators VISSIM and SUMO; the communication simulators ns-3, OMNeT++, JiST/SWANS, and a cellular communication simulator; a Java-based application simulator; and several visualization and analysis tools. VSimRTI is a promising candidate for the objectives of this work. Therefore, it is chosen as the underlying system for coupling power system simulators with simulators from other domains.

III. CONCEPT OF REALIZATION

A. Requirements

This paper proposes the **Gr**id Analysis **In**terface **D**efinitions (GRIND), a specification for the flexible coupling of power system simulators with simulators of other domains. For the realization of GRIND, the following requirements were defined:

- GRIND is to allow interactions between a power grid simulator and simulators of different domains. Interactions occur during the runtime of a simulation. That means, the coupled simulation tools can retrieve and change the state of the power grid simulator during the runtime of a simulation.
- GRIND is specified in a generic way so that it can be used with an arbitrary power grid simulator. Furthermore, its interfaces are to be flexible enough for the coupling of simulation tools of different domains.
- GRIND is to enable distributed simulations, i.e. the coupled simulation tools can run on different machines and operating systems.

In the following sections, the concept is explained, which has been developed to fulfil these requirements.

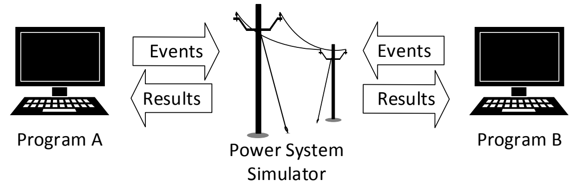

Fig. 1. Interaction between programs and a power system simulator

B. Approach

The aim of GRIND is to provide a specification, which enables developers to couple power system simulators with other tools. Since TraCI [2] follows a similar approach for the VANET domain – to couple traffic simulators with other simulation tools, some concepts of GRIND are inspired by TraCI concepts.

Even if all existing power system simulators provide similar services, each of them has its own way to model its inner working. Some simulators have similar ways for modelling the grid elements, while others use different calculation methods. However, since the simulation models of all simulators are based on the same theories and principles for load flow analyses, some more abstract "information" can be identified that apply to all power system simulators. For example, most power system simulators include loads and generators as part of a circuit albeit in different formats. The high level notion of a load, therefore, is applicable to any power system simulator without having to regard how it is internally modelled. In addition to this abstract data, several events exist, which change the current state of a grid. Typical examples are the addition or the removal of a load triggering the increase/decrease of the power consumption. Analogous to the fact that the same high level information is processed by any power system simulator, events will likewise be independent of the used power system simulator. Regarding the interactions, the power system simulator acts as a provider for data related to the grid. In terms of events, they can be triggered by both – the grid simulator and the external system. Consequently, the exchange of information and events have to be standardized in a way that both sides understand the communication.

A typical work flow is as follows. The power system simulator and another simulator with an interest in grid related data initiate their communication. The grid simulator internally performs any calculations needed to solve the state of the power grid. Once in a while, the external simulator sends queries or update changes to the power system simulator, which are used by the power system simulator to update the state of the power grid. This work flow is depicted in Figure 1.

C. GRIND Server and Client

In order to establish the communication, a channel has to be set up for a bi-directional message exchange. For that matter, simulators have to be extended by the needed functionality to be compatible with GRIND. However, simulators come in different strengths with respect to extensibility. Some simulators can be augmented with ease, while others do not allow expanding practices. To cover every possible

 ISBN: 978-1-61208-308-7

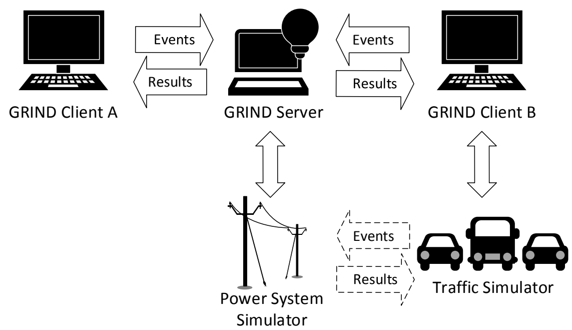

Fig. 2. Interaction between programs and a power system simulator according to GRIND

setup, GRIND implements a server and client architecture. The communication is realized by network sockets, which guarantees a platform independent use of GRIND. Moreover, client and server do not need to be installed on the same machine. According to different setups, the GRIND server or clients can act as either a middleware or an extension of a particular simulator following the GRIND specification. A middleware is needed if a simulator is self-contained and does not support extensions. If a tool is extensible, the interfaces for GRIND can be integrated into its system without the need of a middleware.

The proposed architecture is depicted in Figure 2. The dashed arrows indicates, which parties having a conversation. To realize this conversation in the described generic way, the power system simulator establishes a connection to the GRIND Server. Then, the GRIND Server connects the GRIND Client B, which is coupled to the traffic simulator. In this way, the power system simulator and the traffic simulator can interact with each other.

D. GRIND Messages

To have an interaction between simulators, it has to be defined what kind of information can be exchanged. Both parties have to be able to interpret the received information – i.e. the server and the client have to "speak the same language". For that purpose, GRIND specifies a set of information that is grouped into discrete units. Such an information unit is termed "message" where one message encompasses several related information. The content of the different message types is defined by GRIND in a way that all needed information can be transferred by the available pool of message types. To transfer an information, the suitable message type is chosen, the information is encapsulated there, and, then, the message is sent. Since the message type is known, the other side can interpret the received message.

Most existing power simulators are able to share certain features in common. These features are used to infer information that are universally applicable. Using these features as a foundation similar to the concepts of "mobility primitives" introduced by TraCI [2], actions can be identified. These actions are used to define the message types of the GRIND protocol. Since the pool of messages is to cover an area as wide as possible, not every message is universally applicable. In other words, some power system simulators, providing fewer functionality than others, disregard message types they cannot process.

The following paragraphs give a brief introduction of the message types defined by GRIND. Since the space of this paper is limited, the messages are described on a higher level.

1) NewFile: Most existing power system simulators support the setup of a circuit by loading configuration files albeit in different formats. In the case where these files are not stored on the server side, the client can use a NewFile message indicating the incoming transmission of an actual file.

2) CreateCircuit: Certain power system simulators model an internal circuit within their system prior calculation. This message can be used to prompt the power system simulator to use any existing resource for constructing a circuit.

3) Topology: In order to avoid a redundant parsing, a client is not aware of the structure of a circuit. This message type contains those pieces of grid data, which are to be sent to a client.

4) ChangeLoad, ChangeGeneration: Although the topology remains static, the load dispatch is highly variable. By these message types, common changes in the load configuration are transferred by the client to the server to update the grid when necessary.

5) NewLoad, NewGenerator, RemoveLoad, RemoveGenerator: Loads or generators can be added or removed from the system with help of these message types. However, it is not very common to remove a generator in a running system.

6) SolveGrid: The most important service provided by a power system simulator is to perform a power flow calculation. Since most tools do not perform this action automatically, this message type requests the power system to solve the grid using its current parameters.

7) GridResults: Not every client needs the same parameters from a solved power grid. For example, some clients might only need the total line values, while other clients could require the detail state of the entire topology in one minute steps. Therefore, it is not efficient to specify each parameter as one single message. Instead, an aggregate message that is freely adjustable is defined by GRIND. The detailed content of the message can be specified by the developer according to the need of data of a particular simulator. This message is sent as a response to the SolveGrid message.

E. Addressing Scheme

Each power system simulator models its circuits in a different manner. Some simulators define the entire circuit within matrices while other more sophisticated ones virtually model the elements. In order to address individual elements, the developer has to come to an agreement in form of an addressing scheme. For instance, if a load is saved within a cell in a matrix on the third row and fifth column, this load can be uniquely addressed by using "3,5" as identifier. Consequently, this identifier can be included in a ChangeLoad message whenever the load "3,5" has to be increased in power. In contrast, other simulators, for example OpenDSS, name the buses and do not need such a naming process. Instead, they simply indicate the literal name in a ChangeLoad message.

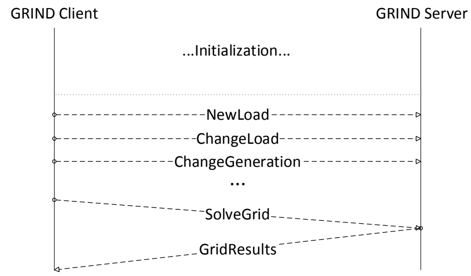

Fig. 3. An example of a message flow following the GRIND protocol

F. Protocol

Since TCP is the underlaying communication protocol, it can be assumed that all messages do arrive in order. Most messages do not require a direct response from the server side, however, they aim to trigger an activity the server is to start. A common scenario is to involve the server to change the load parameters on a bus by a ChangeLoad message or remove an entire generator capacity from a bus with help of a RemoveGeneration message. However, one particular message does require a reply from the GRIND server: When the client asks for results of the power flow calculation using the SolveGrid message. An example of a message flow is depicted in Figure 3.

IV. Proof of Concept

A. Implementation using OpenDSS and VSimRTI

For realizing the proof of concept, the power system simulator OpenDSS and the simulation architecture VSimRTI were coupled following the GRIND specification. OpenDSS was selected because of its richness in features and its well-designed interfaces. The advantage of VSimRTI is that it is already coupled with several traffic, communication, and application simulators. Thus, a coupling of OpenDSS and VSimRTI creates a simulation environment, which can cover a wide range of different simulation aspects. In the planned simulations, the traffic simulator SUMO, the communication simulator JiST/SWANS, and the VSimRTI application simulator are integrated in the VSimRTI simulation setup – additionally to the power system simulator OpenDSS.

B. Scenario

For the planned simulations, a test scenario is set up where the electric grid is presented by the IEEE 30 test feed [7]. The selected area of the simulation is the City of Roanoke (USA). The overall electric grid, including changes induced by charging processes, is modelled by OpenDSS. Roanoke map data from OpenStreetMap[8] are used to model the road network. The vehicular traffic is generated by SUMO. An in-vehicle application is implemented, which guides the driver to an unused charging station and controls the charging processes. The information exchange among vehicles and infrastructure units is simulated by JiST/SWANS.

C. Aim of the Proof of Concept

Additionally to the investigation of performance issues like scalability and simulation speed, the proof of concept is to demonstrate that the coupling of OpenDSS and VSimRTI following the proposed GRIND specification is a promising approach to enable comprehensive simulations of electric mobility scenarios. The different aspects electric grid, vehicular traffic, information exchange among traffic participants and infrastructure units, and emulation of in-vehicle and mobile applications can be modelled by this solution. Since all these aspects influence each other during the runtime of a simulation, a dynamic coupling is necessary, which allows interactions among the simulators during a simulation run. The proof of concept shall illustrate that the realized coupling fulfils these requirements and, thus, enables more detailed investigations of electric mobility and its impacts.

V. Conclusion

This paper proposes a concept for the flexible coupling of power system simulators with simulators of other domains. As the result, simulation tools from different domains can be linked to an arbitrary grid simulator. This is particularly helpful for comprehensive investigations of electric mobility where the influences and interactions of power grid, vehicular traffic, communication, and in-vehicle applications are to be considered, e.g., in cooperative ITS. The generic server-client architecture of the proposed GRIND specification allows cross-platform compatibility and platform independence. The implemented coupling of the power system simulator OpenDSS and the simulation architecture VSimRTI follows the GRIND specification and, hence, creates a simulation environment, which enables a comprehensive assessment of novel electric mobility solutions. In the next step, the introduced proof of concept will be simulated to demonstrate the effectiveness and potency of this work.

Acknowledgment

The presented research work is part of the eMERGE project financially supported by the German Federal Ministry of Transport, Building and Urban Development (BMVBS).

References

[1] E. Knipping and M. Duvall, “Environmental assessment of plug-in hybrid electric vehicles volume 2: United states air quality analysis based on aeo-2006 assumptions for 2030,” *Electric Power Research Institute*, 2007.

[2] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, “TraCI: an interface for coupling road traffic and network simulators,” in *Proceedings of the 11th communications and networking simulation symposium*. ACM, 2008, pp. 155–163.

[3] R. Dugan and T. McDermott, “An open source platform for collaborating on smart grid research,” in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1–7.

[4] B. Schünemann, “V2x simulation runtime infrastructure vsimrti: An assessment tool to design smart traffic management systems,” *Computer Networks*, vol. 55, pp. 3189–3198, October 2011.

[5] M. Galus, R. Waraich, M. Balmer, G. Andersson *et al.*, “A framework for investigating the impact of phevs,” 2009.

[6] M. Geidl, G. Koeppel, P. Favre-Perrod, B. Klockl, G. Andersson, and K. Frohlich, “Energy hubs for the future,” *Power and Energy Magazine, IEEE*, vol. 5, no. 1, pp. 24–30, 2007.

[7] R. Christie, “Power systems test case archive,” *Electrical Engineering dept., University of Washington*, 2000.

[8] M. Haklay and P. Weber, “Openstreetmap: User-generated street maps,” *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

