Smart Grid Software Applications as an Ultra-Large-Scale System: Challenges for Evolution

M. Anvaari, D. S. Cruzes, R. Conradi
Department of Computer and Information Science, Norwegian University of Science and Technology

***Abstract*—Software applications play a major role to support the smartness in Smart Grid. Such applications are in a never-ending state of flux due to rapidly changing expectations from the users and stakeholders. Hence, evolution is an important matter for development of Smart Grid software applications and the challenging factors of software evolution in Smart Grid should be identified. This position paper presents a discussion on the Smart Grid software applications as an ultra-large-scale system, focusing on the possible challenges that need to be addressed for an effective and efficient evolution of the area.**

***Index Terms*—Smart Grid, software applications, software evolution, systems of systems, ultra-large-scale system.**

I. INTRODUCTION

Smart Grid is a relatively new concept that has been introduced and used in the electric power industry since the first years of the 21st century. One of the main common views among different organizations and authors is that Smart Grid refers to the application of information and communication technology to evolve the traditional electric power grid to a two-way digitized grid. It will reduce the complexity issues of today's grid by integrating intelligent control technologies [6][28][34].

The information technology domain employs different decision and control software applications to supply the intelligence of Smart Grid. Such software applications include firmware embedded in the intelligent electronic devices (IED), applications that are integrated in substation automation systems and control centers and also enterprise back office software [7][28]. Furthermore, software failures are among the dominant causes of system cost and schedule overruns especially in large-scale systems [32] such as Smart Grid. Therefore, the software applications play a major role in Smart Grid and their development is a substantial concern for any utility wishing to promote its current grid to a smarter one.

Even though some Smart Grid software applications are already developed, most of them are either in analysis, development or evolution phases. In addition, the whole collection of Smart Grid software applications is in a never-ending state of flux due to rapidly changing expectations from the direct and indirect users and stakeholders. Therefore evolution is an important matter for development of Smart Grid software applications and the challenging factors of software evolution in Smart Grid should be identified.

Software evolution challenges have not been addressed in the Smart Grid literature. Most of the challenges discussed in the area are either in general or in relation to electric power and control aspects [2][11][14]. Even those fewer attempts related to information technology aspects of the concept, mainly correspond to "data and information" integration, storage, analysis and visualization [1][4][15][21], application of web-services and service-oriented-architecture in Smart Grid context [3][33], and cloud and distributed computing in Smart Grid [27][30].

Addressing the challenges of software evolution in Smart Grids is important for developing approaches and methods to improve the management of software evolution in Smart Grid applications. Different kinds of software intensive systems follow different evolution patterns and the nature of each evolution depends on the nature of evolution drivers that in turn depend on the scale of related artifacts [25]. Therefore the initial step to find out the evolution challenges of a software system is realizing its nature and scale.

Smart Grid software applications can be considered as an *Ultra-Large-Scale System.* A Systems-of-Systems (SoS) is a large-scale concurrent and distributed system that consists of complex systems [16]. Ultra-Large-Scale System (ULSS) is a complex software-intensive system that is embedded in a business and social context with diverse stakeholders [29]. ULSSs are special kind of SoSs and have particular characteristics in addition to SoSs characteristics that make them qualitatively more complex and challenging [26][29]. The reason we consider Smart Grid a ULSSs is due to their decentralized operation and control, conflicting requirements, continues evolution and indistinct boundary between the system and its users.

The goal of this paper is presenting a discussion on the Smart Grid software applications as a ULSS, focusing on the possible challenges that need to be addressed for an effective and efficient evolution of the area.

This paper is organized as follows: Section two introduces ULSS and SoS. The last part of this section argues on considering Smart Grid software applications as a ULSS. Section three discusses the evolution challenges of Smart Grid software applications as the main contribution of this paper. Conclusions and some suggestions for future research are followed in the last section.

II. SYSTEMS-OF-SYSTEMS (SOS) AND ULTRA-LARGE-SCALE SYSTEM (ULSS)

A Systems-of-Systems (SoS) is a system that its components are large and complex enough to be considered as systems in their own right [22]. SoSs are comprised of constituent-systems that posses operationally and managerially independent characteristics [18]. An Ultra-

Large-Scale System (ULSS) is a complex software-intensive system that is deeply embedded in a business and social context with many and diverse stakeholders [29]. ULSS is a socio-technical ecosystem and involves people, policies, cultures and economies in addition to large-scale software intensive systems [12]. Although many authors believe that ULSSs are more complex and challenging than SoSs [24][29][32], some of them discuss that ULSSs inherit characteristics of SoSs, therefore have some characteristics in common with today's SoSs [26][32]. Hence in this paper considering the characteristics and taxonomy of both types of systems would be helpful.

Maier considers five principles characteristics for SoSs that distinguish them from very large and complex but monolithic systems: operational independence of the elements, managerial independence of the elements, evolutionary development rather than fully planned existence, emergent behavior and geographic distribution [22].

The report of Software Engineering Institute on ULSS which is the main reference in ULSS area claims that characteristics of ULSSs that will arise because of their scale are much more revealing and Maier's definitions are not so useful for understanding the underlying technical problems of ULSSs [32]. So it considers seven characteristics for ULSSs that some are common with SoSs and others are particular to ULSSs:

1) *Decentralization* in a variety of ways including decentralized data, development, evolution and operational control,

2) *inherently conflicting, unknowable and diverse requirements*,

3) *continues evolution and deployment* rather than staged evolution,

4) *heterogeneous, inconsistent and changing elements* rather than uniform parts,

5) *erosion of the people/system boundary*,

6) *normal failures* rather than excepted ones and

7) *new paradigms for acquisition and policy* [32].

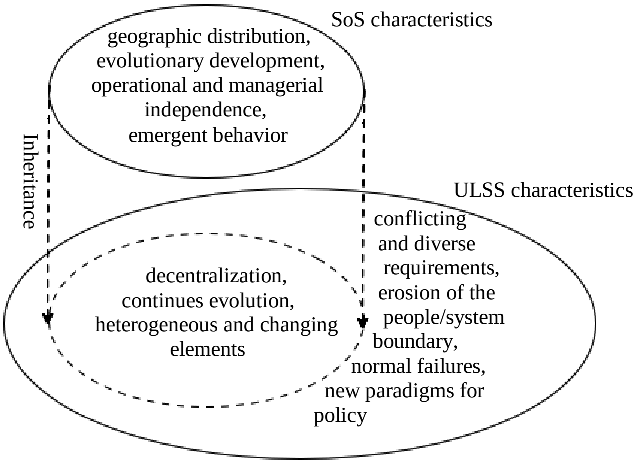

Fig. 1. SoS and ULSS characteristics domains

Fig. 1. demonstrates the domains of characteristics in SoS and ULSS. It shows how similar and distinct are these two system types in the characteristics. Based on the mentioned characteristics for SoSs and ULSSs considering taxonomy for them is possible.

A. Taxonomy

Different SoSs and ULSSs can be placed on different points along the scale of SoS/ULSS characteristics [31]. As a result the methods and approaches of development and evolution of one classification of such systems may be incompatible for another classification. Therefore defining the taxonomy is essential for addressing systems evolution challenges. So far there is not any concrete taxonomy for ULSSs whereas in SoS area some authors have identified taxonomy.

Cook considers dedicated and virtual categories for SoSs [8] while Maier mentions voluntary category besides those two categories [22].

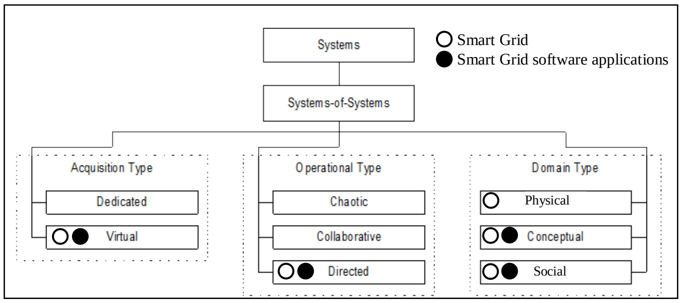

Fig. 2. SoS taxonomy [13] and the position of Smart Grid and Smart Grid software application in the taxonomy

Gideon et al. [13] have defined the broadest taxonomy for SoSs that covers the previous taxonomy too. They divide different SoSs into seven categories with regards to the acquisition, operation and domain of SoSs that is shown in Fig. 2. In the acquisition classification *dedicated* SoSs are those that from beginning are consciously designed and engineered to be SoS while the acquisition and engineering of *virtual* SoSs is generally unplanned.

In the operational classification *chaotic* SoSs are those in which there is no central control authority or managerial entity and thus no agreed upon purpose. Therefore the function of such systems is often random and unpredictable. The other category is *collaborative* SoSs in which the constituent systems interact voluntarily almost out of necessity. Management authorities may issue standard practices, but they have little power to coerce the behavior of component systems and it is up to the component systems to acquiesce to those standards. The overall behavior of these systems may still be unpredictable. The third category is *directed* SoSs that are controlled by a central management authority. Although the constituent systems still operate relatively independently, their operation is predetermined at the highest level.

Finally in the domain classification *physical* SoSs are systems that operate in or on the physical world that would be with/without human interaction. The second category is *conceptual* SoSs that are abstract in nature and do not exist as tangible entities in physical space nor do they operate on or manipulate matter. The third category is *social* SoSs that their essential difference is that the main form of interaction in these systems is between people or organization utilizing policies and procedures [13].

Since the taxonomy presented by Gideon et al. has taken people and policies into account, for the purpose of this paper it can be considered as taxonomy of ULSSs too.

B. Smart Grid Software Applications as a ULSS

Kremers et al. believe that the electrical grid can be considered as a true SoS [17]. They argue that electrical grids mostly satisfy the five principal characteristics of SoS defined by Maier:

1) Electrical grid systems are able to operate independently since generators, loads and other components work properly when they are not connected to the grid.

2) Electrical grid systems maintain a continuing operational existence independent of the whole grid; therefore they have managerial independence too.

3) The existence and future development of the electrical grid is evolutionary and it does not appear fully formed.

4) The behaviors and functions of the electrical grid are emergent properties of the entire system and cannot be localized to any constituent system.

5) The geographic extent of the electrical grid is large and its components can readily exchange information [17].

The mentioned characteristics are also existed in Smart Grid as a smarter version of electrical grid; therefore Smart Grid also can be considered as a SoS. However, when we come into software applications of Smart Grid there are some attributes that are missed in Maier's defined characteristics of SoSs.

In Smart Grid which usually functions at large geographic scales such as county-level, software applications will be used thousands times per day. Therefore even if the failure rate in one application is very low, the failure of all software applications in the whole grid will be trivial. Furthermore, the software applications in Smart Grid will be developed and used by different stakeholders in different organizations from bulk generation part to customers. Hence the requirements of the applications will be diverse and even in some conflicting situation. Finally, by considering the substantial role of plug-in hybrid electric vehicles and solar panels in Smart Grid, people will not be only users of the system. They will be a part of the system and affect the overall behavior of the system as local operators.

Therefore, normal failures, diverse and conflicting requirements and indistinct people/system boundary are the important characteristics of Smart Grid software applications that can change our viewpoint to consider them as a ULSS rather than a SoS. Nevertheless they still have common characteristics with SoSs and considering such characteristics can be helpful in identifying their development and evolution challenges.

It is useful to delimit Smart Grid software applications to a narrower category of ULSS/SoS. So we apply the taxonomy of Gideon et al. (Fig. 2.) that is the most comprehensive taxonomy in ULSS/SoS area. Regarding the acquisition type, since the formation of Smart Grid is evolutionary rather than revolutionary [19] it can be considered as a virtual system and thus the software applications of Smart Grid. Regarding the operational type Smart Grid software applications are directed because local or governmental management authorities control them in spite of their independent operation. Finally about the domain type although Smart Grid itself include physical material, conceptual components, people and organizations and therefore is a physical/conceptual/social system, but the software applications of Smart Grid do not contain physical objects and thus they are a social/conceptual system. Fig. 2. shows the position of "Smart Grid" and "Smart Grid software applications" in the taxonomy of SoS/ULSS.

The conclusion of this section is that Smart Grid software applications can be considered as a virtual, directed, social/conceptual Ultra-Large-Scale system. Hence, studying evolution challenges in ULSS context would be a base to identifying evolution challenges in Smart Grid software applications as a ULSS.

III. EVOLUTION CHALLENGES IN SMART GRID SOFTWARE APPLICATIONS

Lehman and Fernandez-Ramil [20] believe that evolution phenomena in the software context are not confined to the software code. Related artefacts such as requirements specifications, designs, documentation, definitions, goals, paradigms, algorithms, languages, usage practices, the sub-processes and processes of software evolution and so on, also evolve. These entities interact, impact and affect one another [20]. Therefore in order to develop efficient evolution methods and tools in any software-intensive system, different aspects should be considered. This conclusion is true for Smart Grid software applications that are a ULSS.

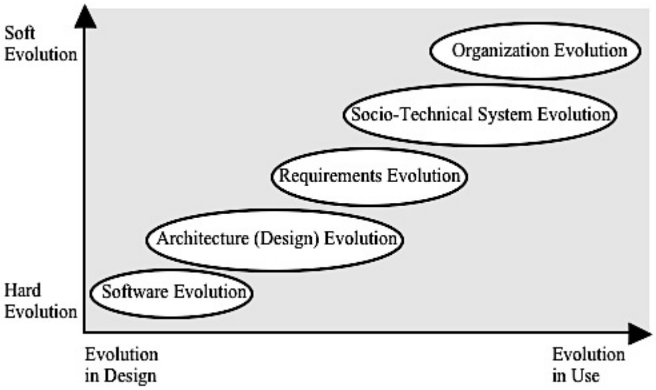

Fig. 3. Evolutionary space for socio-technical systems [9]

Felici [9] introduces a taxonomy of evolution in socio-technical systems that, as Fig. 3. shows, categorizes the different aspects of evolution into five classifications: *software evolution* that takes into account evolution from a product viewpoint, *architectural (design) evolution* that describes how system design captures evolution, *requirements evolution* that represents an intermediate viewpoint where stakeholders affect the evolution, *socio-technical system evolution* that takes into account evolution from a heterogeneous systemic viewpoint and *organization evolution* that emphasizes the interaction between socio-technical systems and surrounding environments [9]. The evolutionary space of this taxonomy is based on two dimensions. A temporal dimension from evolution in design to evolution in use that captures the system lifecycle perspective, and physical dimension from hard evolution to soft evolution that captures different system viewpoint in which evolution takes place [9].

This taxonomy can be applied to the evolution of Smart Grid software applications as a ULSS that in turn is a socio-technical system. In the following sub-sections the five evolutionary phenomena of this taxonomy for Smart Grid software applications are discussed and for each aspect, the possible challenges are presented. The evolution in this kind of systems is of a dual nature. While the whole collection of applications will be constantly subjected to gradual change, individual change drivers will also affect the individual systems [25]. However, the challenges in the evolution of individual software applications have been addressed before [23]. Therefore we concentrate on the challenges of software evolution in Smart Grid software applications at the system level.

A. Software Evolution

In this category the focus is on the software code evolution. *Ultra-large size*, *decentralized control* and *indistinct people/system boundary* attributes of Smart Grid software applications that come from their nature as a ULSS make the change of their code more challenging:

- Change impact analysis issue: In the Smart Grid context analyzing the change impact of one Smart Grid software application on another applications that are directly or indirectly related to that application is a matter. Current change impact analysis approaches mostly focus on analyzing the change impact at the level of single systems. So there is a need for developing new impact analysis approaches at more macro level [31] for Smart Grid software applications.
- Inconsistent bug fixing mechanisms: One of today's assumptions undermined by the decentralized control characteristic of ULSSs is that all conflicts must be resolved uniformly [32]. In Smart Grid software applications also it is impossible to resolve all conflicts and fixing all bugs by a central authority. Nevertheless, developing some approaches and mechanisms that can help different stakeholders to share their bug fixing experiences with each other would reduce the cost of bug fixing in the whole Smart Grid.
- Testing challenge: While change impact analysis is used to analyze the impact of a change before accomplishing that change, testing is conducted after a change is made in a system. Smith and Lewis [31] discuses the testing challenge for SoS that can be used for Smart Grid software applications as well. As they mention, in SoS context testing needs to consider a large number of users. It should also cover exception handling if a capability or service is not available. There is a strong need for integration testing too because the interoperation between different systems may provide capabilities that are not obvious from the individual system capabilities [31]. The situation in Smart Grid software applications is even more complicated because of their *indistinct user/systems and producers/consumers boundary* attribute. So there is a need to develop new or customized testing approaches for Smart Grid software applications including stress testing, load testing, integration testing, usability testing and so on.

B. Architecture (Design) Evolution

Design evolution describes how evolution is perceived at the design level [10]. *Decentralized control* and *heterogeneous elements* characteristics of Smart Grid software applications are the main reasons for their design evolution challenges. Due to these two characteristics, integration and interoperability are two major issues in design of Smart Grid software applications:

- Integration challenge: In Smart Grid, decentralized software applications that are developed by diverse stakeholders and third-party developers should be integrated to the whole system in order to perform their functionalities. These applications include legacy systems, different information sources and off-the-shelf components. So integrating these diverse and inconsistent systems is a crucial design task in Smart Grid that should be accomplished through a joint evolutionary approach.
- Interoperability issue: Inconsistent software applications in Smart Grid should also communicate each other to perform their functionalities. This can be happened through designing interfaces for such applications. The evolution challenge in this aspect arises when some applications change their interfaces.

Some useful approaches and technologies for handling both mentioned challenges are standardization, loose and indirect coupling design, having a commonly agreed-upon infrastructure and set of technologies, using middleware approach, service-oriented architecture, model-driven architecture and global information grid that are suggested by different authors for tackling evolution challenges in SoS/ULSS area [5][25][31][32]. Furthermore, open source and agile methodologies are two other areas can be helpful since they are not centralized development approaches [31][32]. All of the mentioned technologies and approaches would be used to develop approaches for handling design evolution challenges in Smart Grid software applications.

C. Requierments Evolution

Requirements evolution represents an intermediate viewpoint. Requirements are used as a means of interaction between stakeholders, thus represent a natural place where to capture information about the evolution of socio-technical systems [9]. *Unknown, conflicting and diverse requirements, decentralized control, indistinct people/system boundary* characteristics and also *virtual* nature of Smart Grid software applications make their requirements evolution challenging. As mentioned before when a SoS/ULSS is virtual, its acquisition is unplanned and it is not engineered from the scratch. Therefore in spite of traditional systems, requirements in SoS/ULSS are often negotiated rather than developed in the abstract [31]. The possible challenges for requirements evolution in ULSSs are mentioned in [32]. The situation is the same for Smart Grid software applications:

- User-centered requirements: As mentioned earlier, Smart Grid software applications are a social/conceptual ULSS and the scope of requirements gathering in them must expand to include people as consumers/producers of the system. Therefore user-centered requirements gathering approaches in the presence of diverse stakeholders should be considered for their evolution [32].
- Conflict management: Because Smart Grid software applications have different classes of users with different interests, it will be difficult to detect and resolve conflicts in emerging requirements for different system parts and stakeholders. Therefore there is a need to figure out how to represent, analyze, and reconcile distinct and competing interests [32].
- Requirements phase-out: The economics of Smart Grid software applications as a ULSS is such that phase-outs will often be far in the future. So there is a need to represent requirements in a way that accommodates the addition, deletion, modification, or recomposition of requirements over long periods of time and evolution [32].

D. Socio-Technical System Evolution

This category emphasizes the human aspects within evolution of socio-technical systems. A holistic view of such systems may identify and better understand the drivers for evolution and emphasizes how drivers for software evolution may reside outside software artefacts. Any evolution process requires some knowledge that is distributed among the system resources that are embedded in the environment of the system (e.g. socio-cultural, political, etc.) [10]. *Indistinct people/system boundary* and *new paradigms for acquisition and policy* characteristic of Smart Grid software applications link to their socio-technical system evolution challenges. Nevertheless there are also some opportunities coming from these characteristics that potentially allow Smart Grid software applications as a ULSS for analyses leading to types of improvement and adaptation that are not feasible in smaller systems [32]. For instance, with a sufficiently large number of interactions, which is possible in Smart Grid, the system can begin to gather data on observed regularities in people's behavior and begin to build statistically reliable models of what types of services will be required [32]. Architectural knowledge extracting is another socio-technical aspect that if be taken into account for Smart Grid software applications, would become another potential opportunity to improve the system's design evolution.

E. Organization Evolution

"The evolution of socio-technical systems influences the organizational context as well. Thus system evolution implies an organization co-evolution, and vice versa" [10]. *New paradigms for acquisition and policy* and *decentralized control* characteristics of Smart Grid software applications are the main source of their organization evolution challenges.

In SoS/ULSS context there is a need to develop communication patterns between stakeholders who have no formal organizational connection. Also there is a need to manage expectations among a group of stakeholders with widely different goals [31]. The supply chains of stakeholders and third-party developers that will populate ULSSs, such as Smart Grid software applications, must be organized to evolve system capabilities at a rapid pace in response to changing operational needs. Therefore, there is a need to understand how to manage these parties, ranging from large and established contractors to open-source communities and individual entrepreneurs, to achieve a level of cooperation and collaboration that can satisfy requirements for fast system evolution [32].

IV. CONCLUSION

The overall aim of this paper was identifying the possible challenges of evolution in Smart Grid software applications. For this purpose, Smart Grid software applications introduced as an ultra-large-scale system. Such a system has decentralized development and control, diverse and conflicting requirements that come from decentralized stakeholders, continues evolution and normal failures. Furthermore, the people are a part of system rather than just being users and as a result organizational policies become important. Since ULSSs has some characteristics with SoSs, this paper also applied characteristics and taxonomy of SoSs into Smart Grid context.

By employing the evolution challenges of ULSSs and SoSs, this paper introduced the possible challenges of software evolution in Smart Grid applications. Such challenges were presented within five evolution aspects: software evolution, design evolution, requirements evolution, socio-technical system evolution and organizational evolution. The introduced challenges would be a base for developing approaches and tools for improving management of evolution in Smart Grid software applications.

For the future research, one possible direction is going inside each evolution aspect and exploring more specific challenges of Smart Grid software applications in each category. Preforming several interviews with the companies supply software applications for Smart Grid would make the gathered challenges more practical than conceptual. Based on the results, some approaches could be developed and introduced to handle the challenges. Such approaches should be evaluated via conducting long-term case studies in the mentioned companies.

V. REFERENCES

[1] Arenas-Martínez, M., Herrero-Lopez, S., Sanchez, A., Williams, J. R., Roth, P., Hofmann, P. and Zeier, A., *A Comparative Study of Data Storage and Processing Architectures for the Smart Grid*, in Proceedings of 1st IEEE International Conference on Smart Grid Communications, October 2010.

[2] Beidou, F. B., Morsi, W. G., Diduch, C. P. and L. Chang., *Smart grid: Challenges, Research Directions and Possible Solutions*, in Power Electronics for Distributed Generation Systems (PEDG), 2nd IEEE International Symposium on, pp. 670 –673.

[3] Bré dillet, P. Lambert, E. Schultz, E. *CIM, 61850, COSEM Standards Used in a Model Driven Integration Approach to Build the Smart Grid Service Oriented Architecture,* in Proceedings of 1st IEEE International Conference on Smart Grid Communications, October 2010.

[4] Brewer, R. S. and Johnson, P. M., *WattDepot: An open source software ecosystem for enterprise-scale energy data collection, storage, analysis, and visualization,* in Proceedings of 1st IEEE International Conference on Smart Grid Communications, October 2010.

[5] Calinescu, R and Kwiatkowska, M., *Software Engineering Techniques for the Development of Systems of Systems*, 2008.

[6] Cecati, C., Mokryani, G., Piccolom, A and Siano, P., *An Overview on the Smart Grid Concept*. IEEE Industrial Electronics Society, pp. 3316-3321, 2010.

[7] Collier, S.E., *Ten Steps to a Smarter Grid*, IEEE Rural Electric Power Conference, REPC '09, pp. B2-B2-7, 2009.

[8] Cook, S. C., *On the Acquisition of Systems of Systems*, Proceedings of the 2001 INCOSE International Symposium, Melbourne AU, 2001.

[9] Felici, M., *Structuring Evolution: On the Evolution of Socio-Technical Systems*, in: Besnard, D., Gasek, C. and Jones, C.B., Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective, Springer-Verlag, 2006.

[10] Felici, M., *Taxonomy of Evolution and Dependability*, Workshop on Unanticipated Software Evolution, USE, 2003.

[11] *Functionalities of Smart Grids and Smart Meters,* EU Commission Task Force for Smart Grids, December 2010.

[12] Gabriel, R., Kazman, R., Northrop, L., Schmidt, D., and Sullivan. K., *Workshop on Software Technologies for Ultra-Large Scale Systems,* In Companion to the proceedings of the 29th International Conference on Software Engineering (ICSE COMPANION '07). IEEE Computer Society, Washington, DC, USA, pp. 140-141, 2007.

[13] Gideon J.M., Dagli C.H. and Miller, A., *Taxonomy of System-of-Systems,* In Proceedings of CSER Conference on Systems Engineering Research, Hoboken, New Jersey, 2005.

[14] Hommelberg, M., Warmer, C., Kamphuis, I., Kok, K. and Schaeffer, G., *Distributed Control Concepts using Multi-Agent technology and Automatic Markets: An indispensable feature of smart power grids,* Proceedings IEEE Power Engineer Society General Meeting, June 2007.

[15] Ipakchi, A., *Implementing the Smart Grid: Enteriprise Information Integration,* In Proceedings of GridWise Grid-Interop Forum, Albuquerque, NM, 2007.

[16] Kotov, V., *Systems of System as Communicating Structures*, Hewlett Packard Computer Systems Laboratory Paper HPL-97-124, pp1-15, 1997.

[17] Kremers, E., Viejo, P., Barambones, O. and Gonzalez de Durana, J., *A Complex Systems Modelling Approach for Decentralised Simulation of Electrical Microgrids,* In Proceedings of the 2010 15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '10). IEEE Computer Society, Washington, DC, USA, pp. 302-311, 2010.

[18] Lane, J. A., Understanding Differences Between System of Systems Engineering and Traditional Systems Engineering, PhD Qual Exam Report, University of South California, 2007.

[19] Langeland, T. and Greiner, C. (2011). *The Smart Grid – What It Is and What It Is Not*, Fremtiden Er Elektrisk, NEF Teknisk Møte, Trondheim, March 2011.

[20] Lehman, M. and Fernandez-Ramil, J.C., *Software Evolution*, in: Madhavji, N. H., Fernandez-Ramil, J. C. and Perry, D. E., *Software Evolution and Feedback: Theory and Practice,* John Wiley and Sons, Ltd, 2006.

[21] Liu,W-H.E., *Analytics and Information Integration for Smart Grid Applications,* Power and Energy Society General Meeting, 2010 IEEE , pp.1-3, July 2010.

[22] Maier, M. W., *Architecting Principles for Systems-of-Systems,* Systems Engineering, 1: 267–284, 1998.

[23] Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R., Jazayeri, M., *Challenges in Software Evolution,* in Eighth International Workshop on Principles of Software Evolution, pp. 13- 22, 5-6, 2005.

[24] Mirakhorli, M., Sharifloo, A. A. and Shams, F., *Architectural Challenges of Ultra Large Scale Systems,* In Proceedings of the 2nd international workshop on Ultra-large-scale software-intensive systems (ULSSIS '08). ACM, New York, NY, USA, pp. 45-48, 2008.

[25] Mittermeir, R. T., *Facets of Software Evolution,* in: Madhavji, N. H., Fernandez-Ramil, J. C. and Perry, D. E., *Software Evolution and Feedback: Theory and Practice,* John Wiley and Sons, Ltd, 2006.

[26] Rezaei Rad, B. and Shams Aliee, F., *Computational Grid as an Appropriate Infrastructure for Ultra Large Scale Software Intensive Systems*. In Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, 2010.

[27] Safuto, R., *When Smart Grid Meets Cloud Computing*, Clean Energy Digest, December 2009.

[28] Santacana, E., Rackliffe, G., Tang, L., and Feng, X., *Getting Smart*, Power and Energy Magazine, IEEE, 8(2), pp. 41–48, 2010.

[29] Shaw, M., *Empirical Challenges in Ultra Large Scale Systems,* In Proceedings of ESEM, 2008.

[30] Simmhan, Y., Giakkoupis, M., Cao, B. and Prasanna, V. K., *On using cloud platforms in a software architecture for smart energy grid*, in IEEE International Conference on Cloud Computing (CloudCom), December 2010.

[31] Smith, D. and Lewis, G., *Systems of Systems: New challenges for Maintenance and Evolution*, Frontiers of Software Maintenance, 2008. FoSM 2008, pp. 149-157, 2008.

[32] Software Engineering Institute, *Ultra-Large-Scale Systems: The Software Challenge of the Future,* Pittsburg, 2006.

[33] Warmer, C., Kok, K., Karnouskos, S., Weidlich, A., Nestle, D., Selzam, P., Ringelstein, J., Dimeas, A. and Drenkard. S., *Web services for integration of smart houses in the smart grid,* In Grid-Interop Conference, Denver, November 2009.

[34] Wei, C., *A Conceptual Framework for Smart. Grid,* Power and Energy Engineering Conference. (APPEEC), 2010.

VI. BIOGRAPHIES

Mohsen Anvaari is a PhD candidate in the Department of Computer and Information Science at the Norwegian University of Science and Technology since 2011. He received his Master's degree in Software Engineering and Management from University of Gothenburg in Sweden. His research interests are in software architecture, software evolution and applied software engineering. Contact him at mohsena@idi.ntnu.no.

Daniela S. Cruzes is a postdoctoral fellow in the Department of Computer and Information Science at the Norwegian University of Science and Technology since 2009. Previously, she worked as a researcher fellow at the University of Maryland and Fraunhofer Center for Experimental Software Engineering-Maryland. Cruzes received her PhD in experimental software engineering from the University of Campinas - UNICAMP in Brazil. Her research interests are in empirical software engineering, research methods and theory development, systematic reviews, synthesis of SE studies, software evolution and open source software. Contact her at dcruzes@idi.ntnu.no.

Reidar Conradi is a professor in the Department of Computer and Information Science at the Norwegian University of Science and Technology. His research interests are in software quality, software process improvement, version models, software evolution, open source software, and associated empirical studies. Conradi received his PhD from the Norwegian University of Science and Technology. Contact him at conradi@idi.ntnu.no.