# Product Line Architectures for SOA

**Mercy N. Njima**[1], **Maurice H. ter Beek**[2], **and Stefania Gnesi**[2]
[1]IMT Institute for Advanced Studies, Lucca, Italy
[2]Istituto di Scienza e Tecnologie dell'Informazione, ISTI–CNR, Pisa, Italy

**Abstract**—*Service-oriented applications (SOA) are a standard-based and technology independent distributed computing paradigm for discovering, binding and assembling loosely-coupled software services. Software product lines (SPL) on the other hand allow a generic architecture to be configured and deployed in different instances. Product lines facilitate systematic reuse through managing variability. Product line engineering is a more established discipline and so may have more solutions to offer SOA and SPL. Thus, in this paper, we will look the synergies accruing from this powerful combination. We will then look at how the technique can be used to evaluate the latest smart energy management innovation in delivering return on investment for utilities.*

**Keywords:** Service Orientation, Product Lines, Smart Grid

## 1. Introduction

Product line technology, is increasingly finding its way to the software sector allowing companies to sustain growth and achieve market success [2]. The combination of Software Product Line (SPL) and Service-Oriented Application (SOA) development practices is becoming a new development paradigm that helps provide the answers to the need for agility, versatility and economies. SOA and SPL approaches to software development share a common goal. They both encourage an organization to reuse existing assets and capabilities rather than repeatedly redeveloping them for new systems. These approaches enable organizations to capitalize on reuse to achieve desired benefits such as productivity gains, decreased development costs, improved time to market, higher reliability, and competitive advantage [3]. Service Product Lines will achieve flexibility of network based systems through service orientation, but still manage product variations through product line engineering techniques [5]. Service features are selected and/or parameterized at runtime by a user or by a product itself when a certain contextual change or a new service provider is recognized. Service Orientation (SO) accommodates rapidly changing user needs and expectations. Adopting SO in practice for real software and system development has uncovered several challenging issues, such as:

- how to identify services,
- determining configurations of services that are relevant to users',
- current context, and
- maintaining system integrity after configuration changes.

Given that product line engineering approaches have been in the market for a longer period in the manufacturing industry they can be used to address these issues.

The paper is structured as follows. In Section 2, we review some work done to bring together software product lines and service oriented architectures. In Section 3 we present a successful approach to also apply the synergy to a real world problem. Section 4 introduces our case study in which we apply such techniques whilst Section 5, showcases variability concepts in the case study. We conclude in Section 6 with some remarks on future work.

## 2. Related work

There has been efforts to introduce the concepts of software product lines and service orientation: [2] explores their parallels and shows the applicability of software product line methods to service orientation. It also discusses the main obstacles to realizing the synergy between these cutting-edge technologies.

In [7], the authors outline the objectives of a workshop organized to explore how service-oriented architectures and software product lines can benefit from each other. Specifically they outline questions on

- how service-oriented systems can benefit from software product lines' variation management approaches to identify and design services targeted to multiple service-oriented systems and

- how software product lines can benefit from service-oriented architectures by employing services as a mechanism for variation within a product line.

The output from this workshop covers the area very well but it is beyond the scope of this paper to delve further.

There's also been work done in the application of these concepts to diverse domains from which we were motivated. Shokry et al. [12] apply the notion of dynamic runtime variability of software product lines in the embedded automotive software systems to create adaptable service-oriented architectures while [10] proposes an approach for service-oriented product line architectures that combines SPL and SOA concepts and techniques to achieve high customization and systematic planned reuse. They apply the techniques to conference management domain case study shown Figure 1.

The case study consists of a service-oriented product line that intends to produce customized service-oriented applications for the management of different conferences. Section 3 explores this work further in order to form a basis for our own case study.

## 3. Approach

The motivation for Medeiros et al. [10] stems from the fact that SOA lacks support for high customization and systematic planned reuse. In other words, despite the natural way of achieving customization in service-oriented applications, changing service order or even the participants of service compositions, services are not designed with variability to be highly customizable and reusable in specific contexts. Moreover, service artifacts, e.g., specifications and models, are not designed with variability as well. Hence, these artifacts cannot be easily reused by a family of service-oriented applications. Thus, SPL engineering, which has the principles of variability, customization and systematic planned reuse in its heart, can be used to aid SOA to achieve these benefits. In this way, service-oriented applications that support a particular set of business processes can be developed as SPLs. The aim is to achieve desired benefits such as productivity gains, decreased development costs and effort, improved time to market, applications customized to specific customers or market segment needs, and competitive advantage.

The approach for service-oriented product line architectures starts with an identification phase. It receives the feature model and the business process models as mandatory inputs, and produces a list of possible components, service candidates and service orchestration candidates for the product line architecture. Thus, these architectural elements can be reused in all products of the line. This phase is separated in component identification and service identification activities. Subsequently, there is a variability analysis activity. It receives the list of components and services identified previously, and defines and documents key architectural decisions regarding variability. In this activity, it is defined how the variability will be implemented within the services and components. Architecture specification activity concludes the approach. In this activity, the architecture is documented using different views in order to represent the concerns of the different stakeholders involved in the project. The approach takes the following steps:

- Component Identification: the components of the service-oriented product line are identified. A software component is a self-contained artifact with well-defined interfaces and is subject to third-party compositions. This activity starts with an analysis of the feature model to identify architectural component candidates. The purpose of this activity is to put features into modules (components) in order to design an architecture where components can be added or removed to generate customized products. Each of the modules identified in this activity will be an architectural component candidate for the service-oriented product line architecture.
- Service Identification: a set of service and service orchestration candidates that support the business processes are identified. Thus, as the services are supposed to support the business processes, it is reasonable to identify them from the business process models.
- Variability Analysis: variability is the ability to change or customize software systems. Improving variability in a system implies making it easier to do certain kinds of customizations. In addition, it is possible to anticipate some types of variability and construct a system in such a way that it is prepared for inserting predetermined changes. During the variability analysis activity, essential

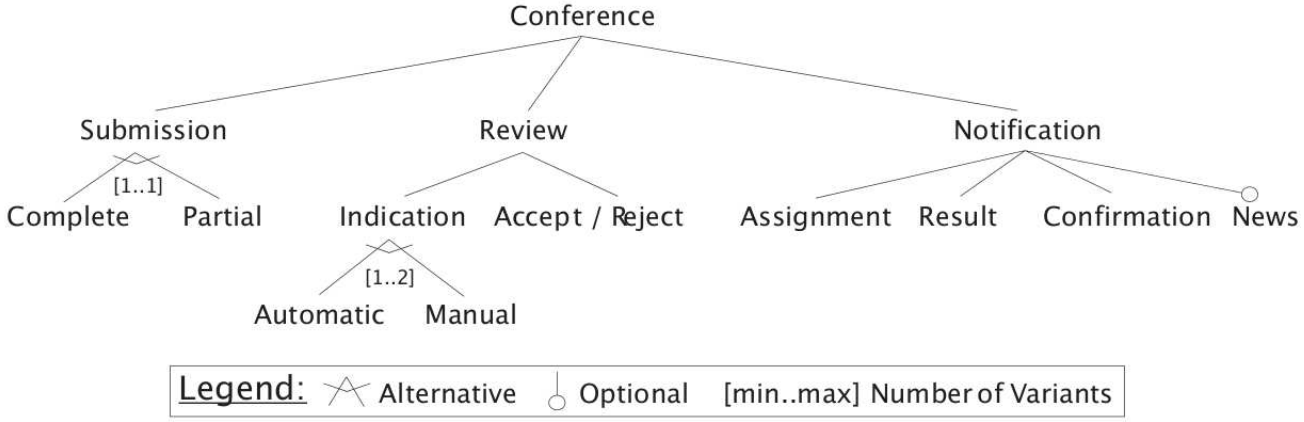

Fig. 1: Feature Model for the Conference Management Domain

architectural decisions about how the variability presented in the feature model and business processes will be implemented within services and components are defined.

- Architecture Specification: in this activity, the models and specification are produced with variability as all the artifacts of core assets development.

The model contains the following features:

- Submission: authors can submit their complete papers or, first submit the abstract, followed by the complete version. Complete and partial submissions are alternative features.
- Review: the indication of papers to reviewers can be made automatically and/or manually. Reviewers can also accept or reject paper indications. Automatic and manual indications are not exclusive, they can work together.
- Notification: the system can send information to reviewers about paper assignments. It can send acceptance or rejection (result) information to authors. It can also send event news, e.g., deadlines, and confirmation messages, e.g., paper or review submitted, to authors and reviewers. Event news notification is an optional feature. Assignments, confirmation and result notifications are mandatory.

## 4. Motivating Problem: The Smart Grid

Most of the world's electricity system was built when primary energy was relatively inexpensive. Grid reliability was mainly ensured by having excess capacity in the system, with unidirectional electricity flow to consumers from centrally dispatched power plants. Investments in the electric system were made to meet increasing demand – not to change fundamentally the way the system works. While innovation and technology have dramatically transformed other industrial sectors, the electric system, for the most part, has continued to operate the same way for decades. This lack of investment, combined with an asset life of forty or more years, has resulted in an inefficient and increasingly unstable system [4].

However, climate change, rising fuel costs, outdated grid infrastructure, and new power-generation technologies have changed the mindset of all stakeholders:

- Electric power causes approximately 25% of global greenhouse gas emissions, and electris utility companies are rethinking what the electricity system of the future should look like.
- Renewable and distributed power generation will play a more prominent role in reducing greenhouse gas emissions.
- Demand-side management promises to improve energy efficiency and reduce overall electricity consumption.
- Real-time monitoring of grid performance will

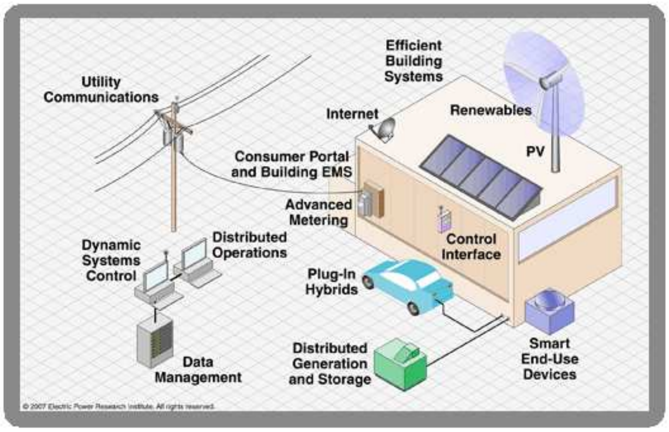

Fig. 2: Components of the smart Grid. Adopted from smartgrid.epri.com

improve grid reliability and utilization, reduce blackouts, and increase financial returns on investments in the grid.

These changes on both the demand and supply side require a new, more intelligent system that can manage the increasingly complex electric grid, see Figure 2. The energy community is starting to marry information and communications technology (ICT) with electricity infrastructure to enable the electric system to become ‘smart’. Thus, the Smart Grid is the next-generation, managed electrical power system that leverages increased use of communications and information technology in the generation, delivery, and consumption of electrical energy. It consists of solutions based on both current and future telecommunication technologies for command and control, metering, and charging [9].

Electric utilities, in a reactive or proactive answer to these new challenges, are adding more intelligence and complexity in their distribution networks. As the grid becomes more intelligent and more complex, see Figure 3, the tools to operate it become increasingly important. Hence the need for interoperability from SOA and flexibility and variability from SPL [11]. The end result is then electricity provision as a service and the Smart Grid as a service product line.

In addition, it is essential to utilize product line technology to manage energy from the consumers point of view. This can be achieved by gathering metrics that report:

- Energy usage by product family;
- Real-time power draw by product;
- Energy consumption by device.

Services can then be used to achieve energy efficiency by helping customers virtualize infrastructure and perform energy efficiency benchmarking to effectively measure, monitor, and fine-tune energy usage. Sharing detailed analysis, recommendations, and design expertise services can also help energy management staff better understand, optimize, and control power to achieve significant cost savings [1].

## 5. Variability Modeling of Smart Grids

We introduce a feature-based approach to product line engineering and illustrate how this approach addresses the technical issues discussed relating to the smart grid. In this approach, a feature model (which captures the commonality and variability information of a product line) and feature-binding information about which features are included in products and delivered to customers, and when, are used as primary input to production plan development [8]. In product line engineering, a feature model plays a central role in the management and configuration of multiple products. Therefore, core assets should be identified based on features.

Feature Diagrams are a family of popular modeling languages used for engineering requirements in SPL represented as the nodes of a tree, with the product family being the root and have the following features:

- optional features may be present in a product only if their parent is present;
- mandatory features are present in a product if and only if their parent is present;
- alternative features are a set of features among which one and only one is present in a product if their parent is present [6].

We model the generic Smart Grid architecture as a family with basic components for basic products and specialized properties for some of the products such as:

- storage: will typically interconnect to the grid as a whole rather than being tied to balancing the output from a specific source;
- renewables: varies with weather, time, season and other intermittent effects. Since the grid is built to handle a lot of demand volatility and uncertainty we have to understand the impact of variable generation on grid reliability;

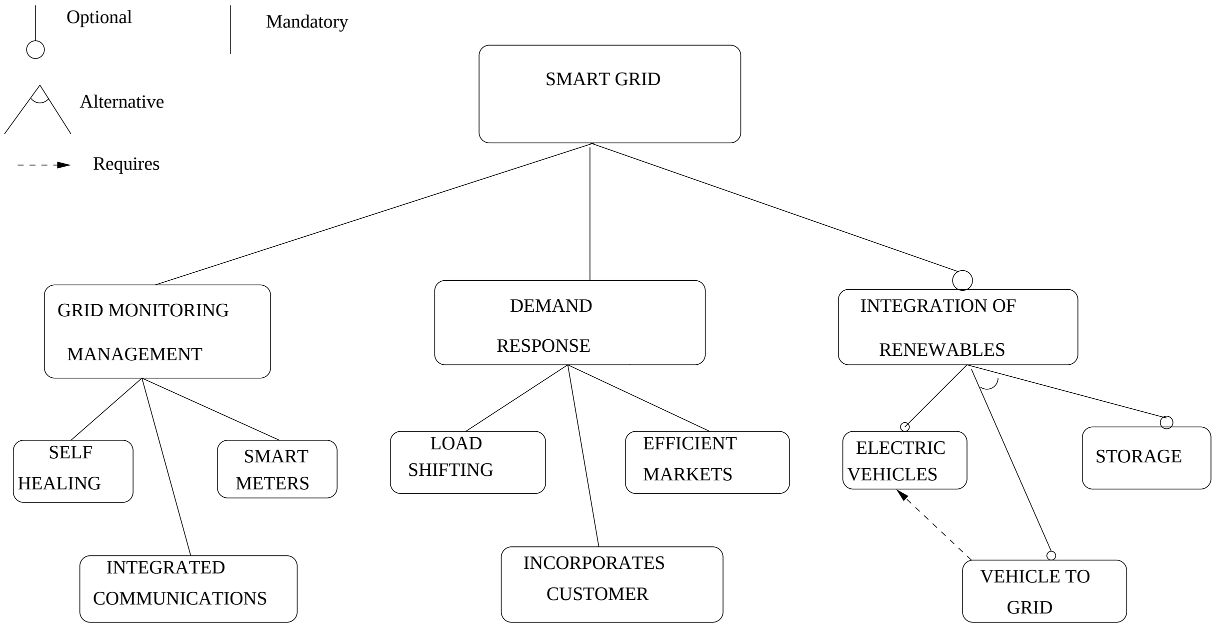

Fig. 3: Feature Model for the Smart Grid Family

- load shifting: practice of managing electricity supply and demand so that peak energy use is shifted to off-peak periods;
- vehicle to grid: establishing a viable transparent business model, guaranteeing the availability and controllability of electric vehicles and vehicle to grid (V2G) capacity, accurate forecasting of renewable energy supply and demand.

Load shifting and V2G can reduce the energy storage capacity required to maintain power quality.

Figure 4 shows the feature model for the smart grid family from which we can derive many other products depending on the needs of the consumer. We have the additional constraint ‘requires’ added to a feature diagram which is a unidirectional relation between two features indicating that the presence of one feature requires the presence of the other. We observe that features increment a product’s functionality and more specifically that:

- the requires constraint obligates feature electric vehicles to be present whenever vehicle to grid (V2G) is.

Unfortunately, as the size of the model grows it becomes more difficult to derive products and deduce their behaviour. Furthermore, we see that most of the system behaviour modeled is static.

To counter this deficiency, we take advantage of the flexibility that comes with service orientation by introducing independently configurable service features [8]. A service feature represents a major functionality of the system that may be added or removed as a unit. From a SOA perspective, it can be viewed as an orchestrator of services and in this case products, in order to derive as many products as possible by initiating them as services. The service features in our smart grid model are grid monitoring management, demand response and integration of renewables all of which are independently configurable.

## 6. Conclusion and Future work

The realization of product line architectures for service oriented architectures is a very challenging issue for designing and developing systems required in the business environment. In this paper, we have pointed out examples from the automotive domain, conference management and built an energy management system from a static behavioural perspective. We have proposed that a service oriented approach can be exploited for such to support the flexibility of features.

We found that our approach would provide electric utility companies with an explicit way to develop their electricity provision products, organize their assets and take care of essential communication with customers. They would also be able to build specific products for

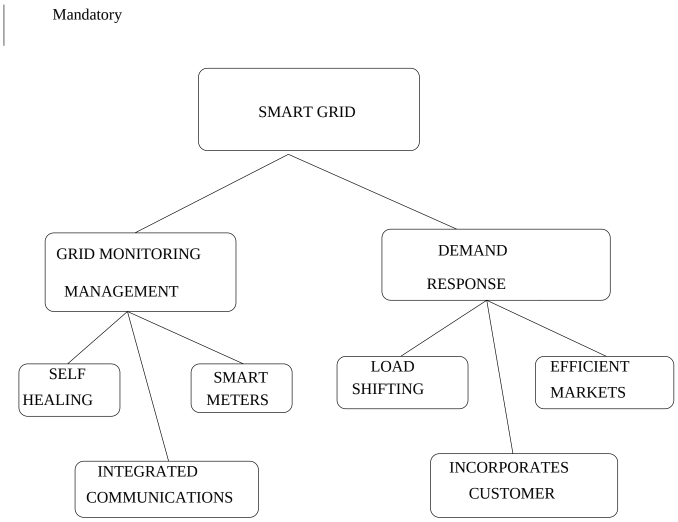

Fig. 4: Feature Model for a product without integration of renewables

different customers using an already existing architecture and thus, save time and bring such products to market quickly and efficiently.

Future work in this line of research will include identifying suitable models for describing feature oriented analysis in order to manage dynamic behaviour. Such dynamic requirements may come about when a consumer requests real-time assistance in managing their electricity load. From the utility's point of view, there maybe an oversupply of energy during off peak hours and they may need to take advantage of pre-existing service agreements with their consumers to utilise more. In addition, we are interested in working on storage using electric vehicles and the vehicle to grid feature given the complexity that comes from the requirement for the system to evolve adaptively according to changing needs.

## References

[1] R. Aldrich and G. Mellinge. Cisco energy management: A case study in implementing energy as a service. `http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps10195/CiscoEMSWhitePaper.pdf`.

[2] O. Bubak and H. Gomaa. Applying software product line concepts in service orientation. *International Journal of Intelligent Information and Database Systems 2008*, 2(4):383–396, 2008.

[3] S.G. Cohen and R.W. Krut, editors. *Proceedings of the 1st Workshop on Service-Oriented Architectures and Software Product Lines: What is the Connection? (SOAPL'07)*, Technical Report CMU/SEI-2008-SR-006, Carnegie Mellon University, 2008.

[4] C. Feisst, D. Schlesinger, and W. Frye. Smart grid: The role of electricity infrastructure in reducing greenhouse gas emissions. `http://www.cisco.com/web/about/ac79/docs/wp/Utility_Smart_Grid_WP_REV1031_FINAL.pdf`.

[5] M. Galster. Describing variability in service-oriented software product lines. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, pages 344–350, New York, NY, USA, 2010. ACM.

[6] K. Kang, S. Choen, J. Hess, W. Novak, and S. Peterson. Feature oriented domain analysis (FODA) Feasibility Study. Technical Report SEI-90-TR-21, Carnegie Mellon University, 1990.

[7] R. W. Krut and S. G. Cohen. Service-oriented architectures and software product lines: enhancing variation. In *Proceedings of the 13th International Software Product Line Conference*, SPLC '09, pages 301–302, Pittsburgh, PA, USA, 2009. Carnegie Mellon University.

[8] J. Lee, K. C. Kang, and S. Kim. A feature-based approach to product line production planning. In Robert Nord, editor, *Software Product Lines*, volume 3154 of *Lecture Notes in Computer Science*, pages 137–140. Springer Berlin / Heidelberg, 2004.

[9] E.M. Lightner and S.E. Widergren. An orderly transition to a transformed electricity system. *IEEE Transactions on Smart Grid*, 1(1):3 –10, jun. 2010.

[10] F. M. Medeiros, E. S. d. Almeida, and S. R. d. L. Meira. Towards an approach for service-oriented product line architectures. In *Proceedings of the Workshop on Service-oriented Architectures and Software Product Lines, San Francisco, CA*, 2009.

[11] D. Savio, L. Karlik, and S. Karnouskos. Predicting energy measurements of service-enabled devices in the future smartgrid. *International Conference on Computer Modeling and Simulation*, pages 450–455, 2010.

[12] H. Shokry and M. A. Babar. Dynamic software product line architectures using service-based computing for automotive systems. In *SPLC (2)*, pages 53–58, 2008.

[13] S. Trujillo, S. Apel, and Kaestner C. Product Lines that supply other Product Lines: A Service-Oriented Approach. In *Workshop on Service Oriented Architectures and Product Lines - What is the Connection? (SOAPL 2007) at the 11th International Software Product Line Conference (SPLC 2007), Kyoto, Japan, Sep*, 2007.