Comparing Specification and Design Approaches for Power Systems Applications

Claudia Zanabria*, Filip Pröstl Andrén*, Thomas I. Strasser*†
*Center for Energy – Electric Energy Systems, AIT Austrian Institute of Technology, Vienna, Austria
{claudia.zanabria, filip.proestl-andren, thomas.strasser}@ait.ac.at
†Institute of Mechanics and Mechatronics, Vienna University of Technology, Vienna, Austria
thomas.strasser@tuwien.ac.at

***Abstract*—The upcoming large-scale integration of distributed energy resources into power distribution grids motivates the implementation of advanced control applications to ensure security of supply and power quality. The realization of such increasingly complex applications is also characterized by higher engineering efforts, even resulting in increased total life-cycle costs. However, by using proper method and corresponding tools there is a huge optimization potential for the engineering process. Various approaches and tools with different characteristics have been developed so far in order to provide support along the engineering and validation process of power system applications. However, it is not always clear which tools are the best choice for the different development steps. In order to overcome this issue this work provides a comparison of specification and design approaches which are being commonly used in the domain of power and energy systems.**

***Index Terms*—Power and energy systems, engineering process, specification, design, UML, SysML, IntelliGrid, SGAM, PSAL, EMSOnto, MATLAB/Simulink.**

I. INTRODUCTION

The upcoming large-scale integration of distributed energy resources (Photovoltaic (PV), wind, etc.) into power distribution grids motivates the implementation of advanced control mechanisms to ensure security of supply and power quality [1], [2]. Additionally, new energy service markets [3] are being developed to provide incentives for increased support. The realization of these complex applications becomes further complicated since it must tackle diverse domains, from end-customer to transmission network, involving the participation of several stakeholders and electrical devices. This motivates the research of innovative approaches for handling the engineering complexity in smart grids.

But, new approaches also introduce new challenges. The realization of complex solutions are associated with increasing engineering complexity, which results in higher total life-cycle costs. However, with the usage of proper approaches, architectures, and tools there is a huge optimization potential for the engineering process [4]. Various tools with different characteristics have been developed so far in order to provide support along the engineering and validation process of power system applications. However, it is not always clear which tools are the best choice for the different development steps. In order to overcome this issue this work provides a comparison of specification and design approaches which are being commonly used in the domain of power systems.

This paper is organized as follows: An overview of covered use cases and applications as well as an outline of the main phases of the engineering process is give in Section II. Typical specification and design approaches are being introduced in the following Section III whereas the proposed comparison of those approaches is provided in Section IV. The main findings of this work are summarized in Section V.

II. ENGINEERING POWER SYSTEM APPLICATIONS

The assessment of potential solutions for problems emerging in power systems need to be carefully studied and analyzed. Typically, the realization of these solutions follow a number of tasks: *(i)* specification, *(ii)* design, *(iii)* prototype, *(iv)* implementation, and *(v)* validation. This section focuses on the first two stages involved in this development process.

A. Use Cases and Applications

A comprehensive study and classification of power system services is provided by [1]. Table I briefly describes three

TABLE I: Selected power system Use Cases (UC).

Use Case	Description
UC1: Frequency Control (open-loop)	The frequency of the grid is stabilized along the transmission and distribution network. The responsible entity is the TSO. Other actors involved are DSOs, customers (interruptible loads), DERs (overfrequency), and power plants. The time requirements to stabilize the deviation of frequency varies from seconds to about 15 min (e.g., primary frequency control-PCR <30 s) [2]. A control algorithm that performs PCR is detailed in IEC 61850-90-7 [5] (i.e., FW21).
UC2: Voltage Control (closed-loop)	Obligations for maintaining voltage levels are shared between TSO and DSOs. Other actors involved are aggregators, DER, end-customer and power plants. This service should be performed at time scales from milliseconds to about 30 min [1]. A voltage control strategy that relies on a proportional gain and integrator controller is defined in [6].
UC3: Min. of Energy Costs & Peak Shaving (opt. function)	Energy costs to be paid by end-customers is minimized by integrating distributed generation and energy storage systems. A peak shaving service is performed by managing power flows. Both services are described in [7], where a control system based on dynamic programming is implemented.

commonly used Use Cases (UC). The use cases also highlight different "control type" patterns: *(i)* open-loop, *(ii)* closed-loop, and *(iii)* optimization function.

By combining these UCs into one application a multi-functional control scheme is created. The control application resulting from this combination is used to compare and evaluate the performance of the specification and design approaches addressed in this paper.

B. Specification and Design Process

This paper concentrates efforts on the first steps of the realization of power system use cases—the specification and design phase. At the specification stage the main problem to be resolved is analyzed, which entails the identification of concerned actors and potential solutions. The physical architecture of power systems is also specified as well as communication between electrical devices and systems (monitoring system, substation operator, etc.). An outcome from this process is a list of requirements that are used during the design phase.

During the design phase, control strategies are developed. It implies the analysis of their behavior and structure to be proofed in a further step. Furthermore, information exchanged in the control applications is detailed, which means that measurements and control variables are identified. This also involves communication links across distributed energy resources. Moreover, aspects about the power system architecture such as configuration of physical interfaces and ICT components are given. Additionally, a matching between control algorithms and specific hardware device is done.

Methodologies to specify and design control applications are studied in [8]. At a later stage, the behavior of smart grid applications is modeled, tested, and validated. Once the proof-of-concept is achieved, the deployment of the use cases into real hardware device and/or software artifact is done. Since some of the approaches studied in this paper also support (semi-)automatic generation of software artifacts, this is also included in the evaluation.

III. COMMONLY USED APPROACHES AND TOOLS

During the whole engineering process—as outlined above—different approaches and tools are being applied. In the following sections, a selected set of them, which are being commonly used for specifying and designing applications in the domain of power and energy systems, are briefly described and characterized.

A. Unified Modeling Language

The Unified Modeling Language (UML) has been adopted by the Object Management Group (OMG) as a standardized, general-purpose language for modeling software systems [9]. Nowadays, UML is also published as an ISO standard. UML provides several diagrams and approaches in order to describe the structure (object, component, class, etc.) and behavior (use case, activity, state machine, interaction, etc.) of a software system. Nowadays, UML is also used in the domain of power and energy systems to model and describe corresponding software applications.

B. System Modeling Language

Similar to UML, the Systems Modeling Language (SysML) has been developed as a general-purpose modeling approach. Instead of describing software systems this language is used for modeling whole technical systems and corresponding applications. Therefore, a broad range of systems(-of-systems) can be specified, analyzed, and designed. The SysML language is an extension of a subset of UML and was mainly designed to support systems engineering activities [10].

C. IntelliGrid (IEC 62559)

The IntelliGrid approach was originally developed by EPRI in 2003 as a response to the increasing complexity of power system automation [11]. Since then, IntelliGrid has become its own standard (IEC 62559 [12]), which is one of the most commonly used methods for describing smart grid use cases. It integrates requirements engineering and best practices, and also explicitly addresses the identification of stakeholders and how to structure communications in a project [13].

The core of IEC 62559 identifies five engineering phases that mainly concern the development and identification of business cases, use cases, and requirements. To support this a use case template is provided. The use case itself is described through a narrative as well as a visual representation (e.g., a UML diagram). Furthermore, a detailed step-by-step description of the use case is provided [12].

D. Smart Grid Architecture Model

Initially the Smart Grid Architecture Model (SGAM) was mainly intended for the coordination of standardization activities but it provides also a structured approach for modeling of advanced power system use cases. Therefore, SGAM provides a three-dimensional framework consisting of *(i)* domains, *(ii)* zones, and *(iii)* layers [14]. Combined together, this framework creates a three-dimensional model.

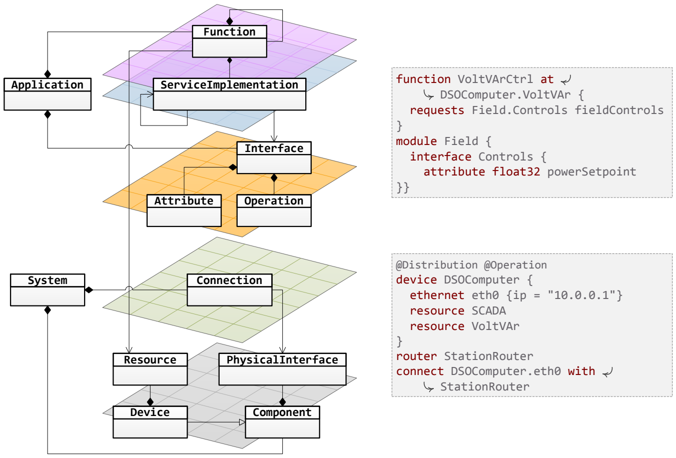
Accompanying, the SGAM framework is also a use case design methodology which is based on the IEC 62559 use case template which is mapped into the different SGAM layers. In order to do this in a structured way, the approach defines a number of design steps [15]. One of the main advantages of this approach is its coordinated set of viewpoints. It allows to depict various interrelated aspects of smart grid architectures and using the different viewpoints it is easier to identify interoperability issues [16].

For supporting the user the so-called "SGAM Toolbox (SGAM TB)" was developed which is a UML-based Domain Specific Language (DSL) available as an extension to the Enterprise Architect software [17]. With this tool in hand, all steps in the SGAM methodology are covered in one environment; which provides also code generation capabilities.

E. Power System Automation Language

The Power System Automation Language's (PSAL) intention is to provide a formalized language for SGAM compatible use case design and at the same time it allows rapid development of automation, control, and ICT functions of power system applications [4], [18]. Thus, PSAL not only supports the development of high-level use case descriptions, it also offers tools for detailed use case specifications. In a further step code generation is also supported.

A main requirement for PSAL was rapidness. For example, it is often the case that one type of application needs to be implemented for different infrastructure configurations (e.g., another distribution grid). In order to support this, PSAL introduces an extra abstraction layer, containing a `System` and an `Application`. The `System` consists of the component and the communication layers of SGAM, while the `Application` contains the business, function, and information layers. This allows the user to define an `Application` independently from the `System`. PSAL is a textual language, but Fig. 1 shows a UML representation of the PSAL metamodel as well as example implementations of an `Application` and a `System`.



```
function VoltVArCtrl at ↩
    ↳ DSOComputer.VoltVAr {
  requests Field.Controls fieldControls
}
module Field {
  interface Controls {
    attribute float32 powerSetpoint
}}
```

```
@Distribution @Operation
device DSOComputer {
  ethernet eth0 {ip = "10.0.0.1"}
  resource SCADA
  resource VoltVAr
}
router StationRouter
connect DSOComputer.eth0 with ↩
    ↳ StationRouter
```

(a) (b)

Fig. 1: Overview of the PSAL engineering approach: (a) UML representation, (b) example code [18].

One of the main ideas with PSAL is that it should allow rapid generation of code and configurations, such as executable IEC 61499 code and IEC 61850 configurations [18].

F. Energy Management System Ontology

Currently, the identification out of inconsistencies at the design phase is a feature that is usually not supported by traditional methodologies [12], [15]. This asset would save implementation efforts at the prototype and design stages.

The Energy Management System Ontology (EMSOnto) approach is conceived to investigate inconsistencies on the list of requirements stated at the design phase and to support the implementation and proof-of-concept phases by an automatic generation of software artifacts [19]. To this end, EMSOnto uses an ontology (i.e., EMS-ontology), as well as the definition of rules and a reasoner engine to infer new knowledge [20]. Resulted information is queried by SPARQL queries [21]. The first step carried out by implementing EMSOnto is based on the population of the EMS-ontology; it means assertions regarding a power system use case are gathered. Hence, this approach proposes spreadsheet templates (EMS-template) that are manually fulfilled by domain experts. Moreover, an automatic pre-fill of those templates is reached by exchanging knowledge with smart grid data models, as shown in Fig. 2.

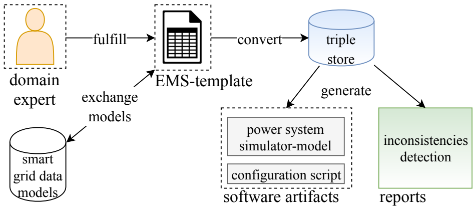


Fig. 2: Overview of the EMSOnto approach.

EMS-templates facilitate the handling, organization and storage of a huge amount of data at the design phase. Additionally, provision of models for control functions (e.g., frequency and voltage regulation [5]) and models of electrical devices, such as energy storage and DERs are supported within the EMS-template. This simplifies the collection of data. The EMSOnto approach together with MATLAB/Simulink also enables an automatic generation of artifacts within the implementation phase [22].

G. MATLAB/Simulink

MATLAB/Simulink is a general-purpose simulation tool. With proper toolboxes and libraries like SimPowerSystems or Simscape Power System (i.e., powersim) it can be also used to model and simulate electrical power systems and corresponding components (inverters, batteries, power grid, transformer, etc.). With additions it supports also the implementation stage by providing C-code generation and enabling the simulation of models in other software/hardware platforms. Thereby, Controller Hardware-in-the-Loop (CHIL) test can be carried out with less manual effort.

IV. COMPARISON OF APPROACHES

The scope of the proposed analysis mainly covers the specification and design stage. Hence, the approaches presented above were applied to these stages followed by a comparison of the results.

A. Applied Methodology and Results

In order to make a proper comparison, each approach was applied to specify and design the same use cases, namely the use cases presented in Section II-A. The evaluation of performance of each approach is carried out by analyzing the fulfillment of certain specification and design features for each SGAM layer. For instance, at the specification phase the definition of control algorithms, constraints to be satisfied, and control and measurement variables are features for the function layer. Those features and others are explained for each layer in the following sections. The study also provides an outlook towards the implementation phase. Here, only rapid prototyping mechanisms are evaluated. An exhaustive evaluation is not possible since most of the approaches do not cover this stage. The categorization is done under the following levels: *(i)* not supported at all (×), *(ii)* not recommended (👎), *(iii)* supported but not totally (☺), and *(iv)* well supported (✓). The resulted evaluation is shown in Table II. The assessed features per SGAM layer are:

1) Business: This layer covers the survey of problems detected in the power systems domain. Thus, business actors affected or responsible of the problem need to be stated as well as a clear definition of envisaged goals. Main concern of this survey is to understand the repercussion of the problem and to propose hypothesis for potential solutions.

The most suitable approaches are the SysML and SGAM Toolbox. SysML is more convenient regarding UML because of modeling constructs that express relation between requirements, those are available in the requirement diagram.

2) Function: In this domain the structure and behavior of control algorithm, constraints, parameters, and variables required for the performing of control strategies (i.e., setpoint, measurements, control, etc.) need to be specified. Moreover, the provision of templates for function models are contemplated at design phase. The use case example presented in Section II-A is implemented as follows: *(i)* the PI controller (UC1) is designed using function block representation, *(ii)* the frequency controller (UC2) is implemented by state machine, and *(iii)* the dynamic programming control (UC3) is designed under data-flow representation.

The evaluation in Table II shows that best approaches to specify and design the aforementioned implementation are EMSOnto and SysML. SysML assures a formal definition of constraint blocks and parametric diagrams. EMSOnto offers function models for frequency and voltage control, motivated from the standard IEC61850-90-7 [5]. On the other hand, IntelliGrid is not a good candidate since at the design phase, neither function data models nor definition of control logic behavior under state machine or function block representation are attended. SGAM Toolbox is a good tool at the specification phase but not totally recommended at the design phase since function block representation is not possible. Conversely, PSAL is encouraged at the design phase but not at the specification because a lack of constraints notation.

3) Information: At the specification phase definition of input, outputs, and parameters are evaluated. Those data are further detailed at the design stage. In there, provision of data models for components (electrical devices, monitoring system, etc.) and the contribution of a taxonomy for classification of information are surveyed.

The employ of the use case example shows that all the approaches except MATLAB/Simulink are appropriate at the specification phase. Indeed, MATLAB/Simulink is not intended for such task. EMSOnto and PSAL are highly recommended because they provide a common semantic for information. Furthermore, implementation of the use case example using those approaches allows the generation of a high amount of data adequately classified.

4) Communication: Communication protocols and ICT techniques have to be specified and configured. This is done by taking the information and data models that were identified in the information layer into account.

This layer is well supported only by PSAL. It allows the user to specify client/server and publish/subscriber communication patterns and also allows a low-level configuration of communication network parameters, such as description of Internet Protocol (IP) addresses, Virtual Area Local Network identifiers among other parameters.

5) Component: In the component layer the system is specified. In some cases it may also be necessary to design new parts of the system (e.g., adding new controllers or ICT equipment). In general, the needed components for the use case can be derived based on the actors that are involved in the use case, as well as any existing system components. Subsequently, the functionality defined in the function layer is assigned to a corresponding hardware.

Convenient mechanisms to represent ICT and power systems components (modeling of a battery, distributed energy generator, etc.) are given by the PSAL approach. MATLAB/Simulink enables also the modeling of electrical devices by the library powersim, but no mechanisms to match software artifacts into real hardware device are employed. Neither the description of physical interfaces.

6) Rapid prototyping: This feature seeks for the reduction of manual amount of work by automating the generation of software artifacts at the implementation phase. Hereby, function, information, and communication layers are analyzed.

The performance of SGAM Toolbox is not good enough. In cooperation with Enterprise Architect tool it provides code generation possibilities, but due to a not constrained syntax, the generated code need to be manually customized. In comparison, the code generated by PSAL could be used directly for tests. Moreover, PSAL also offers the generation of interface descriptions into configurations (e.g., IEC 61850 configurations). Function behavior with parameters and variables are generated under the standard programming language IEC 61499 [18].

EMSOnto in combination with MATLAB/Simulink is a good option for a proof-of-concept and implementation of control applications, offering an automatic generation of code into C. Regarding the communication layer, MATLAB/Simlunk together with Backman M1 control system, integrates fieldbus interfaces and input/output cards into the model enabling HIL measurements. On the other hand, EMSOnto does not cover the communication layer.

B. Discussion

The SGAM Toolbox and SysML have shown good performance at the specification phase. SysML provides a large set of behavior, structural and requirement diagrams, thus it is high recommended, however it is not power system oriented. An advantage of the SGAM Toolbox is the structure of control applications into domain, zones and interoperability layers (communication, information, etc.), hereby the handling of complexity in smart grids is reached. On the other hand, the lack of syntax formalization leads to a poor support for model-driven engineering process. As a result, mechanisms to exploit SGAM and SysML at the design stage are

TABLE II: Comparison of Approaches and Tools.

Phase	**Specification**					**Design**				**Implement.**
Approach	**business**	**function**	**inform.**	**comm.**	**comp.**	**function**	**inform.**	**comm.**	**comp.**	**rapid prot.**
UML	😐	👎	✓	✓	×	😐	×	×	×	×
SysML	✓	✓	✓	✓	✓	✓	👎	×	×	×
IntelliGrid	😐	😐	✓	×	×	👎	×	×	×	×
SGAM TB	✓	✓	✓	✓	✓	😐	×	👎	👎	👎
EMSOnto	👎	✓	✓	×	×	✓	✓	×	×	😐
PSAL	👎	😐	✓	✓	✓	✓	✓	✓	✓	✓
MATLAB	×	×	×	×	✓	✓	👎	×	👎	✓

Legend: not supported at all (×), not recommended (👎), supported but not totally (😐), and well supported (✓)

missing. EMSOnto enables the handling of a high amount of information because it is collected by spreadsheet (i.e., excel templates). However, this process is not the most user friendly since connections between components need to be textually indicated. Same drawback is presented at the PSAL approach. Regarding IntelliGrid, it shows many disadvantages trough all the SGAM layers. PSAL, MATLAB/Simulink and EMSOnto need to be complemented with a good approach at the specification phase. In this study, time employed to carry out the specification and design phase is not considered, this would modify the assessment of the divers approaches. Deriving to a wrong selection of methodologies when the time is a crucial constraint.

V. CONCLUSIONS

This study evaluates the performance of specification and design approaches for power system applications together with an accompanying comparison. The analysis shows that none of the studied approaches fully cover the specification and design phase. Thus an assemble and cooperation of them is required, for instance the SGAM Toolbox and PSAL are enough to make a good specification and design of a power system application. On the other hand, detection of inconsistencies is neglected, a feature that is tackled by EMSOnto. For this reason, it is encouraged to evaluate which support is required in order to tailor and select the right approaches.

Summarizing, there is still a significant improvement potential for the further development of specification and design approaches as well as the integration of them.

ACKNOWLEDGMENT

This work is supported by the Austrian Ministry for Transport, Innovation and Technology (bmvit) and the Austrian Research Promotion Agency (FFG) under the “ICT of the Future Programme” in the MESSE project (FFG No 861265).

REFERENCES

[1] EERA Joint Programme on Smart Grids, “Sub-Programme 4: Electrical Energy Technologies,” 2014.

[2] R. Hollinger, L. M. Diazgranados, F. Braam, T. Erge *et al.*, “Distributed solar battery systems providing primary control reserve,” *IET Renewable Power Generation*, vol. 10, no. 1, pp. 63–70, 2016.

[3] M. Kazemi, H. Zareipour, N. Amjady, W. D. Rosehart, and M. Ehsan, “Operation scheduling of battery storage systems in joint energy and ancillary services markets,” *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1726–1735, 2017.

[4] F. Pröstl Andrén, T. Strasser, and W. Kastner, “Applying the SGAM Methodology for Rapid Prototyping of Smart Grid Applications,” in *42nd Annual Conf. of the IEEE Ind. Electronics Society (IECON)*, 2016.

[5] Int. Electro. Comm. (IEC), *Draft IEC 61850-90-7 TR Communication networks and systems for power utility automation*, Std., 2011.

[6] B. Perera, P. Ciufo, and S. Perera, “Advanced point of common coupling voltage controllers for grid-connected solar photovoltaic (PV) systems,” *Renewable Energy*, vol. 86, pp. 1037–1044, 2016.

[7] Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, “Optimal power flow management for grid connected pv systems with batteries,” *IEEE Transactions on Sustainable Energy*, vol. 2, no. 3, pp. 309–320, 2011.

[8] M. Uslar and K. Heussen, “Towards modeling future energy infrastructures—the electra system engineering approach,” in *2016 IEEE PES Innovative Smart Grid Techn. Conf. Europe (ISGT-Europe)*, 2016.

[9] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified modeling language reference manual, the*. Pearson Higher Education, 2004.

[10] T. Weilkiens, *Systems engineering with SysML/UML: modeling, analysis, design*. Elsevier, 2011.

[11] J. Hughes, “IntelliGrid Architecture Concepts and IEC61850,” in *2005/2006 IEEE/PES Transm. and Distr. Conf. and Exh.*, 2006.

[12] Int. Electro. Comm. (IEC), *IEC 62559, PAS: Intelligrid Methodology for developing requirements for Energy Systems*, Std., 2008.

[13] S. Rohjans, C. Dnekas, and M. Uslar, “Requirements for Smart Grid ICT-architectures,” in *2012 3rd IEEE PES Innovative Smart Grid Techn. Conf. Europe (ISGT-Europe) (ISGT Europe)*, 2012, pp. 1–8.

[14] CEN-CENELEC-ETSI, “Reference Architecture for the Smart Grid,” Tech. Rep., 2012.

[15] ——, “SG-CG/M490/F Overview of SG-CG Methodologies,” Tech. Rep., 2014.

[16] J. Trefke, S. Rohjans, M. Uslar, S. Lehnhoff *et al.*, “Smart Grid Architecture Model use case management in a large European Smart Grid project,” in *2013 4th IEEE/PES Innovative Smart Grid Techn. Europe (ISGT EUROPE)*, 2013.

[17] C. Dänekas, C. Neureiter, S. Rohjans, M. Uslar, and D. Engel, “Towards a model-driven-architecture process for smart grid projects,” in *Digital enterprise design & management*. Springer, 2014, pp. 47–58.

[18] F. Pröstl Andrén, T. Strasser, and W. Kastner, “Engineering Smart Grids: Applying Model-Driven Development from Use Case Design to Deployment,” *Energies*, vol. 10, no. 3, p. 374, 2017.

[19] C. Zanabria, A. Tayyebi, F. Pröstl Andrén, J. Kathan, and T. Strasser, “Engineering support for handling controller conflicts in energy storage systems applications,” *Energies*, vol. 10, no. 10, p. 1595, 2017.

[20] P. Hitzler, M. Krotzsch, and S. Rudolph, *Foundations of semantic web technologies*. CRC press, 2009.

[21] World Wide Web Consortium, “Sparql 1.1 query language,” 2008, https://www.w3.org/TR/sparql11-query/.

[22] C. Zanabria, F. Pröstl Andren, J. Kathan, and T. Strasser, “Approach for handling controller conflicts within multi-functional energy storage systems,” *CIRED – Open Access Proceedings Journal*, vol. Volume 2017 Issue 1, no. 0803, pp. 1575–1578, 2017.