Multi-level configuration in smart governance systems

Salvador Muñoz-Hermoso[1,3,*], David Benavides[2,3] and Francisco J. Domínguez-Mayo[2,3]

[1]Provincial Informatics Company (INPRO), Seville Provincial Council, 32 Menéndez y Pelayo St, Seville, 41071, Spain
[2]Dept. of Computer Languages and Systems, University of Seville, Reina Mercedes St, 41012, Seville, Spain
[3]DiversoLab Computer Sience Laboratory, University of Seville, Reina Mercedes St, 41012, Seville, Spain

Abstract
Smart governance systems have different needs depending on the type of organization, which, together with their inherent complexity, make them difficult to configure. However, we have not found solutions that facilitate the configuration of these systems of great interest in the public sector. We propose a configurability solution compounds of a software framework-based multi-level configuration architecture, and a feature model (FM) to represent the variability in a compact way through the configuration of the different levels of the same software. Thus, the FM we present allows for obtaining a line of customized services for different organizations. On a first level, the variability of the typical collaboration processes is managed, on a second level the different collaboration models are handled, and on a third and fourth level the general smart governance system is configured, and the one adapted to the specific needs of the organization. In this way, while facilitating configuration, a high degree of accuracy is achieved regarding the specific and varying needs of the different stakeholders.

Keywords
Software Configurability, Feature Models, Software Reutilization, Smart Governance, E-Collaboration

1. Introduction

The configurability and variability management of information systems is important, in that it enables software products and services to be adapted to the needs of the organization and its stakeholders [1].

In relation to e-collaboration technologies and systems, they have an inherent complexity that is increased by the fact that their requirements and needs vary according to the application domain, and the type of organization using them [2].

In the smart governance domain (related to the broader field of e-government), the aim is to achieve quality public services and smart management (of territories and societies) through a collaborative government open to citizen, business and professional collectives, maximizing positive results with intensive use of Information Communication and Technology (ICT) [3]. Collaboration is therefore essential and particularly complex due to the different (sometimes conflicting) interests of the actors involved [4].

In the e-government exists a great variety of needs, given that governance processes and public services in a small municipality are not the same as those in a large city, or governance in a regional or state administration [5, 6]; thus it is essential to address the different needs of citizens and governments, in terms of participation and collaboration in public policies and services. Furthermore, smart governance involves multiple stakeholders that enhances this complexity and the variety of unexpected and changing requirements, as this is still a recent field with respect to its implementation. Hence, the development of different tailor-made systems substantially increases development and configuration costs.

It is therefore desirable, to improve the reusability and address this great variety by managing the variability (which can be changeable) of a unique software or, at least, by reducing its variants and modifications. In such a complex environment, so is its configuration, thus it is also convenient to enhance the configurability of these systems. Nevertheless, we have not found any solutions in the literature review that addresses this variability facilitating moreover its configurability.

Software Product Line Engineering (SPLE) and frameworks software favor variability and reusability [6, 7, 8], and consequently, the adaptation of the software developed to the specific needs of the organization.

In this context, we propose a service-oriented configurability approach based on a multi-level software framework-based configuration architecture to provide a software product line (SPL); and a feature model (FM) in the domain of smart governance. The FM allows representing all possible configurations in a compact way [8, 9], through the configuration of the different levels of the software. So, the configuration architecture and the FM model complement each other, to facilitate the implementation of the variability of the services provided

ConfWS'23: 25th International Workshop on Configuration, Sep 6–7, 2023, Málaga, Spain
*Corresponding author.
samu31@dipusevilla.es (S. Muñoz-Hermoso); benavides@us.es (D. Benavides); fjdominguez@us.es (F. J. Domínguez-Mayo)
http://www.lsi.us.es/~dbc/ (D. Benavides); https://www.us.es/trabaja-en-la-us/directorio/francisco-jose-dominguez-mayo (F. J. Domínguez-Mayo)
0000-0002-8130-7869 (S. Muñoz-Hermoso); 0000-0002-8449-3273 (D. Benavides); 0000-0003-3502-8858 (F. J. Domínguez-Mayo)

by these systems.

This paper is structured as follows: Section 2 explores the multi-level configuration architecture as part of the solution. In Section 3, we introduce the proposed FM to describe the configuration aspects of the smart governance system related to the presented architecture. In Section 4, a review of related works is provided to highlight the contribution of the proposed solution to previous research on SPL, e-government and e-collaboration topics. Finally, Section 5 concludes the paper, summarizing our main findings and observations, and proposing some future research works.

2. Multi-level configuration architecture

In the flat configurability approach, if a large variability of complex systems needs to be covered, multiple features and parameters must be considered, and it is difficult to manage a software product line and product customization [10].

In this section, we show a multi-level architecture proposal to support a multi-level services configuration. This approach favors configurability and reduces the complexity of managing the associated variability [10, 11, 12]. Furthermore, it is service-oriented to facilitate the development of Software-as-a-Service (SaaS) e-governance systems.

Thus, the multi-level perspective enables division and hierarchizes the configuration into different levels related to modules or system parts, requirements groups, or service models, facilitating the configuration, customization, and reuse of the services.

2.1. Framework-based architecture

In order to achieve domain adaptability, we opted for an architecture based on domain software frameworks; they favor the reuse and implementation of solutions, in particular in the public sector [7, 6, 5]. Thus, this architecture also facilitates the implementation of the configurability management of the proposed system.

To this end, we consider a first framework (*E-Collaboration Framework*) with common characteristics and functionalities for the collaboration of a group of stakeholders on certain organizational assets (documents, projects, policies, etc.). And a second framework, specialized from the first one, focused on the particular needs and services of the smart governance domain (*Smart Governance Framework*), such as citizen consultations, drafting of regulations and policies, or participation in smart city projects.

In this way, the configurability of services is addressed through the various levels of the software frameworks and the smart governance framework-based system; in order to meet the specific needs of each organization.

2.2. Description

In Figure 1, in TOGAF-ArchiMate notation[1], we show a layered view with the most relevant artifacts of the architecture involved in the configuration, in which the different levels (gray) containing the different business objects that store the configuration are observed. This high-level modeling is suitable for representing in a visual and clear way both behavioral and static storage artifacts. The shown architecture supports the proposed FM described in the next section.

The business layer (yellow) is service-oriented and is structured in two blocks; the one on the left supports the configuration of the general e-collaboration software framework, and the block on the right offers services to manage the configuration of the adapted smart governance framework, as well as the system that is implemented around it.

2.2.1. Purpose of the levels

Since the smart governance solution is based on general domain software frameworks, it is necessary to consider a first level (0) to establish the general configuration of this reusable software in order to adapt it to the needs of the specific smart governance solution to be developed.

On the other hand, smart governance is articulated through participatory processes that are usually typified, in some cases, on the basis of citizen participation regulations. Therefore, it is appropriate to be able to define and characterize these process models through a next level (1) of configuration that will allow their adaptation (both specific and general aspects) to the needs of the organization. E.g. a certain process can be modeled for the collaborative drafting of regulations, or another for participatory budgeting. Given that there are different types of public institutions and different smart governance policies and citizen participation regulations; it is therefore desirable, to be able to model and configure them at a new configuration level (2), so that they can be reused and adapted for each institution. In addition, these models usually involve certain types of collaborative processes, hence their relationship with level 1. E.g. a model could be established for small municipalities, or for smart cities that have collaborative needs in urban projects.

[1]TOGAF is an OpenGroup IEEE standard framework for developing enterprise architectures (https://www.opengroup.org/togaf). ArchiMate is a (graphical and semantic) modeling language for OpenGroup's high-mid level enterprise architecture under the TOGAF standard (https://www.opengroup.org/archimate-forum/archimate-overview).

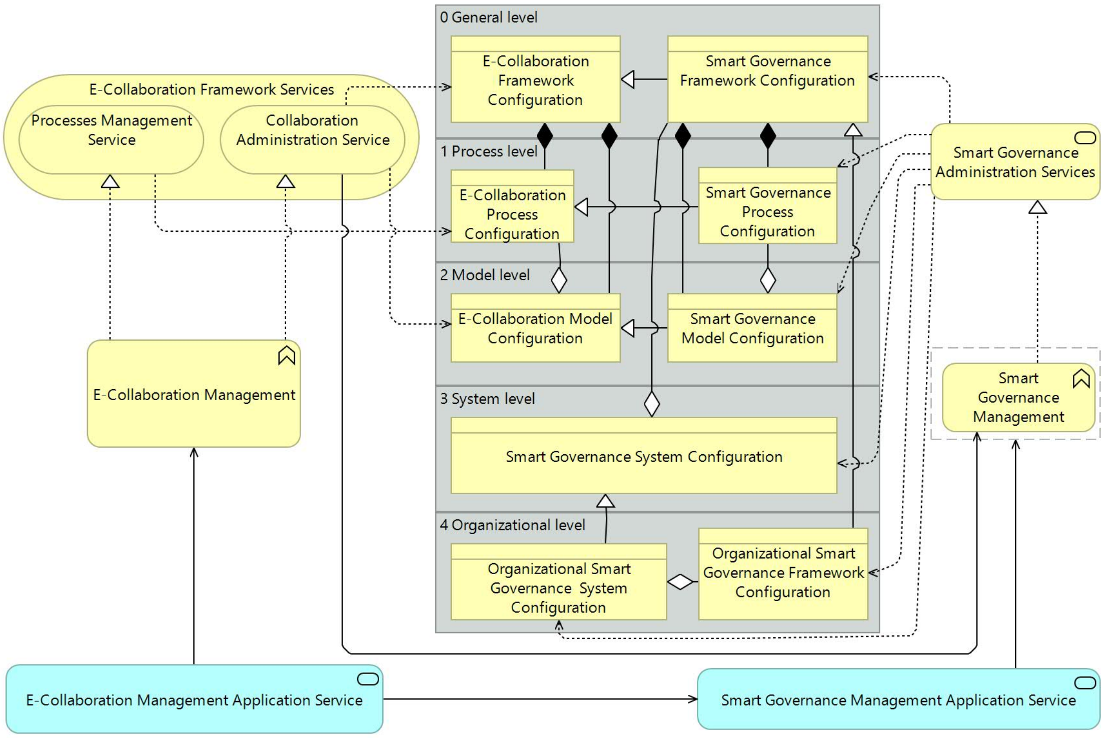

Figure 1: Architecture of the multi-level configuration of the frameworks and the system

These two levels favor dynamic variability, since enable a system in production that requires new needs, to create new types of collaborative processes or new governance models, that can be easily incorporated into the smart governance services of a specific organization, or a set of organizations using the same governance model.

While the above levels characterize the smart governance models with their associated processes in software frameworks, we need a level (3) of configuration relative to the specific system to be developed with the necessary particularities for use by specific organizations that adhere to the unique reusable system. E.g. a local administration could develop a system for use by various city councils. Furthermore, these levels make configurability more efficient, since several organizations can be customized at once, acting on levels 1 and 2 of the configuration, or on level 3 for features that globally affect the behavior of the entire system.

Based on the above, it follows that an additional level of configuration (4) is desirable, so that each organization can tailor the system to its requirements, both in terms of general and specific characteristics of the governance model and its processes. E.g. one municipality may want its participatory processes to be binding, while another may only want them to be consultative. Or regarding specific characteristics such as the duration of certain participatory processes, participation requirements and restrictions, etc. In this way, it contributes to the precision and facility of the customization, since the configurations are inherited from top to bottom (through the levels), it is possible to customize a particular entity, only by modifying the differential features at level 4.

2.2.2. Level 0

Starting at configuration level 0 (general) and the block on the left, the *Collaboration Administration Service* facilitates the adaptation of the e-collaboration software framework through the setting of the general configuration data (*E-Collaboration Framework Configuration*), which helps to customize it in the application area. In addition, in the right-hand block, customization is enhanced by means of the *Smart Governance Administration Services*, for configuring specific features of the particular software framework in the smart governance domain (object *Smart Governance Framework Configuration*).

2.2.3. Level 1

At level 1 (processes), the *Processes Management Service* in the left-hand block allows the different types of collaborative processes to be configured, storing the corresponding information in the object *E-Collaboration Process Configuration*, which is part of the framework configuration. In addition, in the right block, the *Smart Governance Administration Services* provides a finer adjustment of the processes, specifying a specific typology in the smart governance field, the configuration of which is stored in the object *Smart Governance Process Configuration.* In short, this first level manages the variability of collaborative processes, both in general and domain-specific.

2.2.4. Level 2

In order to achieve greater variability, similarly at level 2 (models), the same services of the respective frameworks facilitate the creation of a product line related to the e-collaboration and smart governance models (or organization types) that are established. The corresponding configurations are stored respectively in the objects *E-Collaboration Model Configuration* and *Smart Governance Model Configuration.*

2.2.5. Level 3

Level 3 (system), allows obtaining a specific smart governance type system, based on the framework with the desired features, in relation to the collaboration models to be supported, the types of processes, and other characteristics that were already specified in the previous configuration levels. The configuration of these features is transferred to the system and other domain-specific features are added and stored in the *Smart Governance System Configuration* object. Thus, at this level, we select a product type and a customized SPL product that provide a set of personalized smart governance services.

2.2.6. Level 4

The last level 4 (organizational), enables a configuration adapted to the specific needs of each organization adhering to the system. To this end, through the *Smart Governance Administration Services*, the system and framework configurations are inherited to be customized respectively in the objects *organizational Smart Governance System Configuration* and *Organizational Smart Governance Framework Configuration*, which will make up the complete and specific configuration of the organization's system.

2.3. Services realization

If we go down the business layer, in the left block, we can see how the *E-Collaboration Management* functionality (functional part of the e-collaboration framework) is the one that performs the aforementioned services. Similarly, on the right-hand side, the *Smart Governance Management* is the functionality that realizes the administration and configuration services of the system for smart governance. In terms of reusability, it relies on the general administration service of the e-collaboration framework.

Finally, in the information systems layer (light blue), the application services that support the previous business functionalities are included; *E-Collaboration Management Application Service* for the general configuration of e-collaboration, and *Smart Governance Management Application Service* for the specific configuration of smart governance services, which also relies on the previous one to enhance reuse.

3. Feature model

In this section, to complete our configurability solution, we propose a FM (Figure 2) to represent the variability of smart governance solutions with respect to the defined configuration levels, which can be implemented based on the configuration architecture of the previous section.

The FM considers the five configuration levels to obtain a family of customized products down to the specific system level of the organization.

First, we outline the key features common to the entire line of smart governance systems and, second, those that may vary for each specific system.

3.1. Common features

The system offers, on the one hand, common services for e-collaboration in any type of organization and application domain (*E-Collaboration Common Services*), and on the other hand, in the context of the study, domain-specific services (*Smart Governance Services*). These two groups are mandatory in any configuration, because these general services are necessary to accomplish any process related to smart governance. Nevertheless, these must be customized to adapt to the organization's needs, through their features and subgroups, some of them are optional.

It is also mandatory to establish the e-collaboration model to be used with its features, as well as the specific characteristics of typical collaborative processes to be used in the model. The variable cardinality of *E-Collaboration Model* and *E-Collaboration Process*, increases configurability, by making it possible to define a set of models, and for each of them, different types of processes, covering levels 1 and 2 of configuration that we discussed in the previous section; that is, different services depending on the models and types of collaborative processes that they implement. This multi-level approach to FM, in which these feature trees can be considered as

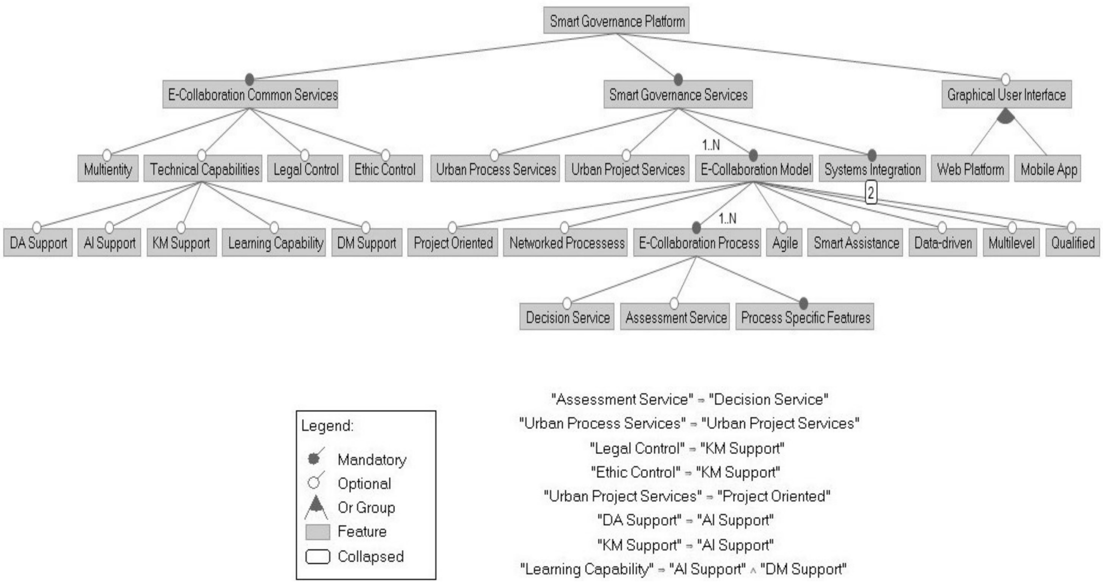

Figure 2: Feature model for a smart governance system

separate but linked FMs, allows the high complexity of these highly variable systems to be better managed [12].

In the particular application area, because of the need for information from the environment, the system cannot operate in isolation (being part of a software ecosystem), and therefore the group *System Integration* is mandatory. Within this group (in Figure 2 is collapsed), the feature *Organizational Systems & Services* is mandatory, as the system must interoperate with other existing information systems in the organization (e.g. basic citizenship data and identification services). However, we do not consider a requirement the integration with other external services (*External Global Services* feature) such as social networks or messaging services, although it would provide more information to the system.

3.2. Variable features

In relation to e-collaboration common services, a multientity system (*Multientity*) can be chosen, to be used independently in different entities or organizations and customized in each one of them. The features *Legal Control* and *Ethic Control*, will activate respectively the control mechanisms to favor the regulatory compliance of the domain and the organization, and its ethical values. To this end, the system must be able to model and store them like rules in a knowledge database. So these characteristics require, as we can see in Figure 2, the activation of some technical features (*Technical Capabilities*), specifically those that offer support to knowledge management (*KM Support*), to be able to handle the rules and their inference.

In the field of smart governance, apart from the specific services offered, governance can be extended to urban processes and projects by activating the *Urban Process Services* and *Urban Project Services*, as the latter is required for collaboration in the former[2].

The group *Graphical User Interface* (GUI) is not mandatory, as an existing external interface layer can be chosen. If the system GUI is selected, a choice can be made between a web interface, a mobile interface, or both; for increased interoperability and accessibility from any device.

These features act at levels 0, 3, and 4 for entity-level customization of both the smart governance system and the software frameworks on which it is based.

In relation to level 1, we highlight within the group *E-Collaboration Process* two features that extend e-collaboration, thus the *Decision Service* will enable decision-making (DM) by the collaborators, which will optionally enable an evaluation of the same with the activation of the feature *Assessment Service.* For effective individual and collective decision-making, it is also necessary to activate technical capabilities such as the *DM Support* feature, whose group *Technical Capabilities*

[2] An urban process usually develops projects in the implementation phase from ideas to proposed solutions [4].

we will see at the end of the section. In smart governance, these services will enable public policy evaluation citizenship processes.

As for the optional features of the collaboration models (*E-Collaboration Model*), there are some that, with their activation, enhance the capabilities (and also the complexity) of the model: *Project-oriented* enable e-collaboration at project and project phase level (required if *Urban Project Services* are activated). *Networked-Processes* allows the creation of more complex collaborative processes based on simpler ones, or to relate processes to each other, forming a network to interoperate between them.

The *Multilevel* feature extends collaboration to the different decision-making levels of the organization (strategic, tactical, and operational), which combined with the previous feature and the *Multientity* feature, supports more complex and transversal processes in different organizations to solve common problems.

The *Agile* feature introduces an agile approach to collaboration through feedback between the different phases of e-collaboration, and even with other networked collaborative processes. This feature, together with the previous one (*Multilevel*), in the field of smart governance, facilitates dynamic collaboration throughout the public policy cycle.

The *Data-driven* feature favors collaboration and decision-making based on evidence or objective data, requiring the activation of the technical capacity *DA Support* that we will see below. Enhanced by the influence of the expertise and qualification of the collaborators (*Qualified* feature), in the decision-making process, it allows knowledge to be promoted in the final results of the decisions. Both features contribute, in smart governance, to a citizen-centric government.

As a final representative feature of the model, smart assistance (*Smart Assistance*) is envisaged, in order to support informed and effective decision-making. This requires the activation of applied AI techniques (*AI Support*) or decision-making techniques (*DM Support*).

Finally, the group of optional features *Technical Capabilities* (some of which are required by other models and processes), enhances addressing complex collaboration and decision-making problems by including technical capabilities [4] for data analysis (*DA Support*), knowledge management from relevant data (*KM Support*), individual and group knowledge-based decision-making (*DM Support*), and autonomous learning (enabled by *Learning Capability* feature), which would enable the system to autonomously configure itself to improve outcomes. It can be seen that AI techniques (*AI Support*) are required to support the above capabilities. The configuration of these features applies to levels 0, 3, and 4, enabling customization of technical capabilities at the system and organization level.

The use of variable features and the multi-level approach will not only facilitate the configuration and the customization, but also the standardization of governance processes. For example, in a context where several city councils need governance based on participatory consultations and surveys, a *E-Collaboration Model* could be defined with the desired features and with both types of processes (*E-Collaboration Process*) with their specific characteristics (*Process Specific Features*) already configured.

In this way, this model would simply have to be activated in each organization; it would also be possible to customize some of the features of the process or model in a specific entity. In addition, new process-specific features can be added to address dynamic variability more accurately, due to stakeholders' changing needs and tasks, as these are mostly related to participatory processes.

It therefore, has clear advantages over the traditional flat approach, as it would involve repeating the same configuration work over and over again for each feature and entity, and the reuse of the configuration would be more consistent, or if some standardization of processes is desired, which is desirable in public administration.

4. Related works

In the literature review, we have not found any articles that specifically address the problem of configurability in smart governance. Nor have we found proposals for multi-level configurations in the more general domain of e-government.

However, we have identified some e-government studies that although they do not focus on improving configurability, do address how to facilitate the development and adaptation of these e-government systems to different needs through Software Product Line (SPL) and domain frameworks [13, 6, 14]. And others like in [15] propose using Feature Models (FM). This work is interesting, because of its broad vision as ours, since proposes a general model for e-governance systems; furthermore, they establish a division by front-office or back-office software, and another by applications typology: Government to Government (G2G), Government to Business (G2B), or Government to Citizen (G2C).

In [6] further distinguish products for central or local governments, which is appropriate as the latter offers public services related to city government; quite different from those offered by the state. Regarding our work, adaptation to a local, regional, or central government, could be accomplished through different models defined in level 2 (models) of configuration.

In [5] they also propose a framework approach but do not address configurability as a specific problem, but focus more on facilitating the development of electronic public services. SPL is also applied in some particular use cases such as the one proposed in [16] for content management systems (CMS).

In short, as in [13] is mentioned, there are few studies that address variability and SPL in the e-government domain, so this is an area that needs to be explored further.

5. Conclusion and future works

The Feature-Model (FM) and the architecture that supports it, proposed in this work, facilitate multi-level service-oriented configurability (at the level of the general e-collaboration software framework and its processes, the particularized framework in the domain, and the smart governance system), product line configurability (each model can be considered a product for a particular type of smart governance or institution), and multi-entity configurability (supporting different configurations for each organization). Therefore, from a single software system, through reuse and configuration, it is possible to obtain a significant dynamic variability of services for e-collaboration and in particular for smart governance. Moreover, the *Learning Capability* and *AI Support* features will enable an autonomous configuration to evolve the system towards configurations more adapted to the organizations.

Concerning other related proposals, ours focuses on the specific problem of configurability from a general perspective by providing several complementary methods and techniques integrated into the solution: multi-level configuration architecture, domain software frameworks, SPL and FM; as well as TOGAF-Archimate as formal modeling framework.

The preliminary results show that the configurability architecture proposed in the present study contributes to the general area of e-collaboration, and in particular of smart governance, to facilitate the characterization and configuration of these systems, also favoring their reusability, and adaptability with respect to the particular and varying needs of the different stakeholders and organizations.

Since we have not carried out a systematic review of other possible configurability solutions in other areas, a follow-up to this work could be to conduct such a study to establish possible relations and synergies. Another future work could be envisaged to further specify the configuration architecture and the FM, aimed at developing software prototypes, either in general, in the specific domain of smart governance, or another application area.

The development of a prototype for a given use case (e.g. a governance system for a specific city council) would help to validate our contribution. To this end, quantitative performance metrics (e.g. related to the time spent on configuration processes, its complexity, or accuracy) could also be studied to empirically assess its efficiency and effectiveness compared to other approaches and proposals to manage configurability.

Furthermore, the FM could also be specified at a higher level of detail by developing feature sets, e.g. the *Process Specific Features*, or exploring new features that may be of interest. Tools to support the proposal would also be of interest, e.g. to validate the consistency of the model in relation to the features that are selected, as well as to generate the corresponding software services from them.

Acknowledgments

This work was supported by the *VII Own Plan* research aid from the University of Seville, FEDER/Ministry of Science and Innovation/Junta de Andalucía/State Research Agency with the following grants: *Data-pl*(PID2022-138486OB-I00) , TASOVA PLUS research network (RED2022-134337-T) and *METAMORFOSIS* (FEDER_US-1381375)

References

[1] J. Bosch, Software variability management, in: Proceedings - International Conference on Software Engineering, volume 26, Elsevier B.V, 2004, pp. 720–721. doi:10.1016/j.scico.2004.06.001.

[2] B. E. Munkvold, I. Zigurs, Integration of e-collaboration technologies: Research opportunities and challenges, International Journal of e-Collaboration (IJeC) 1 (2005) 1–24.

[3] C. Jiménez, Una aproximación al concepto de Gobernanza Inteligente., Perspectiva (2013) 44–48.

[4] G. Tran Thi Hoang, L. Dupont, M. Camargo, Application of Decision-Making Methods in Smart City Projects: A Systematic Literature Review, Smart Cities 2 (2019) 433–452. URL: https://www.mdpi.com/2624-6511/2/3/27. doi:10.3390/smartcities2030027.

[5] A. Ojo, T. Janowski, E. Estevez, A Composite Domain Framework for Developing Electronic Public Services. (2007). URL: https://mural.maynoothuniversity.ie/15885/.

[6] A. N. Fajar, I. M. Shofi, Reduced software complexity for E-Government applications with ZEF framework, Telkomnika (Telecommunication Computing Electronics and Control) 15 (2017) 415–420. doi:10.12928/TELKOMNIKA.v15i1.3195.

[7] I. Achour, L. Labed, R. Helali, H. B. Ghazela, A Service Oriented Product Line Architecture for E-Government, The 2011 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, EEE 2011 (2011) 186 – 192.

[8] K. Pohl, G. Böckle, F. Van Der Linden, Software product line engineering: Foundations, principles, and techniques, Springer Berlin / Heidelberg, Berlin, Heidelberg, 2005. doi:10.1007/3-540-28901-1.

[9] D. Benavides, S. Segura, A. Ruiz-Cortés, Automated analysis of feature models 20 years later: A literature review, in: Actas de las 16th Jornadas de Ingenieria del Software y Bases de Datos, JISBD 2011, 2011, pp. 951–952.

[10] T. Clark, U. Frank, I. Reinhartz-Berger, A. Sturm, A multi-level approach for supporting configurations: A new perspective on software product line engineering (2017).

[11] K. Czarnecki, S. Helsen, U. Eisenecker, Staged configuration through specialization and multilevel configuration of feature models, Software process: improvement and practice 10 (2005) 143–169.

[12] M. O. Reiser, M. Weber, Multi-level feature trees: A pragmatic approach to managing highly complex product families, in: Requirements Engineering, volume 12, Springer Nature, NEW YORK, 2007, pp. 57–75. doi:10.1007/s00766-007-0046-0.

[13] G. Cledou, L. S. Barbosa, Modeling families of public licensing services: A case study, in: Proceedings - 2017 IEEE/ACM 5th International FME Workshop on Formal Methods in Software Engineering, FormaliSE 2017, IEEE, 2017, pp. 37–43. doi:10.1109/FormaliSE.2017.8.

[14] I. Achour, L. Labed, R. Helali, H. B. Ghazela, A Service Oriented Product Line Architecture for E-Government, in: The 2011 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, EEE 2011, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Athens, 2011, pp. 186 – 192.

[15] N. Debnath, L. Felice, G. Montejano, D. Riesco, A feature model of E-government systems integrated with formal specifications, Proceedings - International Conference on Information Technology: New Generations, ITNG 2008 (2008) 27–31. doi:10.1109/ITNG.2008.104.

[16] V. M. A. D. Lima, R. M. Marcacini, M. H. P. Lima, M. I. Cagnin, M. A. S. Turine, A generation environment for front-end layer in e-government content management systems, in: Proceedings - 9th Latin American Web Congress, LA-WEB 2014, IEEE, 2014, pp. 119–123. doi:10.1109/LAWeb.2014.20.