



Decision Support Ecosystems: Definition and Platform Architecture

Jonas Kirchhoff^(✉), Christoph Weskamp, and Gregor Engels

Department of Computer Science, Paderborn University, Paderborn, Germany
{jonas.kirchhoff, christoph.weskamp, engels}@upb.de

Abstract. Decision support systems are crucial in helping decision makers to quickly identify optimal business decisions in increasingly volatile and complex business environments. However, the ideal DSS for one decision maker may not optimally address the requirements for decision support of another decision maker. This is due to differences between decision makers in business goals, regulatory restrictions or availability of resources such as data. By using a suboptimal DSS, decision makers risk implementing suboptimal decision recommendations which endanger the success of their business. This presents DSS developers with the challenge to implement a customizable DSS which can be tailored to the individual requirements for decision support of a single decision maker. In order to address this challenge, we suggest a *decision support ecosystem* in which DSS developers, decision makers and other domain experts collaborate using a shared platform to provide and combine reusable decision support services into a tailored DSS. The contribution of our paper is twofold: First, we define the concept of a decision support ecosystem with respect to existing digital business ecosystems and discuss expected benefits and challenges. Second, we present a reference architecture for a shared platform supporting the realization of a decision support ecosystem. We demonstrate our contributions in the example application domain of regional energy distribution network planning.

Keywords: Decision support ecosystem · Ecosystem platform · DSS generator · Multi-enterprise DSS · Collaborative DSS

1 Introduction

Business environments exhibit an increasing volatility, uncertainty, complexity and ambiguity (VUCA). Decision makers must therefore consider frequent, unpredictable change in many influencing factors with unknown cause-effect relationships when making a decision [1, 13]. This circumstance creates a demand for interactive computer-based decision support systems (DSS) to help decision makers quickly identify optimal decisions [16, 20, 23].

Partially supported by the European Regional Development Fund (ERDF) through grant EFRE-0801186 and the North Rhine Westphalian Ministry of Economic Affairs, Innovation, Digitalisation and Energy (MWIDE) through grant 005-2011-0022.

© Springer Nature Switzerland AG 2022

A. P. Cabral Seixas Costa et al. (Eds.): ICDSSST 2022, LNBIP 447, pp. 97–110, 2022.

https://doi.org/10.1007/978-3-031-06530-9_8

In the context of an interdisciplinary research project for decision support in the domain of regional energy distribution network planning with industry partners [10], we observed that the requirements for decision support vary between individual decision makers. For instance, decision makers can only leverage a cross-sectoral planning approach when they actually manage distribution networks for multiple energy sectors. Decision makers furthermore may have different targets with respect to metrics such as network reliability and network reinforcements costs. Moreover, decision makers have different access to resources such as forecast data or time available to identify an optimal decision. Similar observations can be made in the domain of supply chain management [23, 25] and business model development [8]. A misalignment in the requirements for decision support derived from the situational context of an individual decision maker and the decision support provided by a DSS is expected to result in suboptimal decision recommendations. The implementation of those recommendations can endanger the competitiveness of the associated business or even negatively impact society as a whole in case the business manages critical infrastructure. Consequently, DSS developers need to implement each DSS so that it can be tailored to the individual requirements of decision makers derived from business goals, constraints and availability of resources. However, such customization is often lacking in existing “off-the-shelf” DSS and retrospective extensions are usually a cost- and time-intensive undertake. This circumstance raises the research question: *How to provide decision makers with decision support systems tailored to their individual requirements for decision support in a timely manner?*

Existing state of the art in the domain of decision support and software engineering is only partially suitable for the adhoc creation of tailored DSS (cf. Sect. 2). In this paper, we therefore propose the concept of a *decision support ecosystem* in which decision makers, DSS developers and domain experts collaborate using a shared platform to document individual requirements for decision support functionality and to provide reusable software and data services that can be combined to implement such functionality without extensive software development knowledge. By assembling reusable services without software development knowledge, we expect decision makers to be able to quickly create tailored DSS themselves. Our contribution towards such decision support ecosystem is twofold: First, we provide a definition for decision support ecosystems and discuss their expected benefits and challenges with respect to existing digital business ecosystems (cf. Sect. 3). Second, we describe a reference architecture for the shared platform of a decision support ecosystem (cf. Sect. 4) and explain future research that is required to implement such a platform and other aspects of decision support ecosystems (cf. Sect. 5).

2 State of the Art and Related Work

A **DSS generator** is an “environment for developing an application-specific DSS” by providing “tools that make it easier and faster to develop models, data, and user interfaces that are customized to the application’s requirements” [2].

DSS generators historically require knowledge in mathematical modeling [2] which cannot be expected from decision makers and domain experts [20], thereby leaving any tailoring up to the DSS developer. A search on Clarivate’s *Web of Science* for publications implementing DSS generators throughout the last ten years reveals a narrow focus on maps [11], spreadsheets [20] or software developers [12]. **Adaptive DSS** adapt support to “the high-level cognitive needs of the users, task characteristics, and decision contexts” [7]. While this adaptation can consider a decision maker’s situational requirements for decision support, the run-time adaptation only works to the extent that was considered up-front by the DSS developer during design-time. **Multi-Enterprise Collaborative DSS** aim to provide decision makers with “decision making components (e.g., data, models, solvers and data and process visualizations)” across multiple enterprises [23]. However, existing approaches either focus on a single application domain (e.g., [26]), the selection of data sources or software functionality without recombination (e.g., [23, 24]) or uniting decision makers from multiple enterprises without any customization of decision support [6]. **Service-oriented DSS** [5] provide a conceptual framework for DSS development that targets software engineers, not decision makers or domain experts with limited to no software development knowledge (although these stakeholders may be partially supported using automated service composition [16]). **LowCode- and NoCode-platforms** have emerged as model-driven approaches to enable non-developers to create software applications [19], but without a specific focus on decision support.

3 Decision Support Ecosystem Concept

In this section, we derive the concept of a *decision support ecosystem (DSE)* from existing digital business ecosystems and discuss expected benefits and challenges of DSEs in providing tailored decision support in a timely manner.

3.1 Decision Support Ecosystem Definition

A *digital business ecosystem* is a “socio-technical environment of individuals, organisations and digital technologies with collaborative and competitive relationships to co-create value through shared digital platforms” [22]. The socio-technical entities of a digital business ecosystem are visualized in Fig. 1: A *platform provider* provides and maintains a *shared digital platform* on which *service providers* offer their services. *Service consumers* query the platform to obtain and combine services into a *product* which satisfies the demands of *end users*. In this setting, value (i.e., any financial or nonfinancial benefit [22]) is created for service providers by allowing them to advertise their services via the shared platform, for service consumers by discovering and combining services to implement business ideas more quickly [22], and for end users by being provided with products that satisfy their (individual) demands.

The concept of a digital business ecosystem can be refined based on the types of services which are exchanged via the shared platform. For instance, in

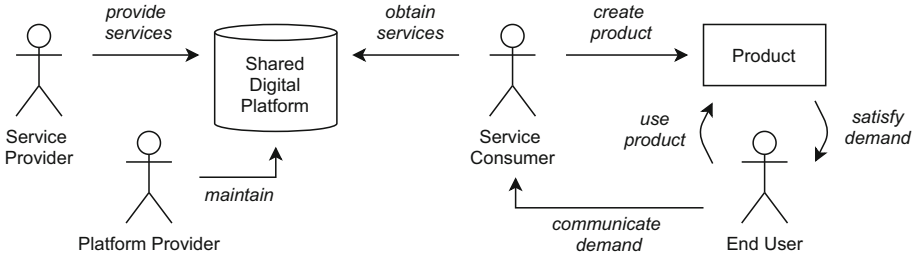


Fig. 1. Overview of entities in a digital business ecosystem

software ecosystems, the services can be software components or applications using a common technology. An example is the *Microsoft Windows* operating system, where developers can provide frameworks utilized by other developers for the development of applications used by end users [3]. In a *data ecosystem*, the primary resource exchanged between platform users is data, but also related resources such as software for data manipulation or infrastructure for data persistency and software execution [18]. Data ecosystems are for instance used in a governmental context to support regulators in policy making [17, 18].

The concept of a digital business ecosystem can be applied to the domain of decision support as follows: The product is a *DSS* which is tailored according to the requirements of an individual end user, i.e., *decision maker*. The DSS is assembled using *decision support services* provided by *decision support service providers*. Similar to the previous explanation of data ecosystems, we expect decision support services to include data, software and computing infrastructure. In the exemplary domain of energy distribution network planning, data might be a forecast of electric vehicle market shares, software might be an algorithm for load forecasting or the optimization of network reinforcements, and infrastructure a high-performance cluster to execute the optimization algorithm. The role of the *service provider* can be assumed by DSS developers who extract reusable functionality from their existing DSS, but also from domain experts such as research institutions or consulting agencies for software and data services as well as cloud providers for infrastructure services. The role of the *service consumer* should ideally be assumed by decision makers using an “end user programming” approach [3] to ensure timeliness of decision support. Assuming that decision makers can communicate their requirements for decision support, the role could also be assumed by DSS developers or domain experts to establish best practices.

As a consequence of the previous explanations and our initially formulated research question, we define a *decision support ecosystem* as follows:

A decision support ecosystem (DSE) is a network of decision makers, DSS developers and domain experts who share requirements for decision support and decision support services (including software, data and computing infrastructure) via a shared digital platform which enables the adhoc composition and execution of decision support services without extensive

software development knowledge to quickly and optimally address an individual decision maker's requirements for decision support.

3.2 Expected DSE Benefits and Challenges

By deriving our concept of a decision support ecosystem from digital business ecosystems, in particular software and data ecosystems, we can expect some of their benefits and challenges to translate to DSEs as well.

Benefit 1 – Mass-Customization: By combining existing services into a DSS, the DSE concept shares similarities with software product lines [3] and enables the development of “a diversity of similar applications” at “lower cost, in shorter time, and with higher quality when compared with the development of single systems” [14]. Combining services offered via the shared platform ensures that all companies, regardless of their research and development budget, can cooperate to fulfill the high demand for customization by users [3].

Benefit 2 – Faster Time-to-Market: The combination of existing services does not only lower development costs, but also results in a faster time-to-market [14]. Consequently, decision makers are able to use the DSS sooner. This benefit is increased when decision makers can apply the customizations themselves, e.g., following an end-user programming approach [3].

Benefit 3 – Innovation: Digital business ecosystems foster innovation [9]. When decision makers and DSS developers exchange their challenges and solution approaches for decision making, we can expect decision makers to profit from new, innovative decision support services. Simultaneously, decision support service providers can receive feedback for their services and use it to improve services or identify new business opportunities [18]. Furthermore, decision makers do not only profit from newly developed services, but also from discovering existing services, especially data, which they can utilize for decision making [18].

These benefits immediately align with our initially formulated research question of providing decision makers with tailored decision support in a timely manner. Nevertheless, an ecosystem approach also introduces challenges:

Challenge 1 – Participation: Value creation based on providing and consuming services in ecosystems only works when enough stakeholders participate in the ecosystem [3, 9, 17]. This requires low entry barriers to encourage participation in the ecosystem [9]. In case of supporting customization by end users, this also includes identifying sufficient abstractions to ensure that end users can intuitively create service-based applications without being limited in the complexity of the applications [3, 18]. Moreover, a governing entity is needed to support service quality [17] and to ensure the sustainability of the ecosystem [9]. A further concern, especially with respect to data, is the consideration of privacy and confidentiality as not every service provider may want to make all of their services publicly available [17].

Challenge 2 – Platform Design: The underlying shared platform of an ecosystem is a deciding factor to what extent the aforementioned ecosystem benefits can be utilized while minimizing the effects of the described challenges [22]. Unfortunately, there is still a lack of technical knowledge and resources to implement such platform [17, 22], especially in the context of decision support.

4 DSE Platform Architecture

The previous section introduces the concept of a decision support ecosystem and discusses its potential benefits and challenges. A particular challenge is the design of the shared platform as key enabler of any ecosystem [22]. In this section, we therefore propose a technological reference architecture for a shared DSE platform. We expect this reference architecture to guide DSE platform implementation in multiple application domains, thereby proving the technical feasibility of DSEs and encouraging future DSE research, e.g., DSE governance.

4.1 Research Approach

We document our reference architecture for a shared DSE platform in form of design principles, i.e., propositions for platform components, user roles and their interactions which can be tailored to a specific application domain. We use the supportive approach described by Möller et al. [15] for the identification of design principles. For this purpose, we first collect meta-requirements which document fundamental functionality requirements of any DSE platform regardless of a specific application domain. We use our experience from the aforementioned research project and the discussion of the DSE concept in the previous sections as a knowledge base for meta-requirement identification. The design principles are subsequently derived as a response to address the meta-requirements. For evaluation purposes, we subsequently demonstrate the design principles in the exemplary application domain of energy distribution network planning.

4.2 DSE Platform Meta-requirements

We identified the following meta-requirements for the shared DSE platform:

MR1 – *Common Terminology.* The ambiguity of VUCA business environments [1, 13] creates the need to establish a common terminology across ecosystem participants. It must therefore be possible to document the entities of the associated application domain as well as potential alternatives among decision makers' goals, restrictions and resources when making a decision. For the exemplary domain of energy distribution network planning, an entity would be a regional electricity network which incurs investment and operating costs that should be minimized while power outages must be avoided. In addition to its primary purpose of establishing a common understanding, the documentation can also be used by ecosystem participants to identify decision making use cases.

MR2 – *Individuality of Decision Support Requirements.* The requirements for decision support vary between individual decision makers, even for the same type of decision within an application domain [10,23]. It is therefore necessary to document and consider the individual requirements of decision makers throughout the DSE.

MR3 – *Discoverability of Decision Support Services.* The fundamental idea of business ecosystems is to co-create value by providing and consuming services [22]. In the DSE context, service discoverability helps to find those decision support services that best align with an individual decision maker’s requirements for decision support. From a service provider’s point of view, the visibility of an offered decision support service likely results in economic gain. The technical platform of a DSE must therefore support the discoverability of decision support services, i.e., software, data, infrastructure. However, limiting discoverability can also be desired in certain cases, either to ensure privacy and confidentiality to avoid the misuse of personal data [17] (e.g., historical electricity consumption per household), or to delay access to innovative decision support services as a means to keep an advantage over direct competitors.

MR4 – *Holistic Decision Support Process.* Due to the complexity of business environments [1,13], identifying a decision recommendation is a multi-stage process with potentially multiple activities for the selection, preparation, manipulation, analysis and visualization of data. By definition, a single decision support service can and should not cover all of these activities to foster reusability. Consequently, multiple “single-purpose” decision support services must be combined to represent a holistic decision support process. In this context, it is important to ensure the correctness of information exchange between decision support services, otherwise the decision support may fail due to technical errors.

MR5 – *Timeliness of Decision Support.* The volatility of business environments requires decision makers to identify optimal decisions in a short amount of time [23]. The available time should therefore not be spent waiting for the individualized DSS, but instead actually using the DSS. Consequently, both the previously mentioned assembly of decision support services into an individualized decision support process as well as its subsequent utilization must be fast. From a decision maker’s point of view, participation of other ecosystem stakeholders such as DSS developers should be minimized for this purpose.

MR6 – *Experience & Innovation.* Once the tailored decision support has been used by a decision maker, it can be rated considering for instance alignment with requirements for decision support, quality of decision recommendations or quality of individual decision support services. On the one hand, such feedback helps selecting a decision support service among multiple alternatives or identifying which service assemblies best address certain requirements for decision support. This naturally requires some kind of traceability to track which decision support services were selected based on which requirements. On the other hand, the feedback can be used by service providers to improve their decision support services [18] or identify functionality gaps to be filled by additional services.

4.3 DSE Platform Design Principles

We identified the following DSE platform design principles for architectural components, user roles and their interaction as a response to address the previously described meta-requirements. An overview of the reference architecture is given in Fig. 2. Note that each DSE participant can potentially assume multiple roles.

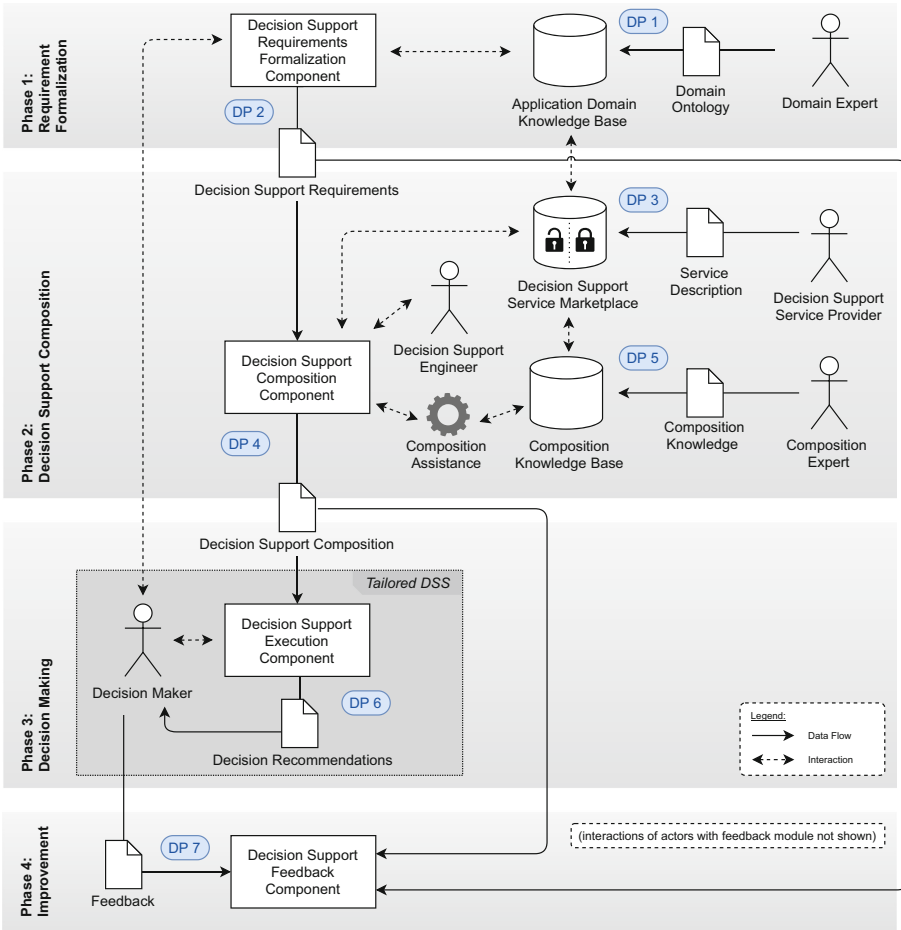


Fig. 2. Proposed reference architecture for the shared platform of a DSE

DP1 – Application Domain Knowledge Base. We expect domain experts to capture entities of the application domain and decision making characteristics with respect to these entities in the form of formal domain ontologies. The domain ontologies are aggregated into a application domain knowledge base

to establish a common terminology between ecosystem participants for other architecture components, thereby addressing MR1 – *Common Terminology*.

DP2 – Decision Support Requirements Formalization. Individual decision makers use the *decision support requirements formalization component* to formally document their individual requirements for decision support based on the *application domain knowledge base*. The requirements documentation is used in other components to ensure that the created decision support aligns with a decision maker’s requirements, thereby partially addressing MR2 – *Individuality of Decision Support Requirements*.

DP3 – Decision Support Service Marketplace. *Decision support service providers* make their *decision support services* available via a central *service marketplace*. For this purpose, they have to provide a *service description*. Based on the *application domain knowledge base*, the service description includes information about the service’s functionality, resource requirements, guarantees (i.e., with respect to availability) and how to invoke the service. The service marketplace is divided into a *public* and a *private* section to control whether a service is available to all or just a limited subset of ecosystem participants. The service marketplace addresses MR3 – *Discoverability of Decision Support Services*.

DP4 – End-User Decision Support Composition. A *decision support composition* describes how to integrate multiple decision support services to address all activities of a decision support process. The composition is created using the *decision support composition component* based on the requirements for decision support specified by an individual decision maker and the decision support services made available via the *decision support service marketplace*. The *composition component* is primarily operated by a *decision support engineer*. Despite the reference to “engineer” in the role title, we do not expect this actor to have a software engineering or software development background. Instead, we suggest to design the composition in a way that allows end-user programming, following a low-code or no-code development paradigm [19]. The role of the *decision support engineer* could thereby be assumed by decision makers or other domain experts. The decision support composition addresses MR2 – *Individuality of Decision Support Requirements*, MR4 – *Holistic Decision Support Process* and MR5 – *Timeliness of Decision Support*.

DP5 – Knowledge-Based Composition Assistance. The *composition assistance* is a digital assistance system that also interacts with the *decision support composition component* in order to support the *decision support engineer* during service composition. The composition assistance may ensure the consideration of best practices or anti-patterns for decision support in the application domain. For this purpose, it utilizes a *composition knowledge base* which is filled by domain experts. The knowledge-based assistance partially addresses MR5 – *Timeliness of Decision Support* and MR6 – *Experience & Innovation*.

DP6 – Decision Support Execution. The created *decision support composition* is forwarded to the *decision support execution component*. The component

orchestrates the services as specified by the composition. By automating this orchestration, the invocation of the included services is transparent to the decision maker, i.e., from the point of a decision maker, they still interact with a “traditional” DSS. This includes the decision maker specifying assumptions or being able to inspect decision recommendations, e.g., by using interactive visualizations. The automated execution addresses MR4 – *Holistic Decision Support Process* and MR5 – *Timeliness of Decision Support*.

DP7 – Decision Support Feedback. Before, during or after execution of a decision support composition, the decision maker may provide *feedback* for the provided decision support. All stakeholders can view this feedback via the *feedback module* and use it to improve their contribution to the DSE. Most importantly, domain experts can identify new composition knowledge. Service providers can analyze room for improvements in their provided decision support services or identify a demand for new services. Decision support engineers can update a composition if they failed to address a requirement for decision support or decision makers can update their requirements if they misunderstood the requirements formalization – an indicator for domain experts to improve the descriptions in the *application domain knowledge base*. These feedback-based improvements address MR6 – *Experience & Innovation*.

4.4 Demonstration for Energy Distribution Network Planning

Ideally, one would implement the platform reference architecture described by the previous design principles for a concrete application domain to evaluate its practicability (cf. “field-testing” in [15]). However, this requires additional refinement of individual architecture components, e.g., (1) adapting existing approaches for the formalization of requirements, services and service compositions to decision support, (2) assessing and adapting low-code platforms for the use of these formalizations, and (3) the formalization, extraction and consideration of composition knowledge via the composition assistance. As these research problems are out of scope for this paper, we defer the field-test to future work and instead demonstrate the application of platform components with an experience-based example in the domain of energy distribution network planning.

In our example application derived from [10], we consider decision recommendations for the reinforcement of regional electricity distribution networks. With respect to **DP1 – Application Domain Knowledge Base**, an electricity network is characterized by capital and operational costs, electrical loads of buildings connected to the network, and a redundancy strategy which describes how many transformers in the network may simultaneously fail without impacting the functionality of the network. Building loads can describe historical or expected future loads. With respect to **DP3 – Decision Support Service Marketplace**, we consider the following services:

1. **LF-Scale:** Computes expected future building loads by multiplying current loads with a constant scaling factor.

2. **LF-Extrapolation:** Computes expected future building loads by extrapolating the historical loads of the last 5 years.
3. **LF-Technology:** Computes expected future building loads by predicting which building might adopt which consumer technology (e.g., electric vehicle, charging station, heat pump). Has a longer runtime than other LF services.
4. **Marketshares:** Data with expected future consumer technology market shares.
5. **OPT-Exact:** Receives an electricity network with expected future loads and uses a mathematical exact optimization algorithm to recommend transformers which should be replaced to support loads while minimizing costs. Considers network redundancy, i.e., one transformer might fail without impacting network reliability. Requires a high-performance cluster to be executed. Runtime scales exponentially in the number of loads to consider. Guarantees that the found optimum equals the theoretical optimum (100% optimality).
6. **OPT-Heuristic:** Similar to **OPT-Exact**, but uses a heuristic optimization algorithm which runs on commodity hardware bundled with the service. Does not support redundant network design. Runtime scales linearly in the number of building loads to consider. Only guarantees 80% optimality.
7. **HPC:** High-Performance Cluster inducing high costs when used.
8. **NR-Reduce** Reduces the size of a provided electricity network by intelligently aggregating building loads.
9. **NR-Revert** Reverts a network reduction by translating all reinforcement recommendations from the reduced network to the original electricity network.

With respect to **DP2 – Decision Support Requirements Formalization**, we consider two distribution network operators (DNOs) who want to optimize their electricity network. The first DNO documents that they want a cost-minimizing, redundant electricity network and are not constrained in monetary or temporal resources. The second DNO documents that they also want a cost-minimizing, redundant network but are very limited in resources. With respect to **DP4 – End-User Decision Support Composition**, the first DNO might initially combine the services **LF-Extrapolation** and **OPT-Exact** services. With respect to **DP5 – Knowledge-Based Composition Assistance**, the composition assistance might point out that the **OPT-Exact** service is missing the **HPC** service and that experience shows that the **LF-Technology** service, albeit more expensive, provides a better load forecast than the **LF-Extrapolation** service. The first DNO can accept these improvements and the composition can be executed automatically. The second DNO might initially combine the **LF-Extrapolation** and **OPT-Heuristic** services to not violate any resource constraints. The composition assistance might point out that the decision maker is missing the historical load data required for the **LF-Extrapolation** service and that the **LF-Scaling** service could be used instead. Furthermore, the **OPT-Heuristic** service does not align with the documented requirement of network redundancy. Instead, the second DNO can apply a network reduction with the **NR** services and utilize the **OPT-Exact** service. **DP6 – Decision Support Execution** may be automated by providing the services as web services and

invoking them via a workflow engine. In the context of **DP7 – Decision Support Feedback**, the second DNO may have forgotten to revert the network reduction and the composition knowledge can be updated to reflect that the use of the **NR-Reduce** service implies that the **NR-Revert** service must be invoked sometime later. Any DNO may communicate demand for decision support functionality which is currently not considered, e.g., the identification of robust reinforcement recommendations across multiple potential future load scenarios.

5 Summary and Future Work

Decision support systems are crucial in helping decision makers to quickly identify optimal decisions in VUCA business environments. However, the identification of optimal decisions requires decision support that is quickly tailored to the individual requirements of decision makers with consideration of their goals and constraints during decision making. The state of the art in decision support and software engineering only partially supports the creation of tailored decision support in a timely manner. We therefore discussed the concept of a decision support ecosystem which allows DSS developers, decision makers and domain experts to collaborate using a shared platform to provide and consume digital decision support services, i.e., software, data and infrastructure. In this context, we presented a reference architecture for the shared platform of a DSE characterized by the assisted end-user composition of decision support services which are available from a central service marketplace.

During the demonstration of the reference architecture, we already presented aspects of individual architecture components that require additional research before the platform can be implemented for a concrete application domain. In future research, we want to address these research questions and subsequently evaluate our presented DSE concept in form of a field-test by implementing the reference architecture for energy distribution network planning. Inspired by digital business ecosystems, additional future work may focus on transitioning into a DSE (cf. [3, 22]), governing a DSE to ensure its health and sustainability (cf. [9, 18, 21, 22]) or supporting DSE research, e.g., by modeling DSEs (cf. [4]).

References

1. Bennett, N., Lemoine, G.J.: What a difference a word makes: understanding threats to performance in a VUCA world. *Bus. Horiz.* **57**(3), 311–317 (2014)
2. Bhargava, H., Sridhar, S., Herrick, C.: Beyond spreadsheets: tools for building decision support systems. *Computer* **32**(3), 31–39 (1999)
3. Bosch, J.: From software product lines to software ecosystems. In: *Proceedings of the 13th International Software Product Line Conference*, pp. 111–119. Carnegie Mellon University (2009)
4. Boucharas, V., Jansen, S., Brinkkemper, S.: Formalizing software ecosystem modeling. In: *Proceedings of the 1st International Workshop on Open Component Ecosystems, IWOCE 2009*, pp. 41–50. ACM (2009)

5. Demirkan, H., Delen, D.: Leveraging the capabilities of service-oriented decision support systems: putting analytics and big data in cloud. *Decis. Support Syst.* **55**(1), 412–421 (2013)
6. Drissen-Silva, M.V., Rabelo, R.J.: A collaborative decision support framework for managing the evolution of virtual enterprises. *Int. J. Prod. Res.* **47**(17), 4833–4854 (2009)
7. Fazlollahi, B., Parikh, M.A., Verma, S.: Adaptive decision support systems. *Decis. Support Syst.* **20**(4), 297–315 (1997)
8. Gottschalk, S., Yigitbas, E., Nowosad, A., Engels, G.: Situation- and domain-specific composition and enactment of business model development methods. In: Ardito, L., Jedlitschka, A., Morisio, M., Torchiano, M. (eds.) *PROFES 2021*. LNCS, vol. 13126, pp. 103–118. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91452-3_7
9. Jansen, S., Cusumano, M.A.: Defining software ecosystems: a survey of software platforms and business network governance. In: *Software Ecosystems*. Edward Elgar Publishing (2013)
10. Kirchhoff, J., Burmeister, S.C., Weskamp, C., Engels, G.: Towards a decision support system for cross-sectoral energy distribution network planning. In: *Energy Informatics and Electro Mobility ICT* (2021)
11. Kita, P., Szczyrba, Z., Fiedor, D., Letal, A.: Recognition of business risks when purchasing goods on the internet using GIS: experience from Slovakia. *Electron. Commer. Res.* **18**(3), 647–663 (2017)
12. Liang, Z., Jia, X.: The technical system establishment and application of decision support system generator based on component coordination. In: *2014 Seventh International Symposium on Computational Intelligence and Design*, vol. 1, pp. 229–233 (2014)
13. Mack, O., Khare, A.: Perspectives on a VUCA world. In: Mack, O., Khare, A., Krämer, A., Burgartz, T. (eds.) *Managing in a VUCA World*, pp. 3–19. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-16889-0_1
14. Metzger, A., Pohl, K.: Software product line engineering and variability management: achievements and challenges. In: *Future of Software Engineering Proceedings*, pp. 70–84. ACM (2014)
15. Möller, F., Guggenberger, T.M., Otto, B.: Towards a method for design principle development in information systems. In: Hofmann, S., Müller, O., Rossi, M. (eds.) *DESRIST 2020*. LNCS, vol. 12388, pp. 208–220. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64823-7_20
16. Mustafin, N., Kopylov, P., Ponomarev, A.: Knowledge-based automated service composition for decision support systems configuration. In: Silhavy, R., Silhavy, P., Prokopova, Z. (eds.) *CoMeSySo 2021*. LNNS, vol. 231, pp. 780–788. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90321-3_63
17. Oliveira, M.I.S., de Fátima Barros Lima, G., Lóscio, B.F.: Investigations into data ecosystems: a systematic mapping study. *Knowl. Inf. Syst.* **61**(2), 589–630 (2019)
18. Oliveira, M.I.S., Lóscio, B.F.: What is a data ecosystem? In: *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, dg.o 2018. ACM (2018)
19. Sahay, A., Indamutsa, A., Di Ruscio, D., Pierantonio, A.: Supporting the understanding and comparison of low-code development platforms. In: *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 171–178 (2020)
20. Savió, D.A., Bicik, J., Morley, M.S.: A DSS generator for multiobjective optimisation of spreadsheet-based models. *Environ. Modell. Softw.* **26**(5), 551–561 (2011)

21. Schwichtenberg, B., Engels, G.: SecoArc: a framework for architecting healthy software ecosystems. In: Muccini, H., et al. (eds.) ECSA 2020. CCIS, vol. 1269, pp. 95–106. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59155-7_8
22. Senyo, P.K., Liu, K., Effah, J.: Digital business ecosystem: literature review and a framework for future research. *Int. J. Inform. Manage.* **47**, 52–64 (2019)
23. Shafiei, F., Sundaram, D., Piramuthu, S.: Multi-enterprise collaborative decision support system. *Expert Syst. Appl.* **39**(9), 7637–7651 (2012)
24. Stănescu, I.A., Ştefan, A., Filip, F.G.: Cloud-based decision support ecosystem for renewable energy providers. In: Camarinha-Matos, L.M., Baldissera, T.A., Di Orio, G., Marques, F. (eds.) DoCEIS 2015. IAICT, vol. 450, pp. 405–412. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16766-4_43
25. Weskamp, C., Koberstein, A., Schwartz, F., Suhl, L., Voß, S.: A two-stage stochastic programming approach for identifying optimal postponement strategies in supply chains with uncertain demand. *Omega* **83**, 123–138 (2019)
26. Zha, X.F., Sriram, R.D., Fernandez, M.G., Mistree, F.: Knowledge-intensive collaborative decision support for design processes: a hybrid decision support model and agent. *Comput. Ind.* **59**(9), 905–922 (2008)