Improving the Assesment of Advanced Planning Systems by Including Optimization Experts' Knowledge

Melina Vidoni, Jorge Marcelo Montagna and Aldo Vecchietti
INGAR CONICET-UTN, Avellaneda 3657, Santa Fe, Argentina

Keywords: Architecture Evaluation, Advanced Planning Systems (APS), Process Optimization, Architecture Trade-off Analysis Method (ATAM).

Abstract: Advanced Planning Systems (APS) are core for many production companies that require the optimization of its operations using applications and tools such as planning, scheduling, logistic, among others. Because of this, process optimization experts are required to develop those models and, therefore, are stakeholders for this system's domain. Since the core of the APSs are models to improve the company performance, the knowledge of this group of stakeholders can enhance the APS architecture evaluation. However, methods available for this task require participants with extensive Software Engineering (SE) understanding. This article proposes a modification to ATAM (Architecture Trade-off Analysis Method) to include process optimization experts during the evaluation. The purpose is to create an evaluation methodology centred on what these stakeholders value the most in an APS, to capitalize their expertise on the area and obtain valuable information and assessment regarding the APS, models and solvers interoperability.

1 INTRODUCTION

Advanced Planning Systems (APS) incorporate models and solutions algorithms to contribute to the planning optimization of different areas of an organization (Stadtler, 2005). They can be used either to improve the performance of a supply chain or the internal production planning of a company (Fleischmann et al., 2015). Consequently, APS is core for the operation of those organizations that implement them.

For the implementation of an APS there exist two types of developers, both of them with dissimilar academic backgrounds, interests and objectives (Kallestrup et al., 2014): the traditional team of software engineers, and the team of process optimization experts that generate the models (Gayialis and Tatsiopoulos, 2004). Despite their differences, the expertise and point of view of both stakeholders groups is highly needed when developing and/or maintaining an APS.

The development of an information system development is based on a Software Architecture (SwA), which should guarantee the applicability of both Functional Requirements (FR) and Quality Attributes (QA). The successfulness of a SwA can be assured by performing an evaluation process (Shanmugapriya & Suresh, 2012). An architecture having a good evaluation provides the required groundwork to develop high quality Information Systems (Angelov et al., 2012).

There are several available methodologies to assess SwA (Dobrica and Niemela, 2002; Ionita et al., 2002), all of them are only focused on technical Software Engineering (SE) aspects, and do not include other types of stakeholders, like those existent in the APS domain. Also, these methods are only focused on traditional SwA, and do not consider the evaluation of Reference Architectures (RA), which are a wider and more abstract design concept, that aims to define and characterize a domain of systems instead of a particular implementation (Martínez Fernández et al., 2013).

Several proposals to adjust existing evaluation methods can be found in the academic literature. Angelov et al., (2008) propose modifications to ATAM (*Architecture Tradeoff Analysis Method*) in order to adapt it to consider particular characteristics of RA. Sharafi (2012) created an ATAM modification to detect potential problems threatening the system from the stakeholder’s point of view. Also, Heikkilä et al., (2011) used a reduced version of several evaluation methodologies, in order to approach a CMM control system assessment. Finally, Diniz et al., (2015) used an ATAM utility tree to model health care education systems.

Vidoni, M., Montagna, J. and Vecchietti, A.
Improving the Assesment of Advanced Planning Systems by Including Optimization Experts’ Knowledge.
DOI: 10.5220/0006362005100517
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 510-517
ISBN: 978-989-758-248-6

Most research regarding APS is done through process optimization, and therefore, there is a lack of SE approaches (Henning, 2009; Framinan and Ruiz, 2010). A proposal on this area created a RA for APS focused on the optimization of intra-organization planning problems (Vidoni and Vecchietti, 2016). However this architecture has not been yet evaluated, due to the absence of a suitable method.

This article proposes a novel modification for ATAM: applying the methodology to a RA, and alter the steps to include process optimization experts. This is used to evaluate the interoperability and relation between the models, solvers and the APS, while keeping a focus on their expertise area. Diversifying the type of participants during the evaluation allows considering a wider range of points of views, different goals, and understanding why stakeholders value the APS differently.

2 ADVANCED PLANNING SYSTEMS (APS)

APS are information software systems conceived to solve production planning problems by means of an advanced solving approach; they must interoperate with the Enterprise System (ES), and work with a set of models and solvers (Vidoni and Vecchietti, 2015). Although there are many definitions in the literature (Fleischmann and Meyr, 2003), the chosen one consolidates concepts proposed by several authors, with the focus on APS as systems and not only in the problems to be solved; they also consider the use of several solution methods for those problems, by introducing the concept of *solving approaches*.

2.1 Stakeholders and Requirements

The APS domain is defined through Functional Requirements (FR) and Quality Attributes (QA) that are more generic and have a higher abstraction level than regular FR and QA, since they frame a domain instead of an implementation.

Since an APS development requires the implementation of both software and models, regardless of their solving approach, the FR and QA are related to both issues. This situation reflects the significance of both types of developers (Gayialis & Tatsiopoulos, 2004). The challenge is to address those different requirements, goals and constraints, and the stakeholders that have different responsibilities and roles, but also very dissimilar academic backgrounds (Kazman et al., 2005).

2.2 Reference Architecture (RA)

RA aim to clarify the boundaries and features of domains of systems (Northrop, 2003). They are based in generic functionalities and data flow (Cloutier et al., 2010), and simplify the design and development in multiple projects, by working in an extensive, more abstract and less defined context, with stakeholders only defined as target groups (Angelov et al., 2012).

This is the case of APS-RA (Vidoni and Vecchietti, 2016), an RA built for APSs. It is defined using the "4+1" View Model, and its documentation includes *variation points*, which give the architecture the ability to be adapted to different situations in pre-planned ways and with minimal effort (Bachmann et al., 2003; Bosch et al., 2002).

APS-RA goal is to unify the development of an APS, including the generic needs of the models and solvers, while keeping in mind how they should integrate with the software, in order to provide a robust and maintainable base design.

3 ATAM-M

As APS-RA defines a domain, it plays a major role in determining each implemented APS quality: decisions made at architectural level can help or interfere with achieving business goals and meeting FR and QA in future projects (Shaw and Clements, 2006). However, an evaluation process can reduce these risks (Shanmugapriya and Suresh, 2012).

Evaluating APS-RA has difficulties. There are those defined by the features of RA, such as high abstraction level, lack of individual stakeholders, etc., and those intrinsic to the domain, i.e. dissimilar types of developers, the FR/QA dual definition, and so on. Consequently, there are no readily available evaluation methods that would cover these issues.

This article proposes ATAM-M, a modification of ATAM (*Architecture Trade-off Analysis Method*) (Kazman et al., 2000) to adapt it to fit the previously mentioned issues.

ATAM is selected as base, because it evaluates more QA than other methodologies (Ionita et al., 2002) and is based on the "4+1" View Model, and can be applied to lead participants to focus on what each of them considers core for the architecture under study (Bass et al., 2013).

From the two core stakeholder groups be involved during the evaluation, process optimization experts, referred as SAS, does not usually have an extensive SE background. Therefore, ATAM-M process is performed in two Stages:

- An innovative and shorter stage, with SAS and planners. The analysis is done with the FR and QA that affect the optimization models, and the goal is to evaluate how they interoperate with the APS. This includes how the models reflect the QA, and if they consider the FR at all.
- A traditional stage focused on architectural analysis, with regular software developers, focused on the SE aspects of the system.

4 SAS IN ATAM-M: STAGE 1

Although ATAM-M implementation includes both Stages, this article focuses on the first one, aimed to work with SAS and planners.

In order to achieve a successful evaluation, participants receive instructions such that they can understand APS-RA, and are able to propose changes, refinements, and an assessment that fulfils the Stage goals (Weinreich and Buchgeher, 2012). Therefore, it is key to decide what parts of the APS-RA are significant to them, to reduce unnecessary complexity and only use valuable data related to what parts of the APS-RA are relevant to them.

Stage 1 is designed to allow participants to focus on what they most value on an APS, capitalize their knowledge by keeping the process centred on their expertise, and obtain an assessment that other points of view would not be able to provide. In particular, SAS and Planners value the models development, their accuracy, maintainability, and how they interoperate with solvers while integrating to APS.

ATAM-M does not include all original ATAM steps, and then the outputs are different. The process to generate ATAM-M is done iteratively, and resulting changes to steps are summarized in Table 1. Also, Table 2 presents the method outputs.

Initially, specific SE steps are discarded, along with those that cannot add value to SAS. Both the QA Questions and the Utility Tree (UT) are steps accepted from ATAM without modifications, and only the input data is different. Questions are elaborated from the model-related QA, and later used to help participants understand non-functional qualities of the models and their interrelation with the architecture.

Regarding the UT, the term scenario is replaced by node, to and avoid nomenclature mistakes, since in the APS-RA, scenarios are the results obtained by executing a model, regardless of its solving approach, and the “4+1” View Model includes a ‘Scenario View’. The remaining ATAM steps require changes to be included on ATAM-M.

The introduction is short and concise to avoid overwhelming participants with information they cannot draw value from. Since the APS-RA is documented with viewpoints targeted to different stakeholders, only some views are used: Process and Scenario View are the most detailed, with a brief introduction of the Logical View, while both Development and Physical views are avoided, since they are targeted to the software team. Also, only those variation points that directly affect the model, solver or their relation with the APS are presented.
The selected information is in line with the stakeholders’ point of view regarding the system. However, it is also enough to understand the APS-RA goal, and what should be evaluated.

Steps regarding architectural tactics are also modified, since this is a SwA concept outside of the Stage 1 scope. The ATAM-M innovative proposal is

Table 1: ATAM-M steps, their reasoning and changes compared to original ATAM for Stage 1.

Original ATAM		ATAM-M Stage 1	
Step	**Outputs**	**Decision**	**Reasoning**
1-3: Present ATAM, business drivers and RA.		Modified	Overview directed to SAS and Planners, avoiding SE-specific views, not valuable for them. The method steps are not introduced, except for the project goals.
4: Identify approaches.	Architectural patterns.	Discarded	Not applicable to the goal of the stage, nor the expertise area of the participants.
5: Generate Utility Tree	L1: QA Questions	Accepted	Questions are elaborated only with model-related QA.
	Utility Tree (UT)	Accepted	The term “node” replaces “scenario”, to avoid glossary issues with APS-RA. The step goals are the same as in the original.
6: Analyze architectural approaches	List of tactics.	Modified	Participants search for Design Decisions (DD) that directly or indirectly affect the model-related QA. Vague decisions become potential risks, while measurable ones become sensibility points.
	Risks, sensibilities and trade-offs.		
	L2: Tactics Questions	Modified	Questions are performed over the, and aim to find risks and sensibility points on the UT nodes.
7: Brainstorm and rank scenarios.	L2: Prioritized scenarios	Discarded	Due to the focus of the stage, generating new scenarios would not produce new results, and would not engage the participants.
8: Analyze approaches	Compare nodes and scenarios.	Discarded	Since the level 2 scenarios are not generated, this step is not applicable.

to use an equivalent output that allows studying the same concept –mechanisms to implement QA– but on the models, instead of the software.

Design Decisions (DD) are structural choices made when planning and developing a model that improve or hinder its relation with the APS, and which QA can be applied. In some cases, DD are also related to the software. For example, a model that uses a heuristic solving approach, in which both the model and the solver are the same; therefore, their development affect the whole project. Vague DD can become risks for the systems, and those measurable can be identified as sensibility, while those affect differently each QA represent trade-offs.

DDs related to the software can be identified by the expertise of SAS, while those applicable to models need to be proposed. This acts as variation points: determining which QA requires a higher priority towards the applicable DD.

Another modified step is Questions Level 2, which are asked regarding the DD, aiming to understand how they affect all the nodes on the UT. Questions are proposed to discover risks, non-risks, sensibility points and trade-offs, and are generated by brainstorming.

Both Questions Level 1 and 2 are classified into *stimuli*, events on the model or solver that may cause changes in the architecture, *responses*, quantities related to them, and *decisions*, aspects that impact on achieving responses.

Finally, participants generate a list of risks, sensibility points and trade-offs, and DDs that enhance a given QA but undermine another are presented as trade-offs. This output can greatly improve the development of models, and synchronize it with the design and architectural goals established for the APS as a whole. Knowing which choices can potentially jeopardize the architecture reduces business risk and time lost reworking and changing the models or software.

5 ATAM-M APPLICATION

Stage 1 recruited participants are researchers and professional experts active on planning and scheduling problems, that have academic and industrial experience. Steps are performed in the original order, and starts with the Architect introducing the APS-RA. Each step is presented as a task, and before starting, the Architect offers a small clarification on its goal and process to follow.

5.1 Results

QA Questions is generated first, while participants engage in a brainstorming regarding the QA. Questions are obtained in two different ways: they are explicitly proposed by stakeholders or they are derived from the generated discussion. This proportion can be seen in Figure 1. Consequently, 56 questions are generated Examples are:

- How much does the solving approach affect the solutions quality? (Decision)
  Which decisions can be taken by the user and which are automatized? (Stimuli)
  • Should performance measures vary for each solving approach or each model? (Response)
  • How is determined the maximum time a model can use to execute, with a normal use of hardware resources? (Response).

This activity helps stakeholders to understand the concept of QA-originally from SE and foreign to optimization-, while relating it to the model and its non-functional qualities. The knowledge built through this step acts a base for the next activities.

Table 2: ATAM-M outputs, required input data, goals and changes reasoning.

Output	Input	Goal	Reasoning
QA Questions (Level 1)	Model-related QA.	Increase the understanding of each QA, analyze their applicability to the APS-RA, their impact and how are addressed.	Participants only work with model-related QA. These changes on the input data are enough to produce a different result.
Utility Tree (UT)	Model-related QA and FR.	Creating nodes reduces the abstraction level by concretizing vague qualities through examples, implementations and interactions. They lead to the priorities, development precedencies, and the reasoning behind them.	Using FR and QA that only affect the model produces a UT focused on studying aspects related to the model. Nodes should still represent use case implementations, and growth and explorative scenarios.
Design Decisions (DD) Lists	APS-RA views. UT nodes.	Helps to detect parts of the architecture that affect the model-solver-APS interrelation, without using specific SE knowledge. It helps identify trade-offs on the QA.	Tactics requires SE knowledge and do not contemplate particularities of the models, therefore it is required to identify choices that affect the model-related QA
DD Questions (Level 2)	Generated list of DD.	They study DD to simplify identifying risks, sensibilities and trade-offs.	Since the input data has a different focus, the results of this output also change.
Risks, Sensibilities and Tradeoffs	DDs, L2 Questions.	These lists help reduce analysis time and simplify development choices on specific case implementations.	DD can be applied on the models, while tactics are SE-concepts pertinent to the architecture.

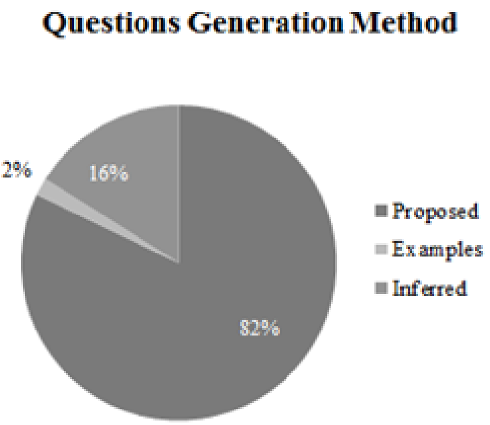

Figure 1: Proportion of how questions are obtained.

After that, the UT brainstorming produces 32 nodes, with 7 of them being high rank. They are either directly proposed, or inferred from a subsequent discussion of ideas between participants. In this case, there is the same number of both types. Since presenting the UT would be too extensive, Table 3 summarizes only an extract of the results, highlighting some of the most relevant nodes.

The UT allows to identify situations in which QA are reflected on the architecture, to study how they react to the design, if they are enforced or hindered. Generated nodes can define where the design priorities are, and which development precedence exists. During this application of ATAM-M to APS-RA, stakeholders focus on maintainability and model exception management.

On the next step, DD are generated upon the analysis of 10 nodes. 33 DD are identified, but many of them are applied to several nodes, resulting in 52 nodes-DD relations. Some of them are:

- Associate models and solutions.
- Consistency check with historical data.
- Process monitor during scenarios solving.
- Available resources for model solving.
- Limiting solver execution parameters.
- Create documentation for each model.

With this, 52 Questions Level 2 are generated, equally proposed as inferred from the brainstorming. Examples of these are:

- Do language semantics limit changes available to a model? (Decision, risk).
- Is the model modifiability affected by the lack of documentation? (Response, risk).
- How much working time is acceptable for a senior user to learn to manage the solutions traceability? (Response, sensibility point).
- How do error tolerance changes affect the solutions quality? (Stimuli, trade-off).

These Questions are used to identify a list of risks, sensibilities and trade-offs for the. From all of them, almost 61% is associated to at least one risk, 42.5% to at least one sensibility and 39.4% to at least one trade-off (see Fig. 2).

As can be seen, a DD can have several aspects. This makes them similar to tactics, including their advantages and drawbacks, but associated to models instead of the software. It also shows that DD are able to provide models with predefined means to achieve different sets of goals, simplifying the decision making process, pointing towards aspects requiring special consideration during early stages.

Table 3: Extract of relevant nodes for the resulting Utility Tree.

	Quality Attributes				**Rank**	
	Attribute	**Sub-Attribute**	**Refinement**	**Node**	**Priority**	**Difficulty**
Usability	Compatibility	Interoperability	Input data depuration	The system reads input data for an execution, and negative values are found. The scenario execution is stopped and the user notified.	High	Medium
		Co-Existence	Solver Modification	A model solver is changed, but it is transparent for the user. The existent models do not need changes.	High	Medium
	Maintainability	Modularity	Model updates	Due to changes on the organization the current models become obsolete. Changing and adapting them do not require more than X weeks.	High	High
	Functional Stability		Solver development	The organization decides to implement a heuristic method. The solver is developed *ad-hoc* in-house, as a module. Development does not take more than X weeks.	High	High

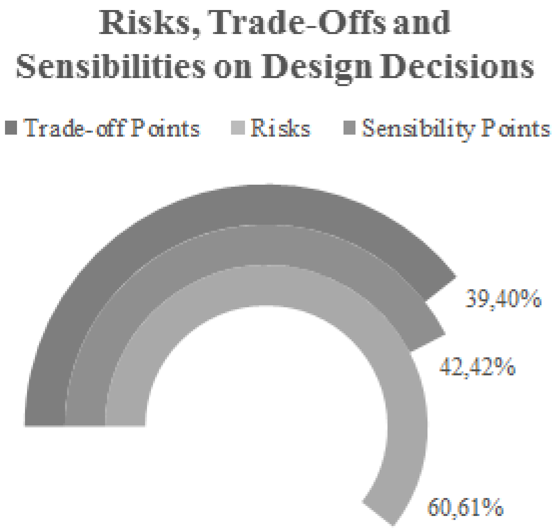

Figure 2: Effect percentage of DDs regarding QA.

5.2 APS-RA Assessment

At the end of Stage 1, the participants assess the APS-RA, including proposed DDs. They conclude that APS-RA works as a framework to simplify the communication between APS and models due to the wide range of considerations, not limiting the solving approach to Operation Research, considering production strategies, and more.

Participants also reflect that the APS-RA enforces the selected QA, the variations are pre-planned in order to maximize the interoperability. They arrive to this conclusion after analyzing the obtained results, answering the questions they made, and comparing them to the proposed DD.

However, new possible features for the APS-RA are elicited. These are considered during the ATAM-M process, and then included in the RA.

The first discovered featured, named *Manage Restrictions* is added as a new FR, impacting on all views, except the Physical View, as new components and processes must be included. The second feature, related to the traceability of solutions, is added as an extension of two existent FR, *Scenario Generation* and *Storage*. Thus, the modification to the RA is only seen on the Process and Scenario View. The documentation of the APS-RA is modified. Changes are present in the FR, diagrams, element descriptions, variation points, and glossary. These FR are detailed on Table 4.

Another aspect included is a variation point regarding elimination of existent models, objectives and parameters: they can be completely deleted, or simply marked as unavailable. This does not impact the APS-RA beyond reworking the description of the corresponding FR and components definitions, but allows the adding traceability to the models, and reducing chances of inconsistency when keeping registries related to previous optimization plans.

Minor changes are also suggested, such as improvements on the FR redaction to clarify concepts, reduce ambiguity and remove wording that could imply a choice on a variation point.

5.3 Discussion of Stage 1 Process

In the obtained results, both QA Questions and UT nodes are focused on the same attributes. Conclusions are that participants are consistent on assigning relevance, revealing their interest to study those attributes. The more frequently addressed QA are Correctness, Interoperability, Modifiability and Operability, which correspond to the Stage 1 goal.

DD provide interesting results, as some of them are related to the software and must be evaluated during the ATAM-M Stage 2, such as those relevant to the available hardware resources. Others can be seen as tactics applied to models, i.e. error tolerance, exceptions recovery, and execution time. However, some DD present a parallelism between software and model elements; e.g., those referred to available documentation, versioning, and traceability.

During brainstorming, participants directed the discussion towards one the traceability of models.

Table 4: New requirements added as a result of the evaluation.

Requirement	Type	Model Aspect	Software Aspect
Manage Restrictions	New Requirement	Subject to the solving approach, the model should run with some or none of its restrictions.	For each model, the user should be able to analyze the model restrictions by using associating those that share the same goal into 'conceptual groups'; this allows him/her to select which groups to apply to the scenario
Scenario Generation	Included on Existent FR	Continuing an interrupted solve may only be possible on some approaches.	If the generation is interrupted or if the execution timed out, the user should have the option to restart the solving from the last optimal solution, if exists.
Scenario Storage	Included on Existent FR		Traceability is done on the final solution, and/or in the latest intermediate feasible solution. [...] Solutions should be stored for a given configurable time before deleted. [...] Each solution should be linked to the used model configuration, so if a model is updated the registry will be linked to the older model version

The main idea proposes that new file versions should be added, so that the previous ones can still be related to generated plans, and to provide an overview of changes done to each model.

This can be understood as Version Control (VC): the management of changes elements related to the project, associated to the user that generated it. VC contributes in many ways to a project, including definition and tracking of artifacts, teamwork support, among others (Wu et al., 2004), and using it during the development of models could contribute positively to the product evolution, by facilitating means to analyze release history, track changes and include the possibility of development performance measures (Breivold et al., 2012).

Participants also state that, after being part of the evaluation process, they realize that the steps from analyzing and developing software are not dissimilar to those applied on many optimization areas. There are academic works mentioning the possibility of generating frameworks to provide a foundation for future developments, such as ERP implementation, industrial management and integrated supply chains, etc. (Stuart et al., 2002), but researches providing specific proposals are scarce.

6 CONCLUSIONS

Advanced Planning Systems (APS) are spreading quickly as systems or modules that can automatize the optimization of production planning problems. However, there is a lack of Software Engineering (SE) research associated to them. APS-RA, a Reference Architecture (RA), has been proposed to characterize the APS domain, and reduce times and costs associated to the *ad-hoc* development of such systems. However, it has not been evaluated, which is required to ensure that it facilitates the Quality Attributes (QA) elicited during analysis.

Evaluating APS-RA has challenges: available methods do not consider particularities of RA, such as a higher level of abstraction and a less defined stakeholder base, and only include participants with Software Engineering (SE) specific knowledge. This is an issue, as APS development requires software developers and process optimization experts to implement both the system and models.

Therefore, this article introduces ATAM-M, an evaluation methodology based on the *Architecture Trade-off Analysis Method* performed in two Stages, each of them centred on different groups of stakeholders. This ensures that participants work within their expertise and focus on aspects of the APS they value the most, while considering specific interests and points of views, shaping the results.

Stage 1 participants are process optimization experts and planners. Steps and outputs are tailored to fit their expertise, avoiding working on specific SE concepts, or analysing architectonic qualities that are the focus of Stage 2. A number of academics and professionals with previous work on the area are invited to take part of the process.

ATAM-M outputs includes novel proposals, but the most relevant are Design Decisions (DDs): structural choices decided when planning and developing a model, that can improve or hinder its interoperability with the APS, and which QA can be applied. DDs obtained during the evaluation present a parallelism with the concept of tactics, representing solutions to similar problems, but applicable to the models within the APS. DDs can also represent risks, sensibilities or trade-offs. Producing these lists can reduce design times and costs, as they provide a foundation analysis on how choices will affect the QA related to the models.

As a result, ATAM-M Stage 1 includes participants with background not related to SE, and obtains a valuable evaluation using the participants' expertise and obtaining rich conclusions.

The application of Stage 1 assessment concludes favourably for the APS-RA, enabling Stage 2, a more traditional architectonical evaluation. Also, during the brainstorming, two requirements are elicited: one is a new FR, while the other is added to existing FR, and only produces description changes.

Finally, participants detect a number of SE concepts and their possible applicability to the development of optimization models, e.g., using version control when working with models, in order to add traceability and improve teamwork.

REFERENCES

Angelov, S., Grefen, P. & Greefhorst, D., 2012. A framework for analysis and design of software reference architectures. Information and Software Technology, 54(4), pp.417-31.

Angelov, S., Trienekens, J.J.M. & Grefen, P., 2008. Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case. In Morrison, R., Balasubramaniam, D. & Falkner, K., eds. Proceedings of the Second European Conference, ECSA 2008. Paphos, Cyprus, 2008. Springer Berlin Heidelberg.

Bachmann, F. et al., 2003. Chapter 9. Documenting Software Architectures. In Bass, L., Clements, P. & Kazman, R. Software Architecture in Practice. 2nd ed. Boston, MA, USA: Addison-Wesley. Ch. 9th.

Bass, L., Clements, P. & Kazman, R., 2013. Chapter 21. Architecture Evaluation. In Software Architecture in Practice. 3rd ed. Pittsburg, USA: Addison-Wesley. pp.397-418.

Bosch, J. et al., 2002. Variability Issues in Software Product Lines. In F. van der Linden, ed. Software Product-Family Engineering. 1st ed. Bilbao, Spain: Springer Berlin Heidelberg. pp.13-21. DOI: 10.1007/3-540-47833-7_3.

Breivold, H.P., Breivold, I. & Breivold, M., 2012. A systematic review of software architecture evolution research. Information and Software Technology, 54(1), pp.16-40. http://dx.doi.org/10.1016/j.infsof.2011.06.002.

Cloutier, R. et al., 2010. The Concept of Reference Architectures. System Engineering, 13(1), pp.14–27.

Diniz, C., Menezes, J. & Gusmão, C., 2015. Proposal of Utility Tree for Health Education Systems Based on Virtual Scenarios: A Case Study of SABER Comunidades. Procedia Computer Science, 64, pp.1010-17.

Dobrica, L. & Niemela, E., 2002. A survey on software architecture analysis methods. IEEE Transactions on Software Engineering, 28(7), pp.638-53.

Fleischmann, B. & Meyr, H., 2003. Planning Hierarchy, Modeling and Advanced Planning Systems. Handbooks in Operations Research and Management Science, 11, pp.455-523.

Fleischmann, B., Meyr, H. & Wagner, M., 2015. Advanced Planning. In H. Stadtler, C. Kilger & H. Meyr, eds. Supply Chain Management and Advanced Planning. Germany: Springer Berlin Heidelberg. pp.71-95.

Framinan, J.M. & Ruiz, R., 2010. Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. European Journal of Operational Research, 205(2), pp.237-46.

Gayialis, S.P. & Tatsiopoulos, I.P., 2004. Design of an IT-driven decision support system for vehicle routing and scheduling. European Journal of Operational Research, 152(2), pp.382-98.

Heikkilä, L. et al., 2011. Analysis of the new architecture proposal for the CMM control system. Fusion Engineering and Design, 86(9-11), pp.2071-74.

Henning, G.P., 2009. Production Scheduling in the Process Industries: Current Trends, Emerging Challenges and Opportunities. Computer Aided Chemical Engineering, 27, pp.23-28.

Ionita, M.T., Hammer, D.K. & Obbink, H., 2002. Scenario-Based Software Architecture Evaluation Methods: An Overview. In Workshop on Methods and Techniques for Software Architecture Review and Assessment at the International Conference on Software Engineering. Orlando, Florida, USA, 2002.

Ionita, M.T., Hammer, D.K. & Obbink, H., 2002. Scenario-Based Software Architecture Evaluation Methods: An Overview. In Workshop on Methods and Techniques for Software Architecture Review and Assessment at the International Conference on Software Engineering. Orlando, Florida, USA., 2002.

Kallestrup, K.B., Lynge, L.H., Akkerman, R. & Oddsdottir, T.A., 2014. Decision support in hierarchical planning systems: The case of procurement planning in oil refining industries. Decision Support Systems, 68, pp.49-63.

Kazman, R., In., H.P. & Chen, H.-M., 2005. From requirements negotiation to software architecture decisions. Information and Software Technology, 47(8), pp.511-20. http://dx.doi.org/10.1016/j.infsof.2004.10.001.

Kazman, R., Klein, M. & Clements, P., 2000. No. CMU/SEI-2000-TR-004 ATAM: Method for Architecture Evaluation. Final Report. Pittsburgh, USA: Carnegie Mellon Software Engineering Institute.

Martínez Fernández, S.J. et al., 2013. A framework for software reference architecture analysis and review. In Experimental Software Engineering Latin American Workshop, ESELAW 2013. Montevideo, Uruguay, 2013.

Northrop, L., 2003. Chapter 2. What Is Software Architecture? In Software Architecture in Practice. 2nd ed. Boston, MA, USA: Addison-Wesley. Ch. 2.

Shanmugapriya, P. & Suresh, R.M., 2012. Software Architecture Evaluation Methods – A survey. International Journal of Computer Applications, 49(16), pp.19-26. http://dx.doi.org/10.5120/7711-1107.

Sharafi, S.M., 2012. SHADD: A scenario-based approach to software architectural defects detection. Advances in Engineering Software, 45(1), pp.341-48.

Shaw, M. & Clements, P., 2006. The Golden Age of Software Architecture. IEEE Software, 23(2), pp.31-39.

Stadtler, H., 2005. Supply chain management and advanced planning—basics, overview and challenges. European Journal of Operational Research, 163(3), pp.575-88.

Stuart, I. et al., 2002. Effective case research in operations management: a process perspective. Journal of Operations Management, 20(5), pp.419-33.

Vidoni, M. & Vecchietti, A., 2015. A systemic approach to define and characterize Advanced Planning Systems. Computers & Industrial Engineering, 90, pp.326-38.

Vidoni, M. & Vecchietti, A., 2016. Towards a Reference Architecture for Advanced Planning Systems. In Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016). Rome, Italy, 2016. SCITEPRESS – Science and Technology Publications, Lda.

Weinreich, R. & Buchgeher, G., 2012. Towards supporting the software architecture life cycle. Journal of Systems and Software, 85(3), pp.546-61.

Wu, X., Murray, A., Storey, M.-A. & Lintern, R., 2004. A reverse engineering approach to support software maintenance: version control knowledge extraction. In Proceedings of the 11th Working Conference on Reverse Engineering. Victoria, BC, Canada, 2004. IEEE.