Architectural Scenario Comparison

Vitor Manuel Grades da Silva
Instituto Superior Técnico (IST), Technical University of Lisbon
{vitor.silva@tagus.ist.utl.pt}

Abstract: The Enterprise Architecture Management is a complex task that involves an analysis of what constitutes an Organization and decision making to ensure its operation and to enable its transformation to meet the challenges and opportunities as they arise. It is fundamental for the decisions to be based on information and that they promote the best possible choices. Each time one of those decisions is made the various alternatives that may exist to develop the Organization should be considered and the decision should be based on the analysis and comparison of those alternatives. This thesis presents an approach to model these alternatives, naming them as Architectural Scenarios and identifying steps to perform an analysis on them within the scope of an Enterprise Architecture, based on specific criteria, so that the choice of the best alternative may be based on complete and useful information.

It is just one step among others seeking to support a process of Enterprise Architecture Planning which is element of investigation by many authors. Although restricted to a concrete problem, some of its concepts may be common to other problems.

Keywords: Architectural Scenario, As-Is, To-Be, Enterprise Architecture Planning, Impact Analysis

1. Introduction

An Enterprise Architecture can be defined, succinctly, as an organized collection of plans and models that describe and represent knowledge of an organization. An organization is a constantly evolving entity, influenced by internal and external factors (1) and is necessary that the Enterprise Architecture can respond to this, not only supporting the change, but also encouraging and guiding it.

The current approach to manage the changes in the Enterprise Architecture is the definition and modeling of states, called as-is and to-be. The as-is state is a representation of the Organization at the time of analysis. This state must contain all relevant information about business processes, data, applications, and other documents needed for that, along with a description of their relationship, this representation can describe the Organization. A to-be state contains a representation of the Organization, in part or in whole, in a more or less near future, a vision of what you want to achieve. Through the analysis of the differences between the current and future states, we can define one or more plans that identify the steps needed in order to transform the Organization.

1.1 Problem Statement

An Enterprise Architecture aims to capture the key elements of an Organization and its relationships to the purposes of documentation, analysis and planning. Since these elements and their relationships change with time, planning and managing the Enterprise Architecture is becoming increasingly complex. (2) The process of transformation of an Organization must identify opportunities or challenges and analyze the possible consequences of change. (3) The analysis of the consequences of this change can be made using scenarios (to-be models) that represent the various possibilities to address the situation, comparing them, so that one can choose the alternative that provides more benefits. Yet those choices are not always based on the best information.

There is a problem in this process of collecting information that concerns the lack of models and methodologies that support an analysis of architectural scenarios, to enable a decision based on information derived from the Enterprise Architecture and comparison of alternatives. Going forward, we have the inefficiency or lack of tools or other forms of automation to cope with this need. Obviously, if there isn’t a model and methodology for analyzing and comparing architectural scenes.
it is not expected the existence of tools to support them, but this type of planning decisions take place in today's organizations, often based on intuition and ad hoc processes. (4)

“... dynamic aspects resulting from changes over time are not addressed well by most current EA management tools. While nearly all tools support as-is and to-be modeling, road mapping, versioning and transformation paths are usually not addressed in a sophisticated manner.”

Stephan Aier, 2009 (5)

In short, despite being recognized that the transformation of an Enterprise Architecture should be implemented through the definition of plans that describe the changes, and these plans should be based on information resulting from the confrontation of different alternatives, the methods, techniques and tools for this analysis have not been targeted for development, have gaps or are still immature.

1.2 Objectives

This thesis aims to explain an approach that allows for a stakeholder to model one or more Architectural Scenarios and analyze and compare them according to predetermined criteria within an Enterprise Architecture.

For that, this approach will try to identify what can be considered the essential aspects that allow the modeling of an Architectural Scenario, including its properties and constituents. Next, it will try to identify a criterion that can be used to analyze an Architectural Scenario, bearing in mind that its purpose is to assist the planning of the transformation of the Organization. Once the foundations of the approach are laid out, this thesis will describe the implementation of a software solution that automates the task of analyzing an Architectural Scenario, and presents its results in a concise, explicit and useful form to a stakeholder.

In short, this paper proposes an approach that meets the following objectives:

- Definition of the concept and model of Architectural Scenario
- Identification of a criterion or metric relevant for the analysis of Architectural Scenarios
- Development of a software solution that allows to analyze an Architectural Scenario
- Presentation of the results of the analysis
- Validation of the proposed approach

2 Basic Concepts

2.1 Enterprise Architecture Planning (6)

The purpose of the Architectural Comparison of architectural scenarios is to assist the process of Enterprise Architecture Planning. There are several models for this planning process proposed in the literature and S. Aier et al (5) sums up the essence of three of them in Figure 1.

![Figure 1 - Enterprise Architecture Planning Process (5)](image)

This thesis does not seek to define or validate a planning process, but focus on one of its stages, common to many planning processes, and that in the present case is the fourth step - "Analyze and Evaluate Alternative To-Be Architectures" stipulating that, for planning the next state of the current as-is
architecture, an to-be architecture needs to be chosen from the existing alternatives. This selection process requires the analysis, evaluation and comparison of the presented options. (5) Is within a planning process like this one proposed by S. Aier et al. that the approach described in this thesis can be applied. The approach described in this document is based in the definition of as-is and to-be architectures identified by Sousa et al. (6)

A to-be architecture in a given time t consists of the set of alive artifacts at that time based on the As-Is state. An artifact is on the pseudo state “alive” if has relations to other artifacts in other states that not "dead." So identifying what makes up a to-be architecture is based on the IT Projects of the Enterprise Architecture and the premise that a project, as a motivational agent of change in the organization, during its development is formed by a set of artifacts that change from state to "alive" or "dead" at specified points in time and accordingly to the project development.

2.2 Architectural Scenario

The key concept of this thesis is the Architectural Scenario. It must contain the relevant information to describe a future state or part of a future state of the Organization, with emphasis on Information Technology. You do not need a scenario that represents the entire state or architecture desired for the future of the Organization, because this future state may depend on the choices between several alternative architectural scenarios, or may not be possible to model in its entirety for lack of information. But it must contain the information deemed necessary to describe the premise associated to the architectural scenario. Drawing a parallel with classical architecture, an architectural scenario is a mockup, modeled to identify aspects that can range from the visual impact of a building to the study of its aerodynamic properties.

An architectural scenario can be modeled, for example, to gather information on the consequences of introducing a new application in the enterprise landscape, so the relevant information should be the characteristics and interactions (repositories, information flows, communication, etc.) that new application will have with the existing applications and maybe the information about the business processes that this new application will support.

Since an Architectural Scenario is intended to aid in the planning of IIT landscape, it has an affinity with the concept of IT Project. IT projects are the agents that contribute most to the change of the IT of the Organization, (7; 6; 8) which means that the Architectural Scenarios can be used to model alternative proposals for projects or parts of projects. An example of two architectural scenarios that model alternative projects is for example, if one of the scenarios describes the introduction of a software solution developed in-house and the other describes the introduction of a commercial software solution. For this two architectural scenarios to be considered as alternative architectural scenarios they just need to be planned for the same period of time in the future.

However it is important to note that a scenario is not exactly a project because a project may have more information than just the relevant for the architectural scenario and obey to rules of planning, while an architectural scenario is focused on the architectural aspects and in terms of planning will be closer to an idea to be examined to determine whether it is feasible, and what is his potential impact on the existing and Enterprise Architecture planning.

So an Architectural Scenario is a description of a state or part of a state of the Enterprise Architecture. Now it becomes necessary to identify specifically what should be modeled on an Architectural Scene.

2.3 CRUD Operations in an Architectural Scenario

As mentioned previously, there is an affinity between the concepts of architectural scenario and IT Project. A project is, by definition, a temporary endeavor, with a beginning and end (typically defined by dates or costs) and carried out to achieve goals and objectives. (10) Furthermore it requires resources to be affected to a project and when it is concluded typically it produced products, reports or other deliverables.

An IT project is based on the same premises and is designed for adding a service, product, report or other artifact to the Enterprise Architecture. It also requires resources to be developed, either physical as programmers or development workstations, or software resources as applications that require access to pre-existing services. If you want that an architectural scenario captures the relevant information of a project or part thereof, then an architectural scenario will have a composition similar to a project.

We can look at the characterization of project and architectural scene from the point of view of the four basic operations in persistent storage, commonly known as CRUD. The end products of a project arise
from operations Create, they are artifacts that are created and are to be incorporated into the Enterprise Architecture. The verification of requirements is a Read operation; it consults the Enterprise Architecture to verify if a given artifact is available and complies with the rules stipulated by the requirement. The Update and Delete operations are present when, during the development of a project artifacts are replaced by new versions (based on the paradigm of versioning), or when artifacts are no longer required to be stored in the Enterprise Architecture or should become disabled or a similar meaning state.

This interpretation is consistent with two references: The TOGAF ADM (11) and the definition of architectural design decision (12).

In the TOGAF ADM, during the phases B: Business Architecture, C: Information Systems Architecture and D: Technology Architecture it’s required to perform a Gap Analysis that essentially compares a base architecture with a target architecture within various scopes, but this comparison is made by analyzing four basic aspects within a gap matrix:
- Identify building blocks to be carried over, classifying as either changed or unchanged
- Identify eliminated building blocks
- Identify new building blocks
- Identify gaps and classify as those that should be developed and those that should be procured

The identification of the added, eliminated and modified building blocks is essentially the listing of Create, Delete and Update operations that are considered fundamental to the comparison.

The Jansen et al. definition of architectural design decision is “A description of the set of architectural additions, subtractions and modifications to the software architecture, the rationale, and the design rules, design constraints and additional requirements that (partially) realize one or more requirements on a given architecture.” Again we can easily identify the Create, Update and Delete operations. The Read operation is present in the form of the requirements and constraints that influence a architecture.

So the listing of artifacts and respective relations according to their CRUD operations is therefore a part of the information that describes an architectural scene, but it is not the only relevant information, because an architectural scenario seeks to capture a state or part of a state for a range of time in the future.

### 2.4 Time in an Architectural Scenario

A key factor in Enterprise Architecture and its management is time. The to-be states are representations of what is intended to be an overview of the architecture in a more or less near future. If an architectural scenario should capture a portion of these states, then it must also be modeled for a future time.

But more than an instant in time, an architectural scenario has an associated range in time, similar to a project, in which their artifacts are added, removed or upgraded and undergo changes of state, i.e., an Architectural Scenario has an associated period of time where it is estimated that the changes that he describes can be applied. This decision, together with the definition that an architectural scenario consists of artifacts, each of which may undergo changes in pseudo state, broadens the definition of alternative scenarios. Two architectural scenarios can be alternative if contain different artifacts at the same point in time, but also if the time deployment of the artifacts they contain are different.

For example if we consider a scenario that considers the replacement of a service interface, and the interface should go out of production in a time $t$, the new version can either go into production in time $t+1$ or $t+2$, as shown in Figure 2. Namely, the fact that an architectural scenario is not confined to a single moment in time allows us to model an internal planning for the artifacts that comprise it.

![Figure 2 - Two alternative architectural scenarios](image_url)
Another date which may be useful to safeguard is the date in which the architectural scenario was modeled, given that when an architectural scenario is specified, the determination of the time period in which it is believed it can be achieved is taken in accordance with the existing plan. Because of this relationship between the time of the modeling of the architectural scenario and the enterprise architecture planning it can be helpful to keep the information of when the modeling was conducted in order to, for example, be able to revisit previously discarded scenarios when a change in planning occurs. Another situation where the date of modeling can be a useful piece of information is within a historic system as advocated by Buckle et al. (7)

In short, the model of an architectural scenario should contain the following information:

- Date on which the scenario is modeled
- Time interval in which the contents of the scenario can be realized
- List of artifacts that are added to the Enterprise Architecture
- List of artifacts that are removed from the Enterprise Architecture
- List of artifacts that are modified in Enterprise Architecture
- List of Enterprise Architecture artifacts whose availability is required

Once identified the constituents of an architectural scenario in the next section will be identified the operations that can be applied to it.

### 2.5 Scheduling Impact Analysis

As previously mentioned, time is an important factor in the Enterprise Architecture Management and architectural scenarios should be prepared to deal with it. This requirement provides, at the same time, an opportunity to analyze the impact of architectural scenes in the rest of Enterprise Architecture. An artifact must have defined the date on which it is planned to be added, modified or removed from the Enterprise Architecture, as well as the identification of the artifacts with which he will have or already have relationships. Thus it is possible to determine the impact on the rest of Enterprise Architecture at the time of those operations. For example by analyzing the impact of an architectural scenario that describes the removal of a service it’s possible to determine if in the planned date remain elements or business processes that may be unable to function properly because they depend on the removed service.

A possible analogy with the classical architecture would be a civil engineer trying to plan the execution of a construction project on which to determine when a roof can be put first he needs to know when the foundations will be ready or during the execution of the project the identification of the moment at which all the conditions necessary to proceed with the wiring are met, such as if the walls are already there.

Scheduling Impact Analysis will identify possible obstacles or complications to the implementation of Architectural Scenarios that are based on the existing planning, or help identify complications that may arise due to the modification of the existing planning, again through Architectural Scenarios. This means that the Scheduling Impact Analysis will respond to essentially two categories of issues. What impact will be caused by the Architectural Scenario in the current planning and what will be the impact of the existing planning in the Architectural Scenario?

To support this analysis it will be necessary to trace the relations between artifacts that comprise the Enterprise Architecture and the artifacts that belong to the architectural scenario and establish an order of dependence or interdependence between them.

The concept of dependence or influence can be derived from the ontological definition of the system, (9) specifically from one of the symbols introduced for its formal definition, the symbol → which means "acts upon". For x and y, x acts upon y if and only if x influences the behavior of y, i.e. y requires that x exists. Moreover if x acts upon y and y acts upon x, they are said to interact, or influence each other's behavior.

In his approach to identify the composition, structure and environment of a system, Sampaio (10) introduces three categories (ISComposition, ISEnvironment and ISStructure). These categories seek to help in identifying the elements that compose each of these properties. The category ISComposition means that for a given relation <x, y>, "x is composed by y" and "y composes x." The category ISEnvironment means that for a relation <x, y>, "x has y in its environment" and "y belongs to the environment of x". The category ISStructure seeks to determine the pairs of elements that make up the structure of the system and means that for a relationship <x, y>, "x and y establish a structural relationship of the system".

The issue of dependence can be added as a layer of detail on this approach, particularly in categories ISEnvironment and ISStructure because a relation of category ISComposition according to its formulation has a relation of influence or dependence, because if "x is composed by y" then "x is influenced by y".

When considering the representation of dependency between elements, the edges of the graph which models the organization are all directed, for the fact that if there is a relation between two elements means...
that there is influence between them, because if that didn’t happen then it would not be relevant to represent this relation.

Figure 3 - Multi-Graph of a System

The above figure represents a multi-graph (11) showing the categorization of relations and the influence of elements. The influence or dependency is an additional layer of detail. The representation in the form of a multi-graph allows you to relate two articles in terms of category and the relation of dependence or interdependence of them. However for the purposes of impact analysis in the planning, it isn’t just necessary to identify if a relation exists between any two articles, it is also necessary to identify whether this relation is valid.

A relation is valid in the range of time in which the two related artifacts are in the pseudo state "alive". For a given relation R between two artifacts A and B:

\[
\text{Validity}(R) = \{\max(DS_A, DE_B), \min(DS_A, DE_B)\}
\]

Where DS denotes the date that the artifact enters the "alive" state and DE the date the artifact passes to the "dead" state.

By applying this analysis we identify the time intervals in which the relations between the artifacts that compose the architectural scenario, and other artifacts belonging to the Enterprise Architecture are valid, i.e., identify the time periods in which all artifacts and respective relations are in the state "alive." But we want to examine if the change of state of an artifact influences others, so we have to take into account the previously described concept of dependence or interdependence. Considering this concept the validity of a relation undergoes a refinement.

For a given relation R between two artifacts A and B, where DS and DE denote the dates of entry and exit of the state "alive":

- If \( A \rightarrow B \) \( \text{Validity}(R) = \{DS_A <= DS_B, DE_A > = DE_B\} \)
- If \( A \leftrightarrow B \) \( \text{Validity}(R) = \{DS_A = DS_B = DE_A = DE_B\} \)

If A acts on B, then B depends on A, so A should be in the "alive" state before B and should only change to state "dead" later, or at the same time that B.

If A and B are interdependent, this implies that A and B must be in the "alive" state at the same time.

With these rules the analysis identifies the set of artifacts that are in the "alive" state and if they have relations of dependence with other artifacts if those artifacts are also on the "alive" state, at a given moment. We stop from taking into account all the possible relations and focus only on those that actually cause impact.

The choice of this criterion of impact analysis on scheduling satisfies the two conditions stipulated for its relevance. First it lets us make judgments of comparison between two architectural scenarios. And secondly this criterion is useful for the planning process of the Organization. For example, if we recover the example of two alternative scenarios in which one describes the introduction of an application developed in-house and the other the introduction of a commercial application, with the impact analysis we can analyze them and identify which of the possible solutions may be introduced with fewer complications in the planning or which may be introduced earlier.
3 Developed Solution

After the definition of the conceptual bases of this thesis, a software solution was implemented and general idea of this prototype is illustrated in Figure 4. The software solution starts by extracting the contents of the architectural repository. This information will be processed in order to identify the existing Enterprise Architecture Planning. The stakeholder then identifies which scenario he wants to analyze. The information gathered from the architectural scene is then analyzed and processed according to the selected criteria, the Impact on Scheduling. Then the software solution produces a set of views in accordance with the existing specifications of viewpoints and information derived from the analysis. After consulting the results, the user will have the chance to generate a detailed report of all findings and restart the process by examining other architectural scenarios.

![Figure 4 - Basic concept of the developed solution](image)

3.1 Detected Inconsistences

The implemented software solution was developed in cooperation with Link Consulting and is formed by a set of software modules that work on top of the Enterprise Architecture Management System (EAMS). EAMS comprises diverse functionalities, some of which described in (6). The solution identifies inconsistencies that have in consideration three aspects:

- The pseudo state of the artifacts
- The dependences or interdependences between the artifacts
- The type of operation in which the artifact is involved

The summary of the seven basic inconsistencies identified are listed in the following table:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CREATE</td>
<td>Check if the artifact to add already exists in the Enterprise Architecture</td>
</tr>
<tr>
<td>2. READ</td>
<td>Check if the artifact exists in the Enterprise Architecture</td>
</tr>
<tr>
<td>3. READ</td>
<td>Verify that in the moment the artifact is requested it is available, i.e. “alive”</td>
</tr>
<tr>
<td>4. UPDATE</td>
<td>Verify that the artifact to be upgraded exists on Enterprise Architecture</td>
</tr>
<tr>
<td>5. UPDATE</td>
<td>Verify if in the time of the upgrade, the relations of the new artifact with existing artifacts remains valid</td>
</tr>
<tr>
<td>6. DELETE</td>
<td>Verify if the artifact to be removed exists in the Enterprise Architecture</td>
</tr>
<tr>
<td>7. DELETE</td>
<td>Check if after removing the artifact remain relations with other artifacts, i.e. if are detected &quot;orphan&quot; artifacts</td>
</tr>
</tbody>
</table>

Table 1 – List of detected inconsistencies

4 Results

The implemented solution permitted to extract and present multiple information about the impact of an architectural scenario on the scheduling, and the following figures represent some of the results obtained.
Figure 5 - Results of the analysis of an architectural scenario

Figure 5 depicts the usage of color codes to represent the detection of inconsistencies. Red means that a potential inconsistency that could prevent the implementation of the architectural scenario was detected.

Figure 6 - Time Interval where no inconsistencies are detected

Through the usage of the top slider to parameterize the interval to which the analysis should be applied, a stakeholder can identify the ranges in time where the architectural scenario doesn’t present any potential inconsistencies.

Figure 7 - Report of the inconsistencies detected

The Figure 7 presents a report taken from the analysis of an architectural scenario depicting an inconsistency resulting from the removal of an article titled "Customer Database" that potentially prevents other artifact entitled "CIMA - Customer Information - DATA" from developing its normal functionality. The approach described in this thesis is not a solver for planning issues, it does not automatically resolve the inconsistencies identified, but allows us to gather useful information to take steps to remove inconsistencies or deal with them.

The architectural scenarios are hypotheses or proposed architectural states, and even though there could be much information to base and justify its development, there are no guarantees that it can be accommodated in the scheduling without causing inconsistencies in the planning. This is due to the complexity of the structures of relations between artifacts and alignment between these structures can be difficult or even impossible without a change of planning. To better cope with these situations one should try to avoid defining architectural scenarios too complex, with multiple operations; otherwise the number of inconsistencies may become too big to cope with all of them.

5 Conclusion
This thesis began with a research on the as-is and to-be architectures and their use within an Enterprise Architecture. From the start it was clear in the literature that the use of as-is and to-be architectures to model the actual state of an Enterprise Architecture and to model future is a practice referred by several authors and used in multiple tools. However a problem existed with regard to drawing up plans for the transformation of the Enterprise Architecture. Though recognized by several authors that the information that should lead to the modeling of these plan should result from comparing alternative to-be architectures or the derivation of intermediate states between the as-is architecture and the to-be architecture that represents the future vision intended for the Organization, the issue is that the information used to prepare this plan of transformation is not always the best or is based on intuition and ad hoc processes. The first step in dealing with this issue was to define what an Architectural Scenario is. Based on the methodology of identification of as-is and to-be architectures established by Sousa (7), this paper defines what I consider to be an Architectural Scenario and how it can be modeled, identifying the information that it should contain. Despite the lack of a more extensive validation of the approach proposed here, the objective was fulfilled:

1. Definition of the concept and model of a scenario Architectural

This was followed by the identification and choice of a criterion that were considered relevant to analyze and compare Architectural Scenarios. The choice fell on the Impact Analysis on Scheduling. This is one possibility among several criteria and metrics. The choice could also rest, for example, on software quality metrics, but once the initial context for identifying the problem was the planning of Enterprise Architecture, the Impact Analysis on Scheduling is directly related with this issue, allowing to identify potential problems or conflicts arising from inclusion in the planning of possible solutions to deal with opportunities or challenges identified in day-to-day operations. However in order to determine the quality or effectiveness of an architectural scenario it is useful to consider more criteria or metrics, to better support the decision. Thus was partially fulfilled the objective:

2. Identification of a criterion or metric relevant to make an analysis of Architectural Scenarios

Once established the conceptual basis i moved to implement a functional prototype. This was based on an Enterprise Architecture tool (EAMS), to which first was added the ability to read an Architectural Scenario and subsequently the functionality of analyzing an architectural scenario. Upholding the objective:

3. Development of a software solution that allows to analyze Architectural Scenarios

Also within the software solution, the presentation of the results of the analysis of Architectural Scenarios was developed based on standards already used by other tools. The pattern of lights is used in many tools and not just in the IT to identify rapidly the state of objects and information, so the trouble began by determining the manner of presenting this solution in a concise and easily interpretable by user. The reporting is also a form of representing the information used in various areas of society, not only within the Enterprise Architecture Management, in agreement with the objective:

4. Presenting the results of the analysis

Finally a series of tests were designed and conducted to verify the quality of the solution, and although the results indicate that this is a useful approach, further tests are needed to ensure the applicability of the solution to the Management and Planning of an Enterprise Architecture that include other steps not covered by this approach. Thus was fulfilled in part the ultimate goal:

5. Validation of the proposed approach and its suitability to real problems

5.1 Lessons Learned

Throughout the development of this thesis various topics and issues related to the problem and the developed solution were identified, which is worth considering. The first aspect concerns the incompleteness of the work. The fundamental purpose of analyzing and comparing Architectural Scenarios is to identify the quality and merit of one or more Architectural Scenarios in order to make the choice between possible alternatives. I think the definition described in this document identifies the bases for this to be possible, but the identification of only one criterion, the Scheduling Impact Analysis, by itself it does not meet this goal. It is necessary to identify and define additional and relevant criteria for determining the quality of an Architectural Scene. And then looking at the definition of Architectural Scenario, it contains a perverse aspect that can influence the quality of the results of an analysis. By stating that an Architectural Scenario should contain the information which is considered relevant to the scenario we run the risk of skewing the results, because what is considered relevant by the proponent of the architectural scenario may not be all the required information. This is a common issue associated with the extraction of information from a repository or database. The quality of information is determined by the quality and accuracy of the data
The determination of influence in the relations between articles is a task that can produce mixed results, often because it is based on an interpretation of the semantic description of a relation. The relations of the type artifact A "is composed by" artifact B, artifact A "uses" artifact B or artifact A "creates" artifact B are sufficiently explicit to determine the existence of an influence and effect, but with relations such as the artifact A "is related to" the artifact B or artifact A "connects to" artifact B can cause different interpretations and a dependency between the artifacts is not easily identifiable because.

Manual definition of architectural scenarios has proven to be a difficult and repetitive task, so it should be automated or at least facilitated with an editing interface, possibly graphical, that allows adding and configuring the artifacts and their relations, while avoiding problems that are caused by manually editing a text file, such as omission or inaccuracy of the data. The implemented solution allows parameterizing the architectural scenario time interval to be analyzed through the user interface, which allowed expanding the amount of information that can be obtained from the Analysis of Impact on the Scheduling. Hence the questions if it may be useful to extend or develop an interface for parameterize other aspects of the architectural scenarios and possible the projects also?

One last thought goes to the information resulting from the Analysis of the Impact on the Scheduling of an Architectural Scenario. Could this information be integrated in another type of report or other artifacts of communication? Since the information generated is based on compliance with rules or not, can these rules be used in conjunction with other information to produce documents to justify the planning decisions, which with the aid of a historic system, could enable a stakeholder to subsequently trace the whole process of decision making of a given. Or could the information derived from the analysis of an architectural scenario be useful to develop solutions to the inconsistencies in an automated manner through, for example, decision trees indicating the “possible” states to resolve a given inconsistency. The latter hypothesis could be a step to evolve the architectural scenario analysis in becoming a solver.

Bibliography
