Visualization and Support of Information Security Architectures on Atlas

Gonçalo Ramos
goncalo.ramos@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2020

Abstract

The protection of enterprise information (being either physical or logical) against unauthorized access/damage (malicious or accidental) is vital for an organization’s health. As such, failure to address this issue often resolves on aggravated long-term consequences such as operationally, legally, financially or damage to the organization’s reputation. This paper addresses the support over the Atlas solution to describe and visualize Enterprise Security Architectures. As such, key concepts related to its security context are identified and how their representation can be performed on an architectural practice is evaluated. The support over the platform is mainly accomplished with the development of a generic matrix view, thus ensuring its reuse regardless of the data model, to facilitate the description of its risk analysis and security controls. The matrix is integrated with a set of features able to support an architectural practice, multiple configurations over its components as well as the impact of the lifecycle over its visualization. The work presented is then evaluated with the presentation of its visualization over different concepts related to the security context.

Keywords: Enterprise Architecture, Enterprise Security Architectures, Information Security, Visualization

1. Introduction

With the growth of collected data by organizations, information systems have taken a major role within corporations due to its inapt ability to support operations, management and decision making from interrelated components that together collect, process, store and disseminate information. As information systems provide a massive support in processing the data generation from corporations’ input to assist in operational management, information security has now become one of the most compelling subjects among organizations as it relates to the rightful management and protection of information while focused on the organization’s processes, systems, services and technology. In order to successfully manage complex organizational systems, corporations rely on the application of Enterprise Architecture practices [1]. These architectures apply principles and practices with the main purpose of representing organizational structures, business processes, information systems and infrastructures deemed necessary to execute their strategies. Enterprise architectures encapsulates principals, methods and models used in the architectural design to represent its organizational structure. More enterprises are taking this approach into practical use to manage change, aligning IT with business, reducing costs, decreasing complexity, improving information quality and manage stakeholders’ concerns [2]. These principles and practices, in an information security context, rely on the guidance provided by already proven methodologies available as frameworks, such as TOGAF-9 to ensure the architecture considers all of information security related risks, and thus give birth to an enterprise security architecture [3][4]. By definition, enterprise security architectures are viewed as an unified framework and reusable services to implement policies, standards, and risk management decisions. This research focus on the identification and development of re-usable artefacts to facilitate the architecture and management applied to information security architectures to be integrated on the Atlas platform. This support is mainly provided in the form of a new visual representation highly demanded among enterprise architects worldwide, and viewed as the core instrument for a successful enterprise security architectural practice. This new artefact should not only be capable of providing an effective support in decision-making and IT planning related to information security, but also enable real-time data manipulation through the implementation of inte-
2. Background

2.1. Enterprise Architecture

As enterprises become increasingly reliable on the data collected from organizational inputs, organizations start to consider that improving their information systems and activities has now become a major priority. To facilitate and guide organizations through business, information, process and technology changes, companies rely on the application of Enterprise Architecture practices to conduct enterprise analysis, design, planning and implementation. This practice is to be aligned with the internal development of the organization and its environment to include its strategic and operational activities. Due to its mutation nature, the architecture should be adapted to change, and so its architectural products (i.e., artefacts) have a temporary status. Meaning, architectures represent a set of elements capable to support all of the organization’s information processes and is mostly used as an instrument to manage the company’s daily operations and future development [1]. An enterprise architecture must provide a clear representation of the relation between the architectural decisions and business goals enforced by the organization. With the shift towards a dynamic enterprise strategy approach, the need to re-structure organizations have increased to improve on its flexibility, scalability and adaptability. To sum up, an Enterprise Architecture is viewed as the representation or detailed plan of an organization, its components, their interrelationships, and the principles governing their design [5].

2.2. Enterprise Security Architecture

An organization’s Enterprise Security Architecture relates to a company’s security design, with the main focus of coping the requirements and risks associated to a particular environment, as well as the specification of what controls are to be applied and where. To successfully implement such sensitive design in detail, the architectural practice requires an already well-established and proven methodology to ensure its completeness and reliability. The definition of the business context is of high importance at the beginning of the architecture’s development as it provides a healthy balance between business drivers and acceptable risks where security policies can be viewed as guidelines. And so, the delivered architecture related to these policies stand as a combination of both processes and technology. This architecture must fit the business context for the enterprise to ensure its security and to provide legal and regulatory compliance [5]. In the information security context, it’s possible to identify a set of principles:

- Integral to Enterprise Security.
- Impacts the entire Organization.
- Risk Management defines Information Security requirements.
- Security accountabilities should be defined and acknowledged.
- Must consider internal and external Stakeholders.
- Requires understanding and commitment.
- Requires continual improvement.

2.3. TOGAF-9

The Open Group Architecture Framework, commonly known by the acronym TOGAF-9, is an architecture framework comprised by an already proven practice methodology along with a set of support tools to facilitate in the development of enterprise architectures to further improve business efficiency. It’s an highly reliable enterprise architecture standard able to ensure the consistency standards of the methods applied and communication among enterprise architecture professionals while aiming to support architecture specialist to avoid getting locked into proprietary methods, apply resources in a more efficient and effective manner, and make a large return on investment. Unlike older frameworks, the TOGAF-9 is much more process-driven, able to present ways to codify architectural patterns. It presents architects with an architecture development method (also known as ADM) to detail the steps needed to successfully develop an architecture with four architectural domains (Business, Data, Application and Technology) [3][4]. While the ADM does not include a security-specific methodology to design its architecture, it contains information regarding what type of activities it may include.

2.4. Lifecycle

An architectural view can represent an organization’s architecture through a graphical perspective representation at any given moment. To manage these views according a time frame its architectural elements are assigned with their respective lifecycle. With a pre-designated color scheme, the artifact can monitor its elements’ lifecycle stage with the assistance of a time slider to visually represent the lifecycle’s evolution in time, and thus enable architects to scan through previous representations (AS-WAS models), actual representation (AS-IS model) or predicted representations (TO-BE models).
2.5. Enterprise Security Architecture Tools
In order to assist an enterprise architectural practice, a wide variety of tools are already available in the market. These tools comprise enterprise modeling techniques developed through enterprise architecture’s frameworks and are capable to support architects, managers and stakeholders in the creation and management of architectural deliverables or products [6]. The integration of these tools provide architects with a wide range of advantages:

- Descriptions of architectural domains stored as part of the same model.
- Increase architectural consistency.
- Automated generation of views and visual representations.
- Teamwork support.
- Description of enterprise architectural principles within the same model.

2.6. Matrix role in Enterprise Architectures
By definition, a matrix stands as a structured graphical representation of information where data is organized in rows and columns. In Enterprise Architecture, matrices are one of the main visual representations to communicate architectural information with stakeholders due to its information-driven properties. Both Relationship Matrices and Gap Matrices are grids able to represent relational data between matrix’s element pairs composed by different object classes and properties at the intersection of its rows and columns [7][8]. It was perceived that among the visual representations available, the matrix view stood as highly demanded in the architecture’s circle as they are able to easily represent object relations and further enhance security architectures when addressing issues like access controls[7]. As these tools are mostly developed as web-based solutions, a simple data virtualization tool does not completely fulfill an architectural practice. Meaning, artefacts developed are to be integrated with visual interactivity support to manipulate the data represented through custom animations and thus increase project management efficacy and efficiency.

3. Solution
As per concluded from an analysis over the present state of the art offers a limited view over the matrix visual tool integration, not capable to provide enough versatility and complexity when considering the full extent of how these views can be represented or manipulated. When contemplating the Atlas’ platform, the available visual representations do not offer enough support for an information security architecture modeling and its inherited challenges. As such, this solution has a major focus on empowering the Atlas’ platform with the conception of a new visual representation, the matrix visualization tool. This matrix covers the topics lacking in the present EA tools, as it’s capable of monitoring the represented element’s Lifecycle as well as support relations with intermediary objects. Moreover, the matrix is integrated with complex features to enable visual interactions and animations by the manipulation of its attributes.

4. Implementation
This chapter presents the key implementation steps in the development of the matrix visual representation, along with the supported features, required not only to produce an optimized visual tool but also capable to empower its architect’s productivity with the assistance of integrated interactions and animations.

![Figure 1: Matrix Representation on Atlas.](image)

It comprises key development considerations regarding the blueprint’s configuration panel adaptation, the deployment of a visual data representation under a matrix layout, how the web-application interactions are integrated and, finally, how the represented data can be manipulated supported by a newly designed customization panel. This web-based solution is developed primarily with the trinity of web development technologies such as HTML5, CSS and JavaScript and it’s available libraries.

4.1. Configurations
The adaptation of the blueprint’s configuration panel is a vital step to successfully deploy the new artefact’s work environment where the data is to be represented in a matrix visual layout, supported by a revisited customization panel, specific to this artefact. During the development phase, an iterative process comprised of the production and test steps, was noted that the initial fully-configured panel did...
not offer enough versatility as a project management tool. As such, the former configuration acquired more and more complexity, giving birth to a wider range of possibilities when considering a matrix visual representation Atlas’ configuration. This artefact evolution resulted in the establishment of three types of initial configurations. Firstly, a fully-configured panel, a partially-configured panel, and lastly, a panel with no configuration requirements.

4.2. Matrix Modes
The artefact conceived integrates two distinct ways of representing a relationships while in a relational matrix context, which are referred as the matrix’s modes.

4.3. Interactions
In the present days a stale visualization tool does not fulfill all of the enterprise architects’ requirements to become an efficient architectural tool. Therefore, the implementation of these artefacts must be packed with dynamic support, capable of data manipulation by interacting directly with the visual representation. To empower the matrix’s efficiency and usability, the visual tool developed also provides assistance to not only represent but also establish or remove relationships in real-time as well as data analysis at a desired position on the matrix. The distinction between the matrix’s modes play a major role on how these animations are integrated, will still considering the least amount of steps required to complete the desired tasks. The trigger of an event deployed from a mouse-click action initiates the process of either relationship establishment or relationship removal at the interacted cell. In an unidirectional relational matrix, already provided with the relational data requirements regarding the relationship direction and property, and considering the traditional one relation per pair of elements, an interaction handler process starts by inspecting the intersection’s content. If no relationship is already defined at the selected position, the handler interprets this action as the establishment of a new relation and not only visual represent the element connection but also updates the object’s property represented as the relationship’s source. As an opposite course of action, if the matrix’s intersection interacted is already represented as an established relationship, the associated event handler recognizes this animation as a removal process, completed by the removal of the relational data and consequent updates over the relationship’s source element and the matrix virtualization. In contrast to the previous events, when in the presence of an intermediary-object relational matrix, interactions get slightly more complex due to its multi-object per cell nature. As this kind of relational matrix allows the definition of multiple relations at the same position, an interactive window is deployed to ensure what course of action is to be taken. The establishment of a new relationship provides the option to store the information within an existing object or to dynamically create a new object related to the matrix intermediary-object class. Meanwhile, the relationship removal process simply presents the option to select which object present at the intersection is to be updated.

4.4. Customization Panel
Another feature capable to empower the matrix data visualization as an enterprise architectural tool is the represented data manipulation support provided by the inclusion of a matrix-specific customization panel. This panel oversees the dynamic variable requirements deemed necessary to successfully deploy the matrix visual layout as well as all its feature-driven capabilities. It offers a wide range of data re-assignment, comprised by options to either fully configure a new relation matrix from an empty canvas or add modifications over current setup, followed by a real-time update to the visual model displayed. This data manipulation focus on axes
and cell information, each embraced by a multi-step process selection which takes in consideration previous actions to filter its enlisted possibilities. The customization panel also addresses the automation of generic tasks through buttons pre-configured to set all relations or to clean the matrix.

4.4.1 Axes Selection

The customization panel encapsulates a simple interface able to re-structure the matrix visual layout by setting new axes related information. This new setup is generated through a three step process. At first, the data class regarding either the vertical or horizontal axis must be selected from the data class list provided by the interface. Upon the class description, architects must decide whether the data to display is to be filtered by either a query or property. This filter decision influences the available options for the last step, and thus presenting either a list of available queries or properties. It should be noted that either of these steps automatically filter the subsequent options accordingly to past decisions, as the queries and properties listed are the ones that consider the initial data class selected.

4.4.2 Cell Selection

The cell selection step is responsible for instantiating the represented cells as objects, powered by visual interactions, and the application of the matrix’s relational methodology as per the description of its representation mode. This process is triggered by the definition of the relationship logic to apply to the current matrix generation. Upon the competition of this decision, architects are presented with its configuration options. If the previous selection considers an intermediary-object relation matrix, architects must define not only which class is to be considered as the relationship’s intermediary-class but also the intermediary-object matrix’s trinity of queries. This trinity is composed by a population query, the queries to add or remove relationships according to the visual interactions. In contrast, the simplistic relational query only requires the definition of the direction to be taken by the relation and the property of it’s source element over which the relationship is to be established. When all the requirements regarding cell’s information are complete, the intended relational representation is displayed.

4.4.3 Set All and Clear All Buttons

Finally, the customization panel designed specifically for the manipulation of the relational matrix’s data also offers automated processes for generic tasks such as establishing one relation per pair of elements or removing all the established relationships represented in the current view while considering the current matrix’s setup configurations. These options transform tasks that could be viewed as tedious into one singular step mechanisms through the implementation of pre-configured buttons.

4.5. Life-cycle

The life-cycle time slider present at the matrix’s user interface provides enterprise architects with means to monitor the evolution over time of all the elements being displayed by the visual representation. The matrix also takes advantage of this feature to monitor the established relationships, that can be visually represented by a change of the cell’s opacity values. If the element identifying a row or column is decommissioned, then the entirety of that row or column is depicted as also being decommissioned. The logic applied to the life-cycle relation representation also differs according to the matrix’s mode displayed. Considering the simple relational matrix, the relation is viewed as decommissioned if one of its parent elements is also decommissioned. In alternative, when representing an intermediary-object relational matrix, its intermediary-object also has a life-cycle of its own. While the previous logic related to its parent elements also applies, the intermediary-objects must also be analyzed. If at the very least one of the objects contained at the cell is alive, then the relationship represented is also viewed as alive.

4.6. Highlights

The highlight panel extends the already implemented highlight-feature available from the mouse-over trigger. While the former event also identifies the parent elements at any matrix position, this panel provides a search mechanism able to identify any desired matrix’s element by simply displaying the chosen element in a more vibrant color.
4.7. Single Row or Column Setting
The automated tasks provided by the customization panel can sometimes be extreme. To reduce the waste of computation resources, the matrix is also capable of setting relational information by either row or column. The interaction with elements that identify the row or column triggers an event to manipulate the relational data. If at least one relationship is established at the desired row or column, the matrix recognizes the interaction as a removal event, clearing all of the relational data information displayed. If no relationships are present, this event establishes one relation per pair of elements.

5. Optimization
In software development, the optimization of the software engineered relates to the ongoing process to slightly modify its system conception to improve its performance and efficiency. The optimization process, through an analysis over the system performed by test-cases, empowers developers with the knowledge required to implement highly reliable tools whilst minimizing the required computation resources. This development step, when related to web-based applications, mainly focus on the system’s execution time and user experience.

5.1. Duplicate Code
Throughout the coding development process, algorithmic challenges can sometimes overlap by requiring similar operations. If a program is not well structured this issue can give birth to duplicate instruction blocks, and thus increase the file’s size affecting the system’s performance. To overcome this issue, challenges should be rendered into smaller problems to enforce code re-utilization.

5.2. Matrix Display
With the development of web-based solution, the variety of display sizes available in the market should also be taken into consideration, as well as the browser’s window re-dimension function. Having into consideration the canvas’ dynamic re-sizing properties, the matrix layout should also inherit these properties to properly display its content. Therefore, the layout implemented through the SVG configuration should take into consideration the current canvas’ measurements to dynamically exploit the matrix visual representation.

5.3. Data Storage
To extend the matrix to monitor elements represented for the life-cycle and highlight features, these elements must be a part of the canvas data structure to be caught by the listener implemented. For this purpose, two scenarios must be taken into consideration: elements being displayed once the matrix is generated and the elements that are incremented to the representation through visual interactions. In either case, when the element’s information is inspected, its object must be added to the referred data structure if, and only if, it’s not already referenced.

5.4. Synchronization
The constant communication between the client (i.e., web browse) and the remote server requires the establishment of a synchronization protocol as this communication requires the synchronous code blocks from the JavaScript program to wait for the asynchronous server’s response to avoid unwanted system behaviour. To fulfill this protocol requirements, the developed artefact makes use of JavaScript’s Promise object, capable to represent either the completion or failure of an asynchronous instruction and its resultant value. Promises can be defined as a proxy for the values that may not be known upon its creation and are mainly used to handle the result of asynchronous operations. It simulates the return of values as in synchronous operations. With the introduction of Promises to the system, the execution of new synchronous code blocks can be postponed until the asynchronous request is completed.

5.5. Execution Time
The way the system implements algorithms and data structures are crucial to its overall performance. To corroborate for the system’s performance, algorithms should be either constant $O(1)$, logarithmic $O(\log n)$, or log-linear $O(n \log n)$, as the quadratic complexity algorithms $O(n^2)$ usually fail when scalable due to high amount of computation resources required. This high resource consumption is mostly derivative from the high concentration of nested-loops required by this artefact. To mitigate the execution time required inherited by the data manipulation that can applied to re-structure the matrix’s visual representation, a data structure was integrated to store the matrix’s current data. This method avoids the re-computation of all of the matrix’s components when a single data alteration is needed. Moreover, object’s data updates from the establishment of new element’s relationships can also be generated by the application of update queries hard-coded, and thus preventing the extended use of loops while searching for the element’s property when in the presence of a simple relational matrix.

6. Documentation
One of the many factors able to contribute for the success of a software product is its documentation. The software documentation can be defined as an artefact with the main purpose of passing down information related to the system’s application with
the incorporation of multiple case scenarios. This
document represents a written description of the
software systems and thus holds an official status
or authority than can be used as evidence or sys-
tem validation. Not only the reported documenta-
tion increases the confidence levels around the end-
product’s deliverable but also ensures the product’s
success as it can educate architects on the artefact’s
configuration and major concepts, as well as the
description of all features available. Therefore, a
successful documentation relies on the information
provided as a mechanism to minimize the developed
tool’s learning curve through a series of scenarios
described.

7. Conclusions
This solution focus on the support of the Atlas’
platform through newly developed visual representa-
tions to further assist architects in an informa-
tion security architectural model context. After an
extensive analysis regarding the evolution of En-
terprise Architectures and how these can be con-
structed to represent information security architect-
tural models, it was identified a lack of the right
tools to fully exploit the major areas of concern re-
tation to information security issues. Moreover, the
study over the already-marketed tools also revealed
limitations in their solutions, as they don’t empower
the matrix’s visual tools with the faculty to monitor
how the life-cycle can affect the relationships over
time (by inspecting TO-BE models). Also, the re-
tional representations observed cannot cope with
more than one relationship per pair of relational
elements that can be solved with the implementa-
tion of an intermediary-object relational matrix’s
concept. As such, the visual support for the Atlas’s
platform takes into consideration the exploitation of
these particular issues while addressing the develop-
ment of one of the most requested visual representa-
tions by enterprise architects. The matrix view also
packed with visual interactions capable of manip-
ulate the data represented, aiming to increase the
artefact’s versatility as well as architect’s productiv-
ity. For future work, considerations regarding the
artefact’s limitations should be taken. The initial
configuration step, based on a blueprint’s settings
adaptation, is only capable to deploy an intermediary-
object relational matrix due to the lack of configu-
ration options. One possible solution to address this
issue would be the development of new container,
matrix-specific, to serve as the project’s root con-
tainer. This new container would then present an
interface similar to the customization panel’s cell
selection process to address the relational setup ac-
ording to the representation mode intended. As far
as the relationships are visually represented, they
could be further enhanced to overlap the cell’s rel-
ationship representation with either a symbol to rep-
resent the relationship direction (in case of a sim-
plistic relational matrix) or to visually inform how
many relations are present at a matrix’s intersection
when in the presence of an intermediary-object re-
tional matrix. Other feature to consider is the ex-
port functionality, as a formatted JPEG, to improve
the capacity to share information among architects
and stakeholders. Lastly, in order to prevent con-
fusion when in the presence of large data sets, rows
and columns could be locked to the canvas in order
to fully exploit the browser’s scroll functionalities.

All in all, despite the evidenced limitations, the
artefact developed successfully addresses the vi-
ual representation of a relational matrix, inte-
grated with powerful features to allow real-time in-
teractions and relational data manipulation, which
proves to be a major step up to the Atlas platform
when modeling enterprise architectures to address
information security issues.

Acknowledgements
The author would like to thank Professor Pedro
Manuel Moreira Vaz Antunes de Sousa for all the
knowledge shared and assistance provided in the
making of this master thesis, along with a special
dthanks to all of Linkconsulting’s personnel the sup-
port provided during this journey.

References
work: Modelling, communication and analysis.
chitecture security architecture development
technology. Master’s thesis, Lund University,
Lund, Sweden, 2011.
[3] A. Movaseghli L. Ertaul and S. Kumar. Enter-
prise security planning with togaf-9.
http://pubs.opengroup.org/architecture/togaf9-
doc/arch/index.html, last accessed 2019/05/27.
Website.
[5] Sift Pty Limited. Secure your information: In-
formation security principles for enterprise ar-
http://archive.opengroup.org/public/member/proceedings/q209/
q209a/Presentations/blom1.pdf. Presentation.
[7] Zec M. Roth, S. and F Matthes. Enterprise ar-
chitecture visualization tool survey 2014. Tech-
nical report, Technische Universität München,
2014.