Architecture Definition and Evaluation

Technical Evaluation Report

Dr. Malcolm R. Vant
Ottawa, Ontario K2A2L7
CANADA

ABSTRACT

The CSO-IST-115 symposium on Architecture Definition and Evaluation was held in Toulouse, France May 13-14, 2013. The symposium addressed several key areas in the use and development of Architectural Frameworks such as the NAF (NATO Architectural Framework) and its associated standards DODAF (Department of Defence Architectural Framework), MODAF (Ministry of Defence Architectural Framework) and others. Standard Architectural Frameworks were first introduced by the United States in an effort to cut the costs involved in specifying and then building complex military systems. Other nations, and NATO, quickly saw the benefits and followed suit. Any of these frameworks provide a common set of viewpoints and way of describing systems of systems. Although they are similar, the frameworks are not the same, and in some cases their underpinning meta-models differ. Differences among them cause difficulties when assembling multinational Command Support Systems, such as the Afghanistan Mission Network. Furthermore, there is no specified methodology associated with the frameworks, and therefore there can be a steep learning curve when adopting them since each developer tends to develop their own methodology and adopt their own toolsets. This diversity of approach and lack of specified methods leads to a lack of interoperability among developers and a reduction in possible productivity. Other issues exist such as the difficulty in dealing with real-time or dynamic situations in some of the frameworks. The symposium covered various aspects of the use of architecture frameworks such as lessons learned, model-based approaches to development, methodologies for executable architecture, dealing with dynamics, re-engineering legacy systems and cloud architectures. During the Symposium, a very strong message came through that a common methodology is sorely needed and that a true single unified architecture framework would be very useful to all the nations. Many other positive lessons learned and successful methods were also discussed.

KEYWORDS: Architecture Framework, NAF, MODAF, MODEM, DODAF, Cloud Architecture, meta-models, model-driven software development, Enterprise Architectures, systems of systems, tools for architecture development

1.0 INTRODUCTION

The symposium addressed several key areas in the use and development of Architectural Frameworks such as the NAF (NATO Architectural Framework) and its associated standards DODAF (Department of Defence Architectural Framework), MODAF (Ministry of Defence Architectural Framework) and others. Standard Architectural Frameworks were first introduced by the United States in an effort to cut the costs involved in specifying and then building complex military systems. Other nations and NATO, quickly saw the benefits and followed suit. Any of these frameworks provide a common set of viewpoints and way of describing systems of systems. As noted in the symposium program, the frameworks are seen "*as an enabler for managing complexity by providing a logical, standardized way to present and integrate models of systems and as a key enabler for the specification and implementation of interoperability between systems.*" They are *"primarily used to define the operational context, the system architecture and the supporting standards and artefacts that are necessary to document the enterprise, solution or system(s)."* Although these various frameworks are similar, they are not the same and in some cases their underpinning meta-models differ. Differences among them cause difficulties when assembling multinational Command Support Systems, such as the Afghanistan Mission Network. Furthermore, there is no specified methodology associated with the frameworks, resulting in a steep learning curve when adopting them since each developer tends to develop their own methodology and adopt their own toolsets. This diversity of approach and lack of specified methods leads to a lack of interoperability among developers and a reduction in possible productivity. Other issues exist such as the difficulty in dealing with real-time or dynamic situations in some of the frameworks. There is a lot of work currently being done in both the civil and military sectors to grapple with these concerns. Given this background, the goal of the symposium was *"to bring together this growing expertise to achieve a consensus on the most effective approaches in order to support systems integration in the future."*

With these issues in mind, the symposium technical program committee asked for papers on the following topics, which were fairly well covered:

Lessons learned from Architecture development

- Methodologies that enable executable architecture
- Model-based approaches that enable executable architecture
- Tools for Architecting
- Architecting with an Enterprise scope
- Formalisms and languages for Enterprise Architectures
- Methods and technologies for architecture evaluation
- Application and practices of Enterprise Architecture modelling
- Types of Architecture
- Roles of Architects
- Architectural patterns
- Architecture and System Engineering
- Semantic foundation for enterprise architecture
- How relevant is architecture in a Cloud environment?

From the abstracts received, 25 papers were selected, and of those, 22 were presented at the meeting. A presentation entitled "NAF, MODAF and MODEM" by Ian Bailey was substituted for the original paper 12, and papers 1 and 8 were withdrawn. In addition, one Keynote address was given. The papers were organized in the following sessions: Introduction and Keynote, Framework and Modelling Techniques Parts 1 & 2, System Architecting Parts 1 &2, Model-driven Approach, Software Architecting, and Enterprise Architecture Parts 1&2.

2.0 VIEWS AND OBSERVATIONS

2.1 General

The venue for the symposium was very good; no major issues were raised by the participants. Overall, the event was well organized and ran smoothly.

The event had 75 registrants, which is lower than most IST Panel events. Two possible reasons for this are the current severe spending and travel restraints in some countries and a competing event on NEC (Network Enabled Capability) Architectures being held by the SCI Panel. Participants came from 17 nations and at least two NATO organizations. From these 75 registrants, 22 completed surveys were returned, which made it possible to get some good statistics on the audience's impression of the symposium. On the question of the "Overall Quality of the Event", the average score in the surveys was between "excellent" and "very good" with 9 votes being given to each score. For "Overall Value to My Organization" the score was again high with an average of 81-90.

Overall, the quality of the papers was good. At least two-thirds of the papers presented substantial results. The remaining third tended to be superficial in their treatment of the topic, or presented early work with no results yet. One indicator of the paper quality was the difficulty faced in choosing the Best Paper Award recipient. In the panel of those voting for the best paper, at least 12 of the 22 papers were mentioned at least once, and several were very close in terms of being the best paper. In addition, the surveys indicated the majority (over half) of the papers were of high quality.

2.2 Keynote

The single Keynote Address was given by Patrick Chanezon, previously the Senior Director of Developer Relations at WMware and now at Microsoft. This choice of Keynote Speaker was very appropriate. The speaker was very well qualified and very interesting to listen to. He is working on the cutting edge of cloud computing at Microsoft now, but previously held posts at Google, and VMware. He exposed the trends in the industry in terms of dealing with large data and computing problems. He outlined the trends in cloud architectures in moving from software as a service, to platforms as a service, and now the emerging software development as a service. If software development as a service delivers on its promise there could finally be some large gains in software development productivity, given that developers will be able to focus on adding value, and not building the software infrastructure, which will be developed automatically.

2.3 Summary of Papers

As mentioned above, the symposium sessions were organized in the following way: Framework and Modelling Techniques, System Architecting, Model Driven Approach, Software Architecting, and Enterprise Architecting. For the purposes of this evaluation, the following alternative way to look at the issues was adopted to more clearly identify them:

- Practical Use of the Architecture Standards
  - Standard Methodology needed
  - Differences among them (NAF, DODAF, MODAF, etc.)
  - Pitfalls and Lessons Learned
  - Focus on Early Design
  - Collaboration

- Temporal Aspects
- Vulnerability and Security Methodology
- Model-Driven Approaches and Compliance
- Applications to Systems of Systems and Enterprises

2.3.1 Practical Use of the Architecture Standards

Several papers in the symposium dealt with issues involved in using the standards. Paper 20, selected as the Best Paper of the symposium seemed to resonate with many people. It described the experience of Denmark over the past number of years in introducing architecture methods based on NAF. Their frank assessment contained many lessons learned, such as:

- There is an initial high learning curve.
- Do not underestimate cultural barriers with stakeholders - they do not speak NAF. They want to see how their requirements are met, more than to look at NAF views, so translation is required.
- There is no associated methodology with NAF, so the learning curve is steep. Each user has to find and choose their own methods and tools. A standard methodology should be established for NAF.
- Interoperability among NAF implementations is not assured.
- Implementation of the NAF meta-model by different toolsets is inconsistent.
- The NAF meta-model provides a good way of ensuring overall coherency of the design.
- It is not always easy to identify the "right" stakeholders. Frequently, the stakeholders defer to their IT people and do not take ownership of the design.
- There should be a forum where a collective understanding of the meta-model can be refined.
- There should be a way to easily exchange architecture descriptions among nations in NATO.
- It would be useful to have a mapping from the civil TOGAF (The Open Group Architectural Framework) to NAF.

Paper 2, from the French standards group AFNOR(Association Française de Normalisation), discussed the need for the methodologies required for the design of systems with the NAF standard to be made part of a new proposed standard. They quote from the TOGAF standard definition of an architecture, "An architecture should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks.” In their words, “no Architecture Framework is currently fully compliant with this definition." As a first step, the authors called for support of a new standard that embodies an agreed ontology containing the definitions and concepts supporting architectures, not only during their development, but also over their life-cycle. They asked for support from other nations at the ISO (International Standards Organization) to adopt such a standard.

Presentation 12 was offered by Ian Bailey, one of the session chairs, to replace the original paper 12. There was no accompanying paper. In the presentation, the author traced the evolution of the MODAF, DODAF, DNDAF ([Canadian] Department of National Defence Architecture Framework) and NAF standards. He pointed out that the various standards are similar, but not the same. In particular, their underpinning meta-models are different. In the UK, the transition to NAF v4 has been mandated with the proviso that NAF v4 adopt the MODEM (MODAF Ontological Data Exchange Mechanism) meta-model currently in MODAF and fix the sparse and somewhat confusing documentation for NAF. At that point, MODAF would cease to be used by the UK. This transition work is being done under the auspices of the NATO Arch CAT (Architecture Capability Assessment Team). This is a very promising development; however the US DOD's standard DODAF 2.x will not align with NAF v4, but rather a new proposed UAF v1.0 (Unified Architectural Framework), which is further along the timeline for convergence of the standards. Similarly, the Canadian DNDAF would also align with UAF 1.0. Hopefully, this evolution will occur as envisioned and UAF will emerge as the logical replacement for NAF v4 and result in a true convergence of the standards.

At least two of the papers (7, 9) emphasized the need to focus on the early design and get it correct before investing heavily in the rest of the architectural design process. Paper 7 pointed out the pitfall that more and more acquisition teams are experiencing by specifying framework-oriented views or NAF views, rather than a set of requirements focused on the primary business expectations or objectives. The authors proposed an automated approach that realigns the design tasks with the project's objectives and results in a product aligned with NAF Architectural View 1. The approach has been used successfully on several projects within their company. The approach ensures that the business objectives are primary and that compliance with a framework is deduced by the design process and not defined as the starting point.

Paper 9 indicated that given the significant cost involved in developing complex systems, it pays dividends to do as much early engineering as possible to define and evaluate candidate architectures. This paper proposed an early engineering method that used a set of well-known, relatively inexpensive and easy to use, tools and techniques to simply do rapid design iterations of the candidate architecture. The novel contributions in the paper consisted of suggesting a tool set "fit for context", and in developing a workflow to use with them. The paper was more of a think piece though, since the method had not yet been applied.

As noted in the above lessons learned, interoperability among development tools remains an issue for the community. In Paper 4, a method was proposed for overcoming some of the lack of interoperability that exists among vendors' tool sets. A portal is created that provides for the data exchange of Enterprise Architecture or UML-based models and diagrams among developers using different tool sets. The common data model embedded in the portal enables collaboration and sharing among the users of the portal. During the question period, a member of the audience asked how it is possible to maintain the translation among toolsets as they each continually migrate to new releases. The authors explained that one of the main business lines of their company is translation tools, which makes it feasible to deal with the problem of continual upgrade. The project is still under development.

2.3.2 Temporal Aspects

Several papers dealt with how to handle temporal issues in the architectural frameworks. These ranged from life-cycle management, modelling real-time processes, and self-repair of software.

Paper 3 discussed the temporal aspects that have been introduced into later releases of MODAF (via MODEM) and DoDAF 2.0. MODEM (MODAF Ontological Data Exchange Mechanism), the meta-model for MODAF, introduced temporal aspects to architecture modelling. The term temporal in the context of this paper means all of the changes during the lifecycle, the changes for the mission and the changes due to maintenance. The ontological concepts for MODEM come from the exchange foundation objects in IDEAS (International Defence Enterprise Architecture Specification), which provide the detail necessary to express temporal concepts in precise and testable ways. The authors discussed the various components of MODAF, NAF and DODAF and how they deal with time. They also explained the ongoing evolution of the UPDM (Unified Profile for DoDAF and MODAF). UPDM is a standardized means of describing architectures using UML (Unified Modelling Language)-based tools, as well as a standard for interchange. It allows one to describe the architectures in a far more usable format.

Paper 5 examined how to create a framework for real-time system design. The authors emphasized that decisions made early in the design process can have a tremendous impact on non-functional properties such as performance, cost and safety. They proposed creating a framework for real-time design based on a combination of the SysML and MARTE (Modelling and Analysis of Real-Time Embedded systems) languages. This framework is used to model the functional and non-functional system properties. Evolutionary algorithms are used as automated guidance for an iterative optimization process. Using automated optimization at the early stages is important since it avoids many manual iterations of the architecture, which would be extremely costly.

Continuing on the theme of real-time systems, Paper 6 presented a methodology for designing time-critical systems using the Schedulability Analysis Modelling package of the MARTE profile in UML 2.0. A case study was presented for a representative avionics application. The complexity of modelling even simple real-time tasks was illustrated.

In a slightly different view of temporal aspects, Paper 17 illustrated the role software architecture can play in autonomic software systems (systems that manage themselves). During the execution of the system, the software-based autonomic manager maintains explicit traceable links between the architecture currently executing and the original design architecture. Much like commanders use the OODA (Observe, Orient, Decide and Act) Loop in Command and Control, the autonomic manager senses its environment, in this case the executing code, and decides what actions to take based on a logic model. According to the authors, their novel contribution is the creation of a more precise mapping between the run-time and design-time architectures. The paper remains at a very high level, so presumably the work is still in the early stages.

2.3.3 Vulnerability and Security Methodology

The smooth integration of security and vulnerability into the design process of complex systems remains problematic. Three papers discussed various approaches to the problem.

First, the development of high assurance systems was discussed in Paper 10. The authors from NCIA (NATO Communications and Information Agency) found that the scope of NAFv3 is too broad and generic and not suitable for the detailed definition of the high-level design process for security-critical systems, such as a high-assurance guard in an information system. They found that NAF is adequate for creating an initial high-level architecture, but to do the detailed work a different approach is needed. They maintained that the use of the NAF alone in this type of system design introduces too high a design burden, and furthermore the lack of well-defined methodologies for the application of NAF creates a barrier to its use. The authors proposed a lighter weight design process based on a combination of international commercial industry standards, which includes a structured way of collecting requirements, and the use of the CORAS risk-analysis tool. They detailed the process to arrive at a Common Criteria Protection Profile and a System Requirement Specification. The authors view the methodology as a cost-effective means to complement the use of NAFv3 for the design of security-critical systems for NATO. The group has asked IST 114 RTG 57 “Trusted Information Sharing for Partnerships” to review and endorse their approach for security assessment.

In a similar vein, Paper 15 proposed an approach for integrating the consideration of security requirements and assurance into the system design process right at the beginning – something that is not usually done at present. They propose the use of a tool (the SPT (SecFutur Process Tool)) that supports the definition of domain-specific security knowledge, and the creation of DSM and CSM (Domain- and Core-Security Meta-models) implemented using the UML (Unified Modelling Language). These models drive the consideration of the security requirements in the system model. The consideration of security is interwoven naturally into the initial architecture and system description by means of the SPT. As part of the Security Engineering Process, the designer, using the SPT, must assign security requirements to the elements of the system model. Security properties in the form of Security Building Blocks and Security Patterns are assigned to the elements so that they meet the requirements. Security Patterns are products or services designed to meet a specific security need. An XML signature could be a Security Pattern for example. Security Building Blocks, on the other hand, are component level elements assigned to the Patterns. The authors are working on a way to integrate their approach into the NAF methodology. The main issue is the creation of equivalent structures in the NAF meta-model to serve the function of the Core Security Model used in the authors' approach. The efforts described are part of the European Union SecFutur project. A question arose during the discussion on how flexible the security policies are when they are built into the patterns. Can they be easily modified for use where security policies are dynamic, such as in Policy-based Management approaches in networks? The question was left unresolved, at least during the public discussion.

Paper 16 described a Software Risk Analysis Tool that can be used during the design phase to develop an architecture that will likely have fewer exploitable vulnerabilities. The SARA (Software Architecture Risk

Analysis) methodology, which was derived from the NIST 800-30 standard, is coherent with current practices. It allows the analyst to structure their thinking and the application of their expert knowledge by means of rapid, iterative design assessments. SARA has been applied to the analysis of various system components aboard CF (Canadian Forces) platforms. The tool found vulnerabilities in both the technology and procedures employed with the technology. An example of the process was provided for a software component used in data sharing onboard a CF aircraft.

2.3.4 Model-Driven Approaches and Compliance

In Paper 13, the authors described the early stages of the French MIMOSA project, which is concerned with the use of IMA (Integrated Modular Architectures) for civil and military avionics systems. The aim of the project is to provide a capability to precisely capture the requirements for embedded modular software architectures and to provide an acceptable means of demonstrating that the architecture is compliant with these requirements. The ongoing work addresses the enrichment of the architecture model to handle safety and real-time requirements, and the definition of an argumentation model that can be used for the certification of compliance.

Paper 18 generated a fair amount of interest. The author challenged the usefulness of model-driven development in situations where rigorous, formal semantics have not been used in the modelling notation. According to him, without this rigour, tools cannot be built to automatically predict or analyze the expected system behaviour, based on the architectural description. The author proposed the use of the SOL (Secure Operating Language), which is a verifiable language, meaning that programs built in it can be analyzed mathematically and shown to be compliant with the design. The paper proposed extensions to SOL to deal with attributes such as fault tolerance, security, and real-time requirements of applications. It also described the development of methods and tools for formal reasoning about the “artifacts” extracted from the architectural framework. The objective is to be able to eventually compile verifiable code using SOL. At this time, SOL is still a work in progress.

A considerable amount of interest was also generated by Paper 19 which dealt with “The Elephant in the Room – Modernizing Legacy Software and Architectures.” The authors have developed a process for modernizing legacy code, using ADM (Architecture Driven Modernization). ADM is a relatively new type of system modelling specifically designed to enable the effective use of MBSE (Model-Based Systems Engineering) in the modernization process. Normally, legacy software does not have architecture models and detailed system designs associated with it, and therefore MBSE is typically of little use. It needs that type of documentation as a starting point. The ADM process solves this problem by deriving the needed information using a bottom-up formal analysis of the existing code. The analysis creates models from which the executable architectures can be constructed. To enable this process, "The Software Revolution Inc" company has developed a proprietary tool called JANUS, an ADM tool suite, comprising a parser-generator and a code pattern recognition and inference engine. The paper described the use of JANUS to transform legacy code. The authors stated that using JANUS costs one tenth of what a manual process costs to re-engineer legacy code and that it has been successfully used on a large number of US DOD projects.

2.3.5 Applications to Systems of Systems and Enterprises

Most of the papers in this section of the report come from the sessions on Enterprise Architecture. In a way they do consider architectures of enterprises in a general sense, but most are applications or considerations of architectural approaches to building systems of various sorts.

Paper 11 was a bit of a departure from most of the others in that it described the use of a private cloud computing architecture to store all-source maritime surveillance information shared by many government agencies, each with different mandates and roles. The cloud architecture is the basis of interoperability among the agencies. A significant portion of the paper dealt with the formal concept analysis approach to anomaly detection,which was used to do the logical reasoning in the surveillance system's decision support tool, and not the architecture design itself. During the question period, the authors indicated that they are looking at methods to protect information so that only information that can be legally shared among the agencies is shared. They are considering a public-private cloud architecture to separate some types of data. They are also negotiating protocols between pairs of partners for data exchange.

In Paper 14, the authors proposed a process for integrating enhancements into an existing tactical C2IS (Command and Control Information System) used by the German military. The enhancement takes the form of new interfaces and middleware that support unified access to the various communications technologies in the C2IS. The paper illustrated how the middleware model was used to guide the transformation process from the original system architecture to the enhanced one. The enhanced architecture was then used to develop the implementation model, which in turn was used to create an actual implementation in the laboratory.

Paper 21 proposed an approach for creating an intelligent discovery service for use in finding relevant, reusable system components in architecture repositories. The scope of the problem was described and the components defined, but the work appears to still be in the early stages and nothing has been demonstrated yet.

An interesting twist on the use of architecture frameworks was presented in Paper 22. This work described how MODAF was applied to an intelligence problem of modelling blue force countering red force in a Counter -Integrated Air Defence System scenario. Two simple extensions to the MODAF meta-model were created: Capability Addresses Threat, and Service Counters Service. The architecture was modelled as a set of capabilities with services being used to implement effects. The approach provided an interesting logical model for understanding the various interactions possible.

Paper 23 described a three-year effort, in support of the EU, to model the provision of geospatial services required by the security forces to enable them to carry out various missions. NAF v3 was used in the modelling process. The authors indicated that lessons learned in this effort might provide hints for the improvement of NAF, the creation of the requirements for affordable architecture evaluation tools, and guidelines for the methodology. Again, there is a call for better tools and consistent methodology.

The contribution in Paper 24 examined how important it is, at the early design stage, to have a simple means for the evaluation of candidate sensor systems' effectiveness in achieving the mission goals. The paper concentrated on techniques suitable at the macroscopic level of abstraction for weighing system design alternatives. It was a bit of a departure from most of the other papers that were concentrating on the use of NAF-like architectures.

In Paper 25, the actual openness in terms of open architecture systems was evaluated in this case study of a French naval combat direction system. Methods of characterizing openness, as well as an openness assessment and qualification process, were described. The study indicated that there was an insufficient standardization effort from an open systems architecture point of view. Standardization has mostly concentrated on the technical level, rather than at the system level. Most systems remain quite closed.

3.0 KEY MESSAGES

In analyzing the content of the papers, as outlined above, and taking into account the discussions at the event, several key messages emerged from the symposium. The following list provides a quick summary for the reader:

- MODAF, DODAF, DNDAF, NAF, etc. are architectural frameworks, but lack specific standardized methodology. Each major user chooses their own methodology and toolsets to create the architecture, which leads to further interoperability and collaboration issues, not to mention a steep learning curve. A common methodology needs to be developed.
- Although MODAF, DODAF, NAF, DNDAF are all supposed to be interoperable, they are only so to a certain level. In some cases their underpinnings are different. For example, the MODEM meta-model in MODAF differs from that in the others. In some cases, there are also differences in the viewpoints associated with the frameworks.
- Luckily, there is a move towards convergence to a UAF (Unified Architectural Framework). In the interim, MODAF is likely to be subsumed by NAF v 4.0, provided that the documentation for NAF is improved and the MODEM meta-model is adopted.
- Security risk-analysis methods are not built into NAF; at present independent tool sets are used.
- Extensions are needed to deal with time and dynamic situations. These are coming in NAF v4, and are in the MODEM meta-model in MODAF now.
- A NATO hosted collaborative forum or blog would be very useful for developers grappling with some of the issues involved in using NAF.
- The community needs to overcome cultural barriers and have the discussion on architectural models with the stakeholders themselves, instead of with their Information Technology staff. Perhaps, better means of visualization or translation of the architectural views are required.
- The number of views and sub-views in all the formal architectural frameworks is quite high and since the cost to build the architecture model is directly linked to the number of views, it is necessary to have some suitable ways to choose which views to cost into the project.
- Related to the above point, it is useful to have a toolset to use in the early engineering stage to translate the project objectives into the modelling objectives. Good visualization tools would help to make the results of the iterative stages required in early design more meaningful to the stakeholders. It would be too expensive to manually generate a lot a NAF views during this stage of the design. Some have tried doing automatic view generation with early success.
- An interesting approach was presented for re-engineering legacy code using model-based approaches.

3.0 CONCLUSIONS AND RECOMMENDATIONS

One main message from the symposium was that collaboration on standard methodologies and toolsets for use in developing system architectures is needed. The military standards do not have a standard methodology associated with them, so there is a steep learning curve when using the standards. There is also a lack of interoperability during development, since everyone uses their own approach at present. This was one of the assumptions in the Call for Papers, and it was certainly shown to be true. During the presentations there was a valuable exchange of views on lessons learned and what tools are being used, where and why. Several suggestions for the next steps to deal with the methodology question, as well as other issues, were provided by the speakers. These could be grouped roughly into two main clusters: issues where groups want collaboration or help, and suggestions for new activities in the IST Panel.

Following is a list of the issues where groups want collaboration or help:

- AFNOR (Association Française de Normalisation) Architecture Framework Working Group would like help from other NATO nations at the ISO (International Standards Organization) to support a new standard, which creates an agreed ontology containing the definitions and concepts to describe architectures, not only during their development, but also over their life-cycle (Paper 2).

- NCIA (NATO Communications and Information Agency) asked IST Panel IST-114 RTG-57 “Trusted Information Sharing for Partnerships” to review and endorse their approach for security assessment (Paper 10).
- The Danish authors of Paper 20 suggested that there be a NATO-hosted online forum on the use of Architectural Frameworks.
- The authors of Paper 20 also suggested a mapping between the civil TOGAF (The Open Group Architectural Framework) and NAF be developed. This could be a useful role for the IST Panel.
- During the discussion, the NATO Arch CAT (Architecture Capability Assessment Team) member from Canada suggested Paper 20 on lessons learned using Architecture Frameworks be presented at the Arch CAT.

In terms of new activities:

- Clearly the IST Panel ET 71 “NATO Method for Architecture Definition and Evaluation in-line with NAF” is sorely needed. This ET addresses the main conclusion from the symposium –a common methodology is needed to use with the Architectural Framework(s).

The long list of topics in the Call for Papers, was quite well addressed and only one was left partially uncovered – Cloud Computing. This topic was dealt with by the keynote speaker and touched on by one other paper, but it was not a major focus of the symposium. Given that Cloud Computing is assigned to the IST Panel as an Emerging and Potentially Disruptive Technology to watch, it may warrant further attention in future activities of the panel.

In summary, the symposium accomplished its objectives of exposing the main issues and irritants encountered when developing formal software architectures using the main military standards such as NAF, MODAF and DODAF. Many useful development tools and techniques were exposed, and valuable lessons learned were discussed. The level of discussion was high and probed the issues well.