
Benefits and Challenges of Architecture Frameworks

Daniel Ota

Michael Gerz

{daniel.ota|michael.gerz}@fkie.fraunhofer.de

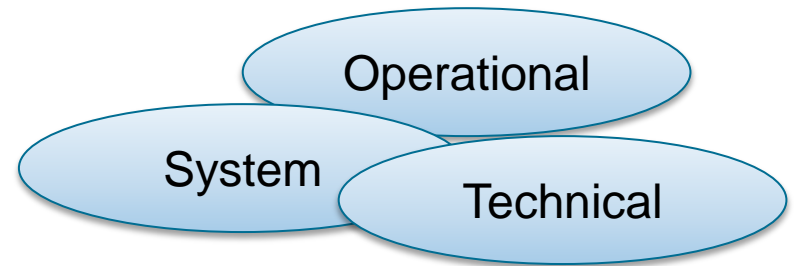
Fraunhofer Institute for Communication,
Information Processing and Ergonomics FKIE
Neuenahrer Straße 20
53343 Wachtberg
Germany

Outline

- Introduction
- NATO Architecture Framework
- User Expectations
- Semantic Issues
- Organizational Aspects
- Tool Support
- Conclusion

Introduction

- Architectures describe parts and excerpts of the real world
 - Considered from different perspectives
 - Varying in levels of abstraction
- Definition (IEEE 1471)
 - The fundamental organization of a system embodied in
 - Its components, their relationships to each other and to the environment
 - And the principles guiding its design and evolution
- Architectures tend to be large and complex
- Require different modeling techniques



Architecture Frameworks

- Modeling architectures requires guidance
 - Architecture frameworks as “templates” for a variety of different architectures

- Architecture frameworks are based on similar concepts
 - Set of architecture views
 - Common terminology
 - Meta model
 - Architecture types
 - Methodology and procedures

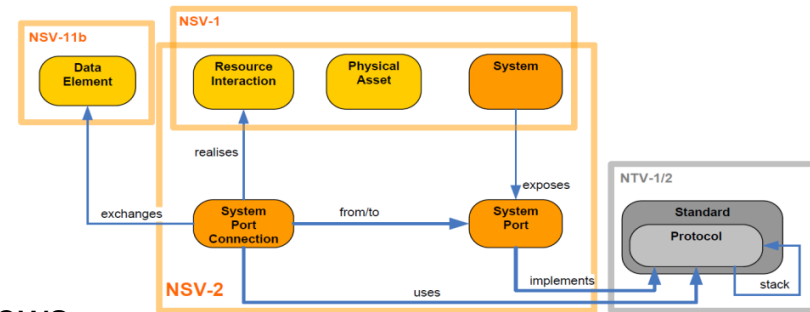
NATO Architecture Framework

NATO Architecture Framework

- NATO Architecture Framework (NAF) Version 3 published in November 2007
- Seven groups of views
 - NATO All View (NAV)
 - Overarching aspects of the architecture (context, scope, etc.)
 - NATO Operational View (NOV)
 - Tasks and activities of organizational elements
 - Types of information flows and frequency of information exchanges
 - NATO System View (NSV)
 - Systems, their components, interfaces, and interconnections
 - Performance parameters and properties of connections
 - Further: Technical View, Capability View, Service-Oriented View, Program View

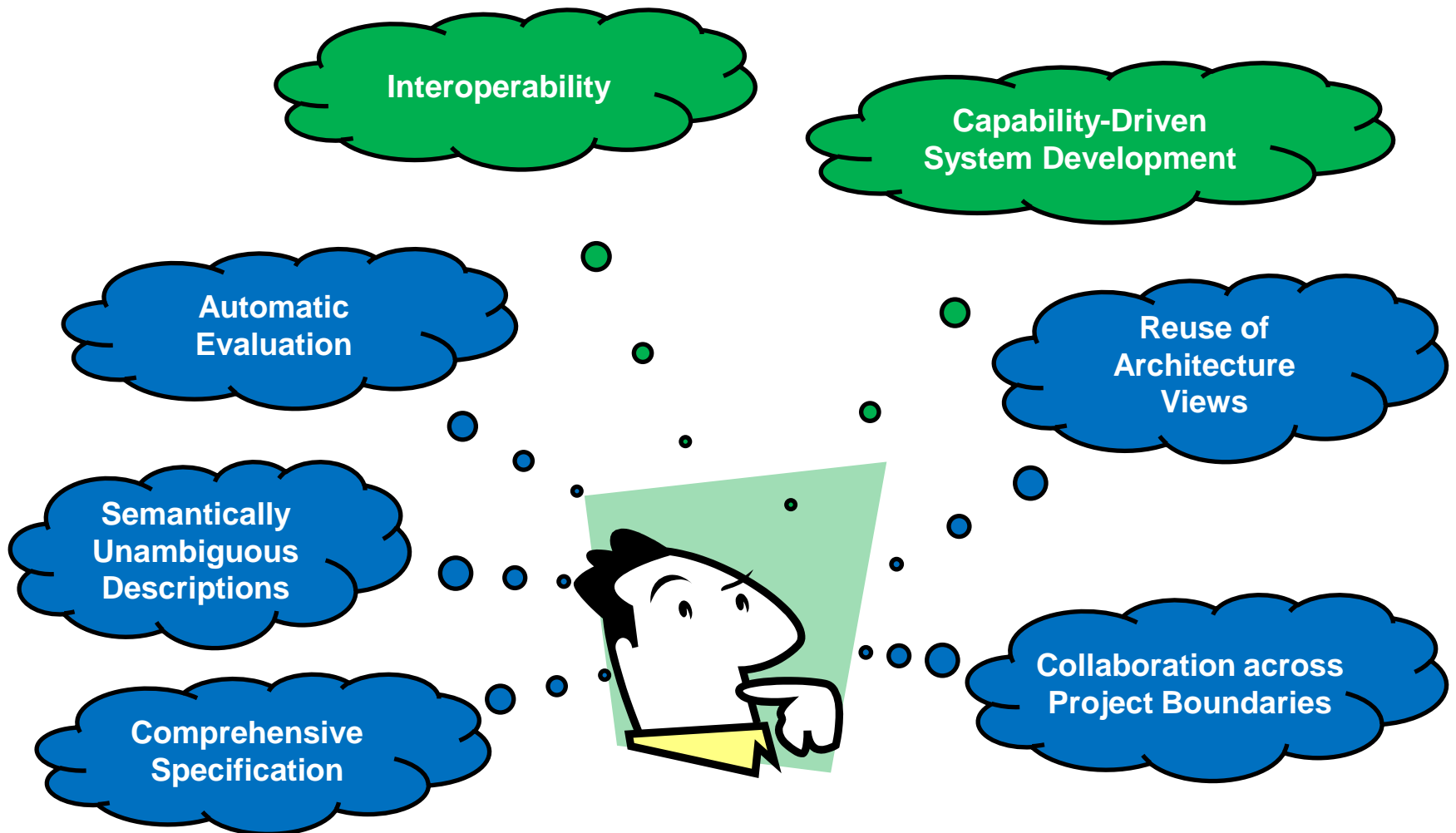
Views & Meta Model

- Views divided into a number of subviews
- Details for each subview
 - Purpose and definition
 - Allowed objects and components
 - Relationships within a view and to other subviews
- NAF Meta Model defined in Unified Modeling Language
 - Formal syntax
 - Ensuring consistency of views
 - Linking architectures and their components
 - Contains glossary
 - Semantics of each element to achieve common understanding



[NAF v3, ch. 5, p. 80]

User Expectations



Semantic Issues

Terminology

- Adapting the terminology of architecture frameworks
- Example: “Capability”
 - NAF 3 Glossary: “A high level specification of the enterprise's ability.”
 - Army: intelligence, mobility, resistance, ...
 - Interoperability program: ability to exchange data in joint operations
 - Distinguishing between the concepts “capabilities”, “services”, and “system functions” is difficult
 - Too detailed capabilities may resemble services
- ➔ Common terms need to be stated more precisely in a specific application context

Design of Views

- NAF distinguishes between operational and system concepts
 - Operational Node: “A logical entity that performs operational activities”
 - System: “A coherent combination of physical artefacts, energy and information, assembled for a purpose (software-intensive)”
- How to model the interaction between systems and human operators?
 - Operational nodes that make use of systems
 - Systems
 - Parts of a system (e.g., a commander within a vehicle)
- Implications on the reuse of architecture views and the representation of specific aspects, such as swivel chair interfaces

Context of Views

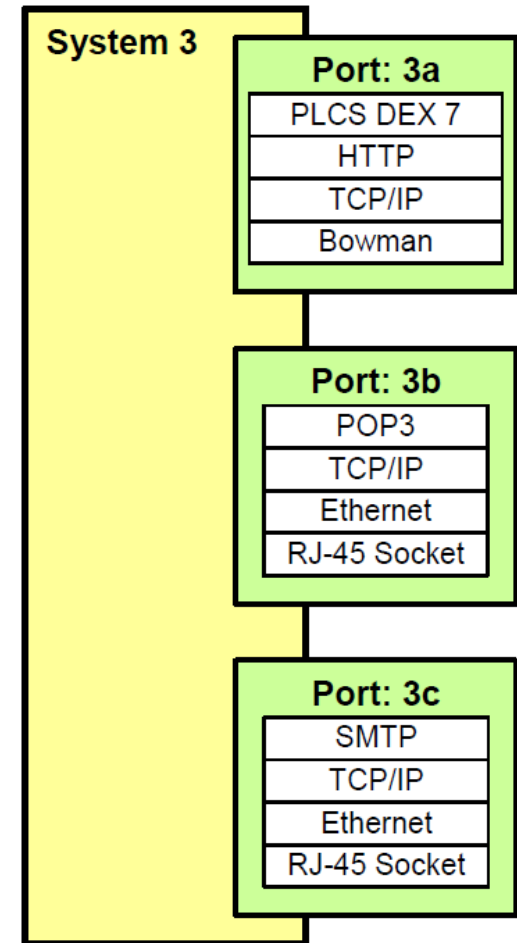
- Architecture description by a collection of views
 - Even individual systems characterized by series of views:
 - NSV-1 – System Interface Description
 - NSV-7 – System Quality Requirements Description
 - NSV-11 – System Data Model

- Isolated products without mechanism to
 - Group several views logically
 - Define their context

- Solutions outside the scope of NAF
 - Naming conventions
 - Specific features of modeling tools

Semantics of Model Elements

- Internal structure of technical systems
 - Informal description of the semantics of ports and port connections
 - [NAF v3, ch. 4, p. 64]
- Determine the number of physical ports
 - Three distinct ports?
 - Two physical instances?
- No automated interoperability checks
- Confusion of inexperienced users



Complexity of Real-Life Systems

- Many variants of a system
- Options
 - Model all system variants explicitly in independent views
 - Model a generic base system and document variants informally
- Modern C2IS supports many interfaces
 - MIP, Link-11/16/22 , ADatP-3 (selected message text formats only), etc.
 - Formal documentation impossible
 - Decide on what information is relevant and what has to be generalized
 - No reasoning on interoperability of heterogeneous C2IS

Organizational Aspects

Cross-Organizational Modeling Process

- Architecture design requires a modeling processes
 - Who provides which views at which stage and with what level of detail?
 - Mapping of process and associated user roles onto existing organizational units
 - Consideration of all interest groups into the modeling process from the very beginning
 - Sharing of common understanding of this process by all participants
 - Continuous checks if organization structure and organizational processes still adequate

Maintenance of Architectures

- Changing operational requirements and constraints
 - ➔ Architectural descriptions need to be maintained continually
- Reuse of architectural elements
 - Central architecture repository useful
- Organizational unit to coordinate all architecture modeling work
 - Provide methodological support
 - Enforce and adjust the enterprise modeling process
 - Identify relationships between different architectures
 - Avoid redundancies among different architectures
 - Harmonize views with regard to the level of abstraction, terminology and structure

Tool Support

Key Factors of Tool Sets

- Licensing fees
 - Availability of viewer application
- Export functionality
 - Representing information in different ways (graphics, lists, matrices, etc.)
- Linking formal and informal elements
 - Not all relevant information can be modeled formally
- Distributed modeling
 - Role-based approach to enforce proper access control
- Support and extensibility of the meta model
 - Offering allowed elements on a per-view basis
 - Point out potential inconsistencies across individual views
 - Meta model extensions on the level of individual architectures

Conclusion

Conclusion

- Architecture frameworks provide “templates” to design architectures in a structured manner
- Weak semantics of the NAF meta model
 - Restricted automated analysis
 - Not perfectly suited for detailed system specifications
 - Reuse of architecture views problematic
- Lack of guidance and ambiguities
 - Permanent coordination throughout entire modeling process
 - Modeling approach must include development and maintenance procedures

Thank you for your attention!