Characterizing Architecture Description Languages for Software-Intensive Systems-of-Systems

MILENA GUESSI, EVERTON CAVALCANTE, AND LUCAS B. R. OLIVEIRA

ICSE/SeSoS, 17 May, 2015 – Florence, IT

Agenda

Introduction

Features of ADLs for Describing SoS Software Architectures

ADLs Assessment

Conclusions and Future Research

INTRODUCTION

SoS Software Architecture

SoS are formed by independent, complex software systems, called constituents

- Independent action and decision making

Comprise the structure of constituents, the relationships that exist among them, and the principles and guidelines governing the SoS design and evolution

Architecture Description

Main artifact expressing the software architecture

- Support analysis, construction, and evolution of SoS

Architecture descriptions for SoS:

- Minimize the risk of constituents' stakeholders not being aware of important elements of the SoS
- Support evolution and analysis

Techniques for Describing Software Architectures

Types:

- Formal languages
  - Based on formal syntax and semantics
- Semi-formal languages
  - Based on defined syntax and lack a complete semantics
- Informal languages

Each type promotes different aspects of architecture descriptions:

- Ease of use, reuse, unambiguous, accuracy, correctness

Motivation and Objectives

There is no consensus about:

- Important characteristics for describing SoS
- Appropriate ADLs for describing SoS

To identify the main features that should be provided by ADLs for SoS

Assess ADLs applied/suggested for describing SoS software architecture

FEATURES OF ADLS FOR DESCRIBING SoS SOFTWARE ARCHITECTURES

Traditional Features of ADLs

Building Blocks:

- Components
- Connectors
- Configurations

Features covered:

- Interfaces
- Constraints
- Behavior

Medvidovic, N. and Taylor, R. N. A classification and comparison framework for software architecture description languages. IEEE Transactions on Software Engineering, vol. 26, no. 1, pp. 70—93, 2000

Features of ADLs for SoS

Constituents

- Systems or SoS
- Interface (contracts, assumptions)

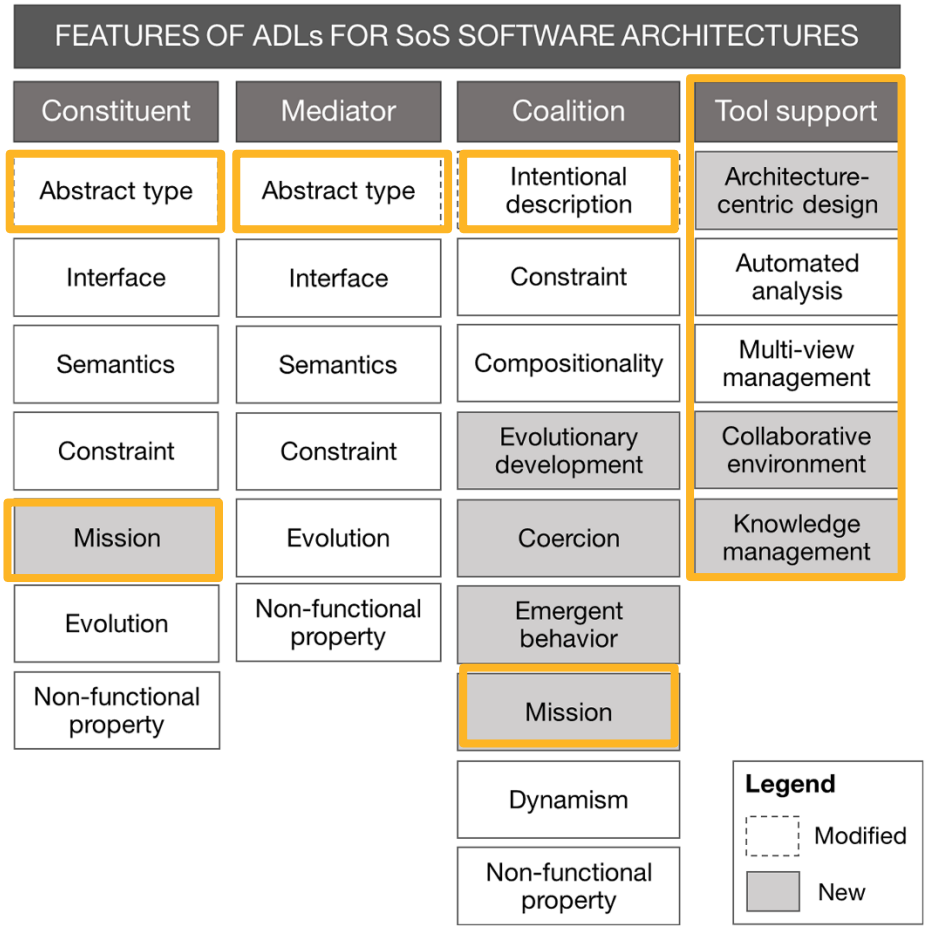
Mediators

- Complex entity negotiating the interaction between constituents

Coalitions

- Arrangement of constituents and mediators

Features of ADLs for SoS



ADLs ASSESSMENT

ADLs Assessment

UML
- Semi-formal
- Most commonly used language for describing SoS

SysML
- Semi-formal

SysML + CML
- Semi-formal + formal
- Definition of contracts

X-UNITY
- Formal
- Definition of interfaces

TABLE I
DESIGN TIME FEATURES AVAILABLE IN CURRENT ADLS FOR REALIZING SOS ARCHITECTURAL DESCRIPTION

Features		ADL			
		UML	*SysML*	*SysML + CML*	*X-UNITY*
Constituent	Abstract type	●	●	●	●
	Interface	●	●	●	○
	Semantics	●	●	●	-
	Constraint	○	●	●	●
	Mission	-	○	-	-
	Evolution	-	-	-	-
	Non-functional property	○	●	●	-
Mediator	Abstract type	○	○	-	-
	Interface	●	●	-	-
	Semantics	●	●	●	-
	Constraint	○	●	●	-
	Evolution	-	-	-	-
	Non-functional property	○	●	●	-
Coalition	Intentional description	○	○	○	●
	Constraint	-	●	●	●
	Compositionality	●	●	●	●
	Evolutionary development	-	-	○	-
	Coercion	-	-	●	○
	Emergent behavior	○	○	●	-
	Mission	-	○	-	-
	Dynamism	-	-	●	-
	Non-functional property	○	●	●	-
Tool support	Architecture-centric design	○	○	○	-
	Automated analysis	-	○	●	-
	Multi-view management	○	○	●	-
	Collaborative environment	-	-	●	-
	Knowledge management	-	-	-	-

● : feature is supported, ○ : feature is partially supported, - : feature is not supported

ADLs Assessment

None of these languages completely support this set of features at runtime[1]

- Lack support for evolution
- SysML + CML seems the most complete approach in terms of covered features

[1] features that can be both designed and enacted at runtime

CONCLUSIONS AND FUTURE RESEARCH

Conclusions

Set of features for expressing SoS software Architecture

Desirable features of tools supporting these ADLs

Assessment of current ADLs points out directions for future research

Future Research

Formal ADLs for SoS

- Ensure correctness and support the evolutionary development
- Requires specialized tools

Properties of ADLs for SoS

- Contributions for improving understandability, scalability, refinement, traceability, and automation

Viewpoints for describing SoS

- Most ADLs focus on the structure and behavior of SoS
- Investigate other viewpoints that might be necessary

Characterizing Architecture Description Languages for Software-Intensive Systems-of-Systems

Contact authors: milena@icmc.usp.br,
evertonrsc@ppgsc.ufrn.br, buenolro@icmc.usp.br

ICSE/SeSoS, 17 May, 2015 – Florence, IT