CBSoft2014
Congresso Brasileiro de Software: Teoria e Prática
28 de Setembro a 03 de Outubro de 2014
Maceió - Alagoas

Do Search-Based Approaches Improve the Design of Self-Adaptive Systems ? A Controlled Experiment

Sandro S. Andrade and Raimundo J. de A. Macêdo

Distributed Systems Laboratory (LaSiD)
Department of Computer Science
Federal University of Bahia

{sandros, macedo}@ufba.br

LaSiD UFBA

Agenda

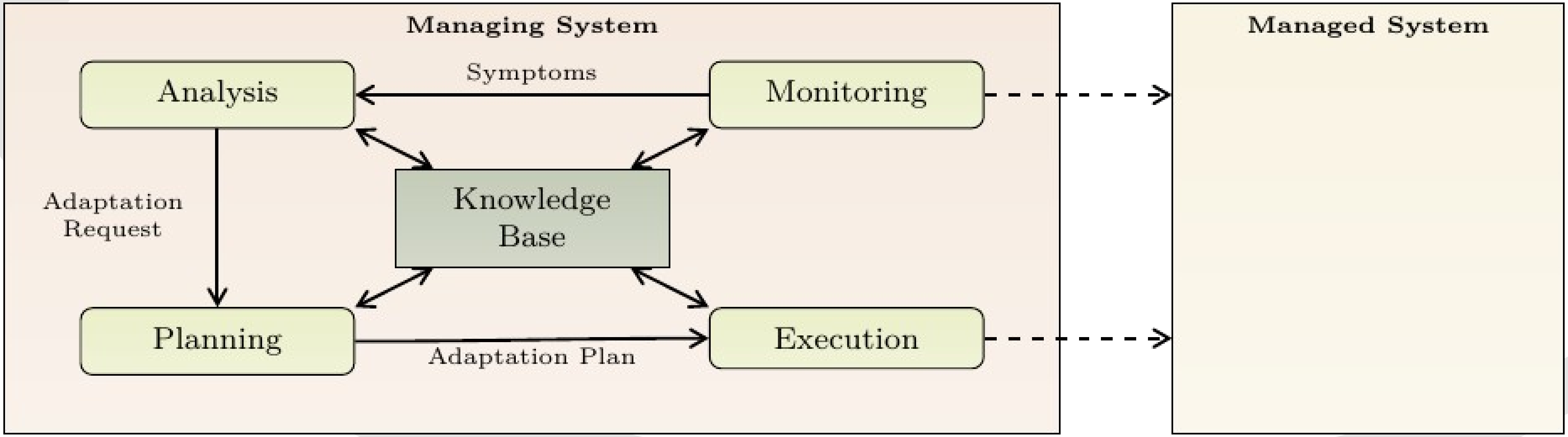
- Context & motivation
- Our search-based architectural design approach
- The quasi-experiment
- Contributions of this paper
- Final remarks

Context & Motivation

- The need for self-adaptive systems (SAS):
  - Dynamic and uncertain operating environments
  - Stringent requirements for scalability, dependability, energy-efficiency, and performance
  - Incomplete/partial requirements
  - Problem/design space complexity approaching the limits of human capability
  - Rapidly varying workloads and data characteristics
  - Some problems are inherently self-adaptive or self-organizing (eg.: robots motion coordination, particles/swarm optimization, etc)

Context & Motivation

- MAPE-K Reference Architecture [1]



[1] J. O. Kephart and D. M. Chess. Computer, vol. 36, no. 1, pp. 41–50, Jan. 2003.

Context & Motivation

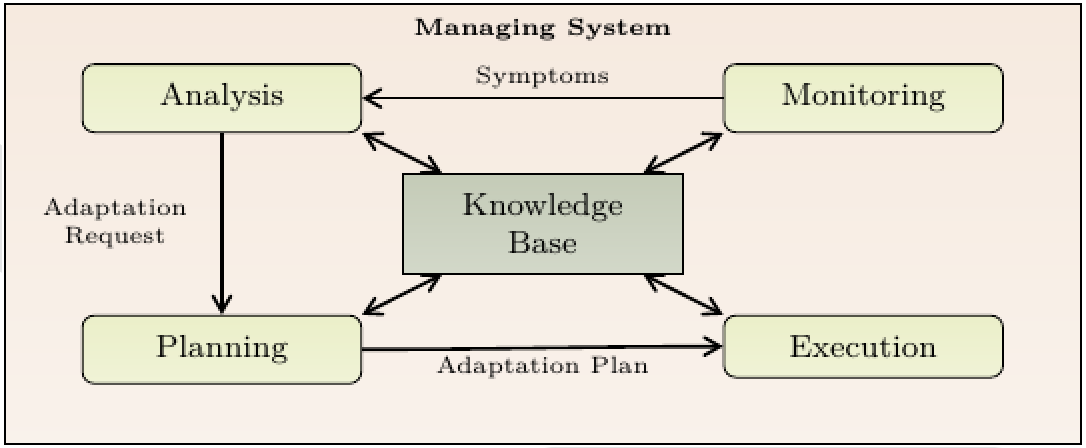
- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?

Goals:
- Type (regulation, tracking, disturbance rejection, optimization)
- Evolution (static, dynamic)
- Duration (persistent, temporary)
- Multiplicity (single, multiple)
- Dependency (independent, complementary, conflicting)
- Flexibility (rigid, constrained, unconstrained)

Change:
- Source (internal, external)
- Anticipation (foreseen, foreseeable, unforeseen)
- Frequency (rare, frequent)
- Type (functional, non-functional, technological)

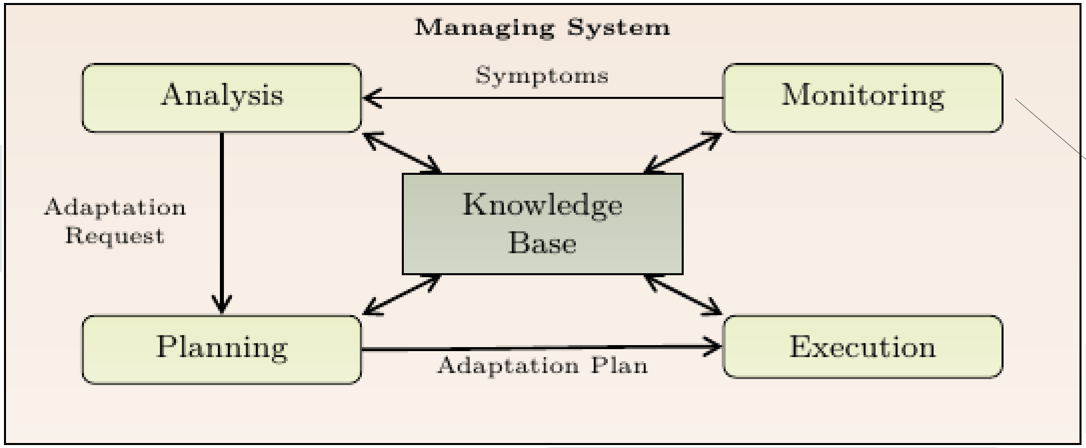
Context & Motivation

- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?
  - Large and complex solution space



Context & Motivation

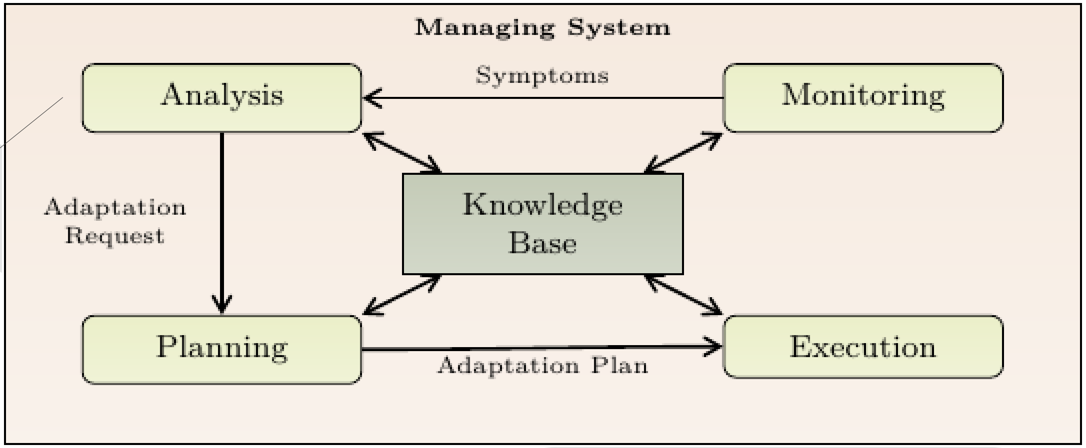
- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?
  - Large and complex solution space



- Type (parametric, structural)
- Mechanism (signaling, frameworks, profiling, etc)
- Uncertainty handling (redundancy, filtering, smoothing)

Context & Motivation

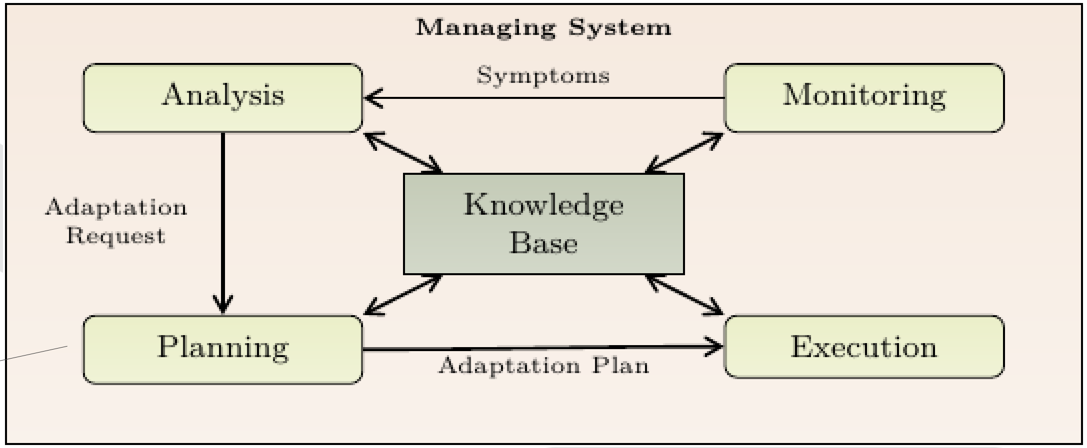
- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?
  - Large and complex solution space



- Mechanism (symptoms databases, policies, queue networks, game theory, etc)
- System/Environment representation (ad-hoc, models)

Context & Motivation

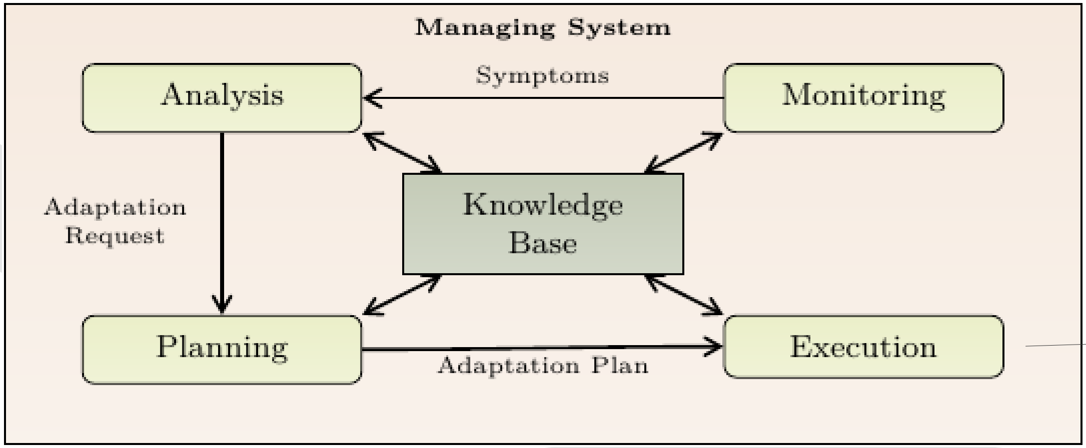
- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?
  - Large and complex solution space



- Actuation law (intelligent agents, control theory, search-based, etc)
- Cardinality (single, utility functions, parent-front)
- Type (static, dynamic)

Context & Motivation

- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?
  - Large and complex solution space



- Type (parametric, structural)
- Mechanism (function pointers, aspect weaving, metaobjects, middleware-based)
- Timeliness (best-effort, guaranteed)

Context & Motivation

- Current challenges:
  - Intricate problem space
    - How to effectively elicit the adaptation requirements ?
  - Large and complex solution space
    - How to come up with an effective managing system architecture for the adaptation requirements at hand ?
    - How to judiciously evaluate all available design alternatives ?
    - How to make well-informed decisions about quality attributes trade-offs ?
  - Consequences:
    - Lack of organized design knowledge for routine use
    - False intuition, design bias, and sub-optimal architectures

Our Search-Based Design Approach – DuSE

- Research questions

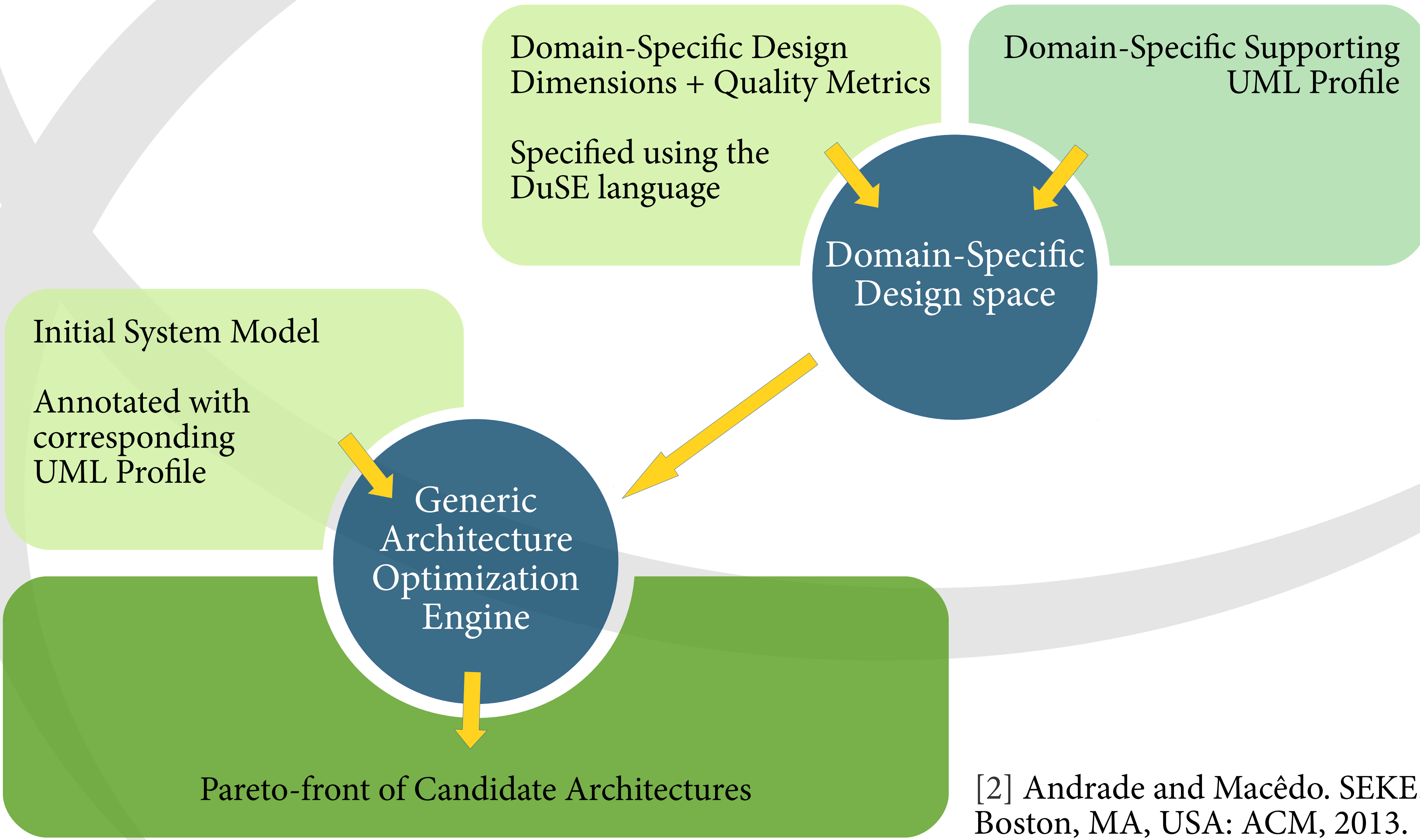
To which extent may SAS design knowledge be systematically represented for routine use ?

How to support well-informed decision making regarding quality attribute trade-offs between alternative architectures for SAS ?

Our Search-Based Design Approach – DuSE

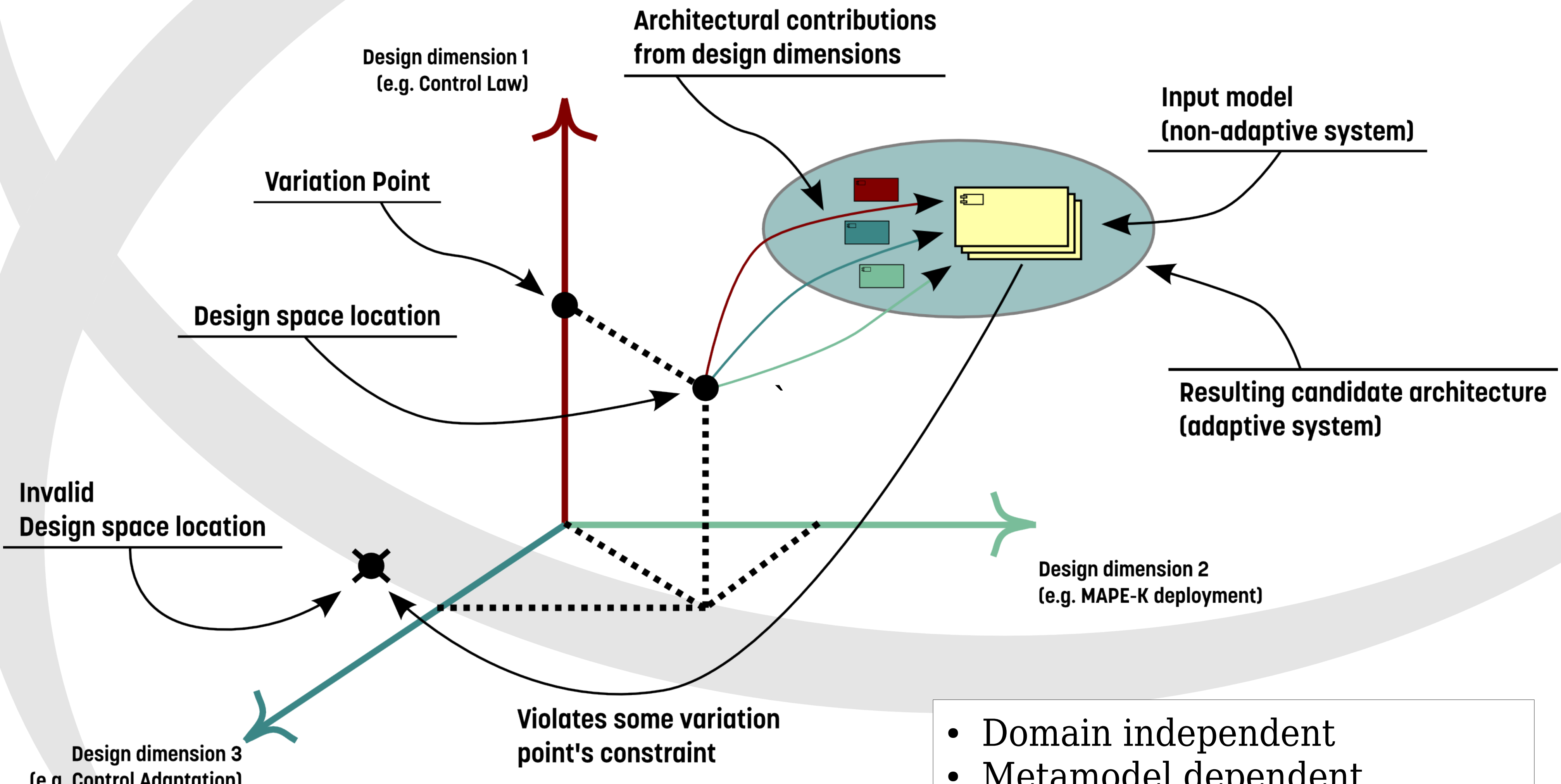
- We combine the use of ...
  - Metamodeling and Domain-Specific Languages (DSL)
  - Structured Architecture Design Spaces
  - Multi-Objective Optimization
- … in order to …
  - Enable a more disciplined and automated "handbook" of SAS design
  - Provide a solid basis for choosing between architectures which exhibit different quality attributes

Our Search-Based Design Approach – DuSE



[2] Andrade and Macêdo. SEKE. Boston, MA, USA: ACM, 2013.

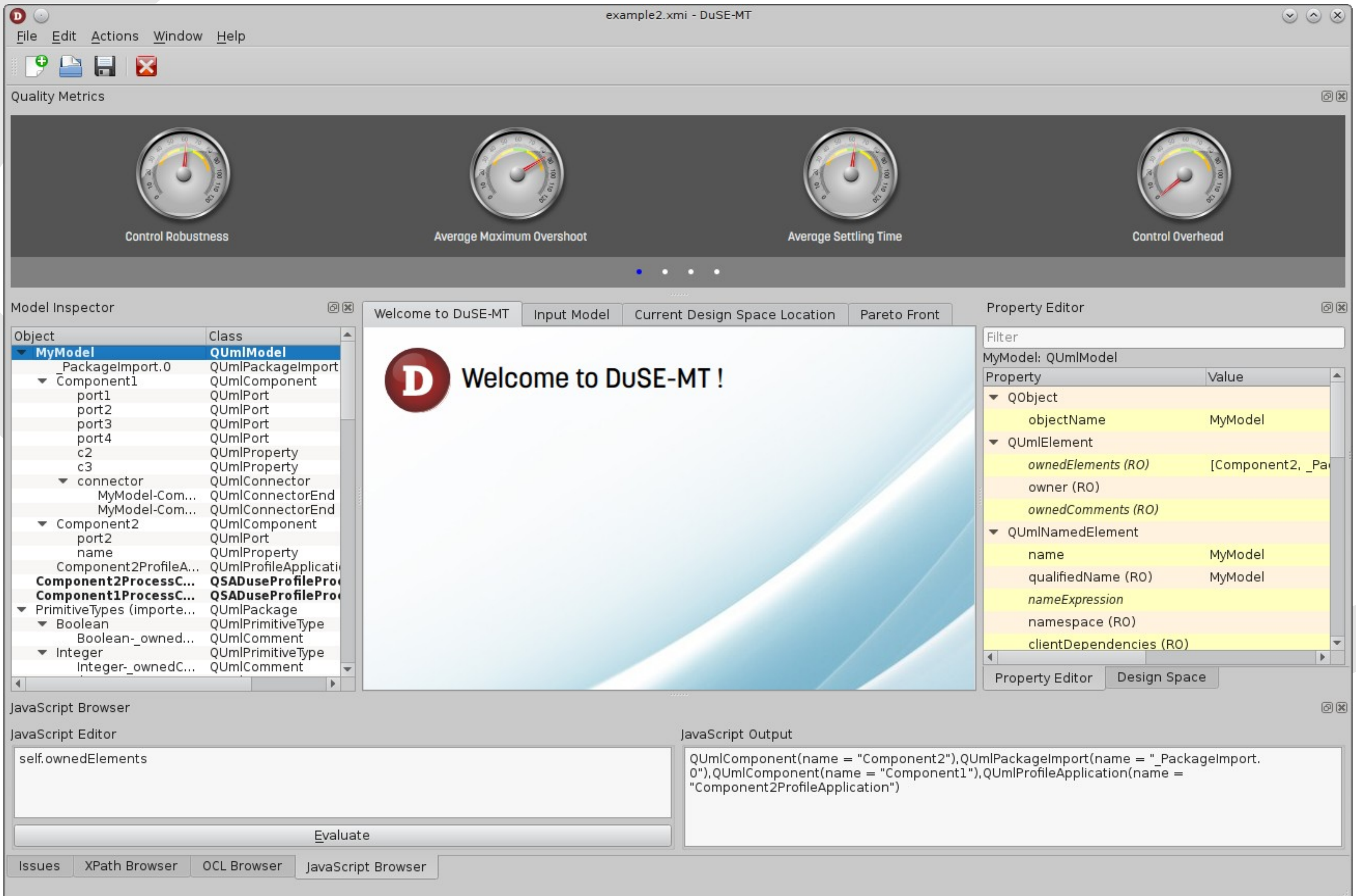
Our Search-Based Design Approach – DuSE



- Domain independent
- Metamodel dependent
- Design space navigation engine
- Optimization engine (NSGA-II)

[3] Andrade and Macêdo. SASO. Philadelphia, PA, USA: IEEE, 2013.

Our Search-Based Design Approach – DuSE



[4] Andrade and Macêdo. SASO Demo Session. Philadelphia, PA, USA: IEEE, 2013.

Our Search-Based Design Approach – SA:DuSE

- SA:DuSE
  - A particular DuSE instance which captures the most prominent design dimensions of SASs
- Evaluation dimensions:
  - Is SAS design indeed a multi-objective problem ?
  - To which extent the quality of Pareto-optimal architectures are indeed observed in real prototypes ?
  - **Do search-based approaches improve the design of SAS ?**

[5] Andrade and Macêdo. WoSiDA 2014. Florianópolis, SC, Brasil. 2014.

The Quasi-Experiment

- **Analyze** the design of SAS
- **For the purpose of** evaluating the search-based design approach we propose and a design process based on architecture styles catalogs
- **With respect to** the effectiveness and complexity of resulting architectures, as well as the method's potential for leveraging the acquisition of distilled design knowledge by novice architects
- **From the point of** view of researchers
- **In the context of** graduate students endowing systems with self-adaptation capabilities

The Quasi-Experiment

- Quasi-experiment:
  - Blocked subject-object study with a paired comparison design
- Subjects:
  - 24 students of a graduate program in Distributed and Ubiquitous Computing
- Experiment objects:
  - Two UML models representing the managed system (web server and MapReduce distributed architecture)
- Experiment tests:
  - Design a managing system architecture for both objects
  - Answer a questionnaire on quality attribute trade-offs in SAS

The Quasi-Experiment

- Independent variable:
  - Design method
    - Search-based approach
    - Style-based approach
- Dependent variables:
  - Effectiveness of resulting SAS architectures
  - Complexity of resulting SAS architectures
  - The method's potential for leveraging the acquisition of distilled SAS design knowledge by novice architects
- Experiment website:
  - http://wiki.ifba.edu.br/tr-ce012014

The Quasi-Experiment

- Experiment timetable

Part	Day	Activity
Lectures	1	Self-Adaptive Systems Foundations (motivation, MAPE-K reference architecture, current approaches, challenges)
	2	Feedback Control Introduction (control goals, control properties, fixed gain SISO approaches)
	3	Feedback Control (MIMO and adaptive approaches)
	4	Self-Adaptive Systems - Case Studies
Exam and Training	5	First hour: discussion Next 3h: Pen and paper exam
	6	First hour: exam discussion Next 3h: Training (DuSE-MT and architectural styles catalog)
Experiment	7	First 110min: Tests #1 and #2 Next 110min: Tests #3 and #4 Next 20min: Tests #5 and #6 (questionnaire)
	8	First 110min: Tests #7 and #8 Next 110min: Tests #9 and #10 Next 20min: Tests #11 and #12 (questionnaire)

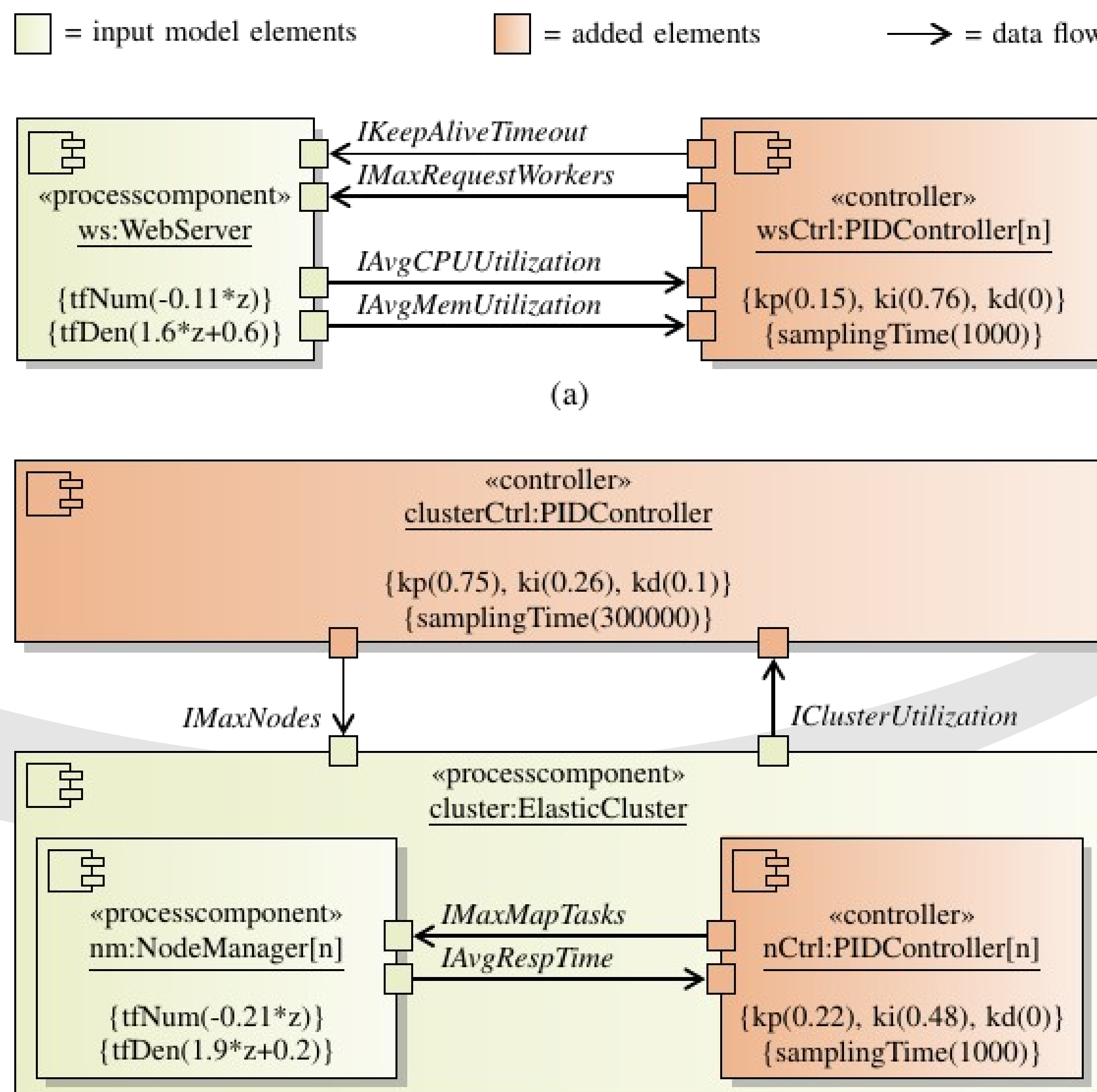
The Quasi-Experiment

- Experiment tests

#Test	Object	Treatment	Subjects
1	Web Server	Style-Based Approach	Group 1
2	MapReduce Architecture	Search-Based Approach	Group 2
3	MapReduce Architecture	Style-Based Approach	Group 1
4	Web Server	Search-Based Approach	Group 2
5	Questionnaire	Style-Based Approach	Group 1
6	Questionnaire	Search-Based Approach	Group 2
7	MapReduce Architecture	Search-Based Approach	Group 1
8	Web Server	Style-Based Approach	Group 2
9	Web Server	Search-Based Approach	Group 1
10	MapReduce Architecture	Style-Based Approach	Group 2
11	Questionnaire	Search-Based Approach	Group 1
12	Questionnaire	Style-Based Approach	Group 2

The Quasi-Experiment

- Experiment objects



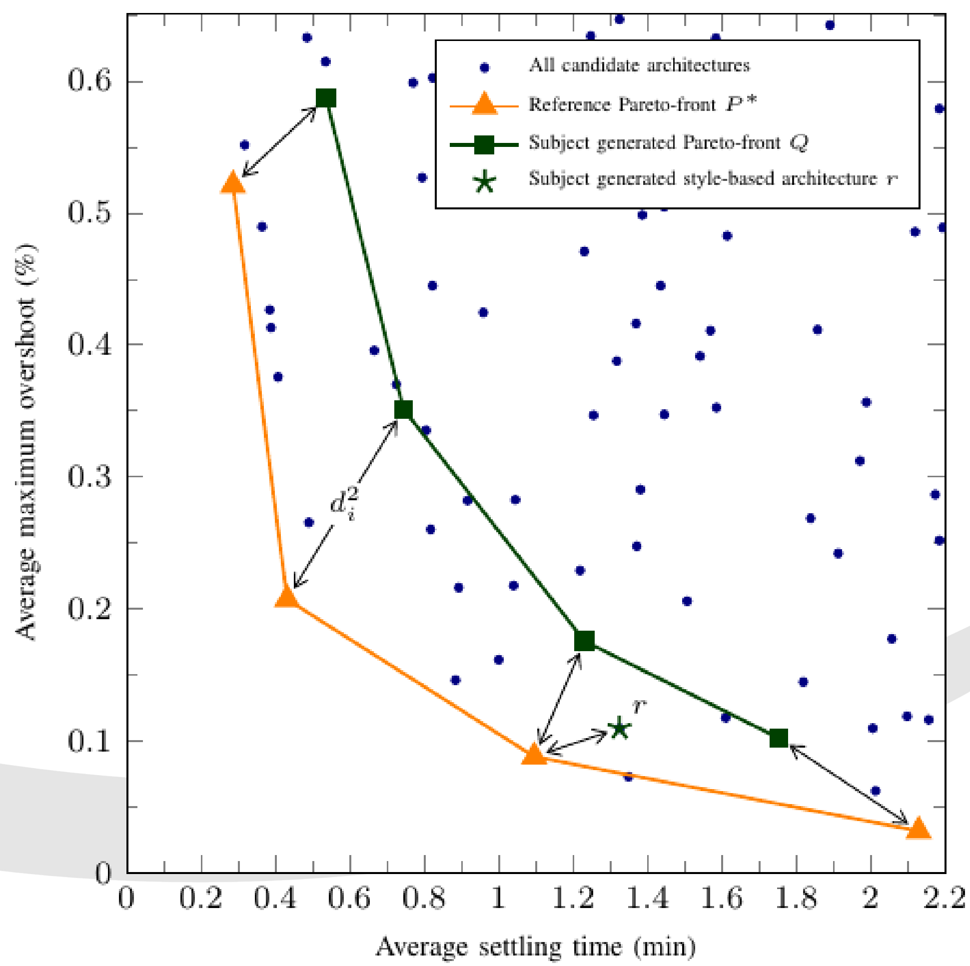
The Quasi-Experiment

- Measuring effectiveness (Generational Distance)

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|}$$

where

$$d_i^2 = \min_{k=1}^{|P*|} \sqrt{\sum_{m=1}^{M} \left(f_m^{(i)} - f*_m^{(k)}\right)^2}$$



[6] Deb and Kalyanmoy. John Wiley & Sons, Inc. NY, USA: 2001.

The Quasi-Experiment

- Measuring complexity (Component Point)

$$CC_c = IFCI_c + ITCI_c = \frac{IFC_c}{n_c} + \frac{ITC_c}{m_c}$$

$$IFC_c = \sum_{j=1}^{2} \sum_{k=1}^{3} I_{jk} \times W_{jk}$$

$$ITC_c = \sum_{i=1}^{p} \sum_{j=1}^{q} \left(IF_{ij} \times \sum_{k=1}^{r} CM_{ijk} \right)$$

$$CM_{ijk}(D, L) = L + \sum_{n=1}^{m} CM(DT_n, L+1)$$

[7] Wijayasiriwardhane and Lai. Journal of Systems and Software. 2010.

The Quasi-Experiment

- Measuring the acquisition of distilled design knowledge:
  - Questionnaire with 10 multiple choice questions
  - Questions related to quality attributes trade-offs in the SAS domain
  - Questionnaire answered at the end of each experimentation day
  - Normalized grades assigned to each student/day

The Quasi-Experiment

- Hypotheses formulation

H_0^1 - there is no difference in design effectiveness between a feedback control loop design created using the style-based approach (RA) and a feedback control loop design created using the search-based approach (IA)

$$H_0^1 : \mu_{GD_{RA}} = \mu_{GD_{IA}}$$
$$H_1^1 : \mu_{GD_{RA}} > \mu_{GD_{IA}}$$

H_0^2 - there is no difference in design complexity between a feedback control loop design created using the RA and a feedback control loop design created using the IA

$$H_0^2 : \mu_{AC_{RA}} = \mu_{AC_{IA}}$$
$$H_1^2 : \mu_{AC_{RA}} > \mu_{AC_{IA}}$$

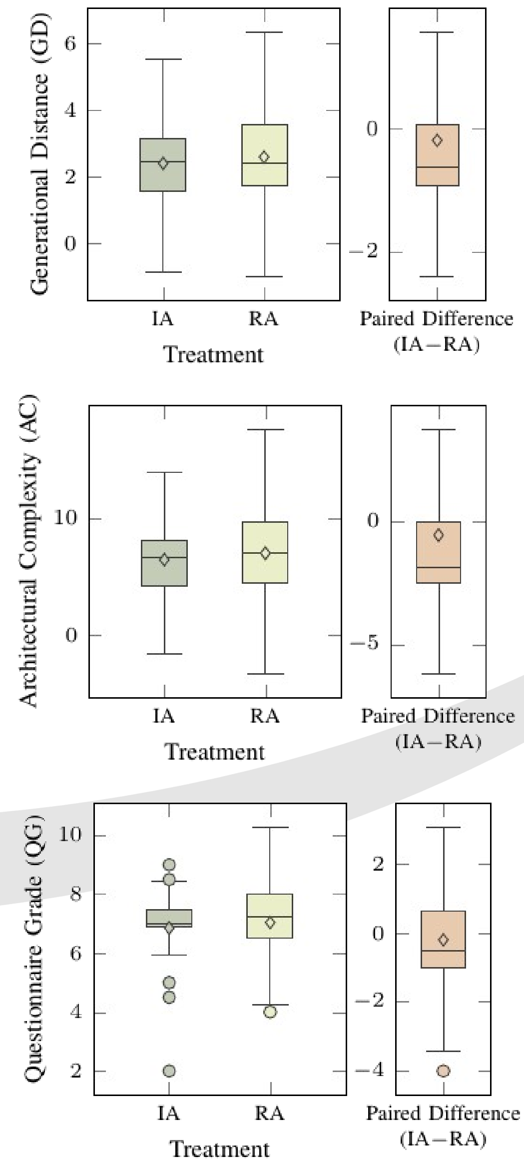
H_0^3 - there is no difference in the acquisition of distilled design knowledge between a feedback control loop design created using the RA and a feedback control loop design created using the IA

$$H_0^3 : \mu_{QG_{RA}} = \mu_{QG_{IA}}$$
$$H_1^3 : \mu_{QG_{RA}} < \mu_{QG_{IA}}$$

The Quasi-Experiment

- Descriptive statistics

Generational Distance (GD)			
	mean(μ)	median	std. dev.
Search-Based Approach (IA)	2.40	2.45	1.08
Style-Based Approach (RA)	2.59	2.41	1.03
Difference (IA−RA)	-0.19	-0.62	1.32
Architecture Complexity (AC)			
	mean(μ)	median	std. dev.
Search-Based Approach (IA)	6.46	6.65	2.77
Style-Based Approach (RA)	7.02	7.05	2.70
Difference (IA−RA)	-0.57	-1.90	3.47
Questionnaire Grade (QG)			
	mean(μ)	median	std. dev.
Search-Based Approach (IA)	6.85	7.00	1.43
Style-Based Approach (RA)	7.04	7.25	1.27
Difference (IA−RA)	-0.19	-0.50	1.51



The Quasi-Experiment

- Assumptions of parametric tests:
  - Data is taken from an interval or ratio scale
  - Observations are independent
  - Population variances are equal between groups (homoscedasticity)
  - Measured values are normally distributed

Dependent Variable	Anderson-Darling p-value	Brown–Forsythe p-value
Generational Distance	1.29814505086953E-007	0.9009324909
Architecture Complexity	2.88672812219819E-010	0.7207666486
Questionnaire Grade	0.635529605	0.7167840476

$\alpha=0.05$

The Quasi-Experiment

- Assumptions of parametric tests:
  - Data is taken from an interval or ratio scale
  - Observations are independent
  - Population variances are equal between groups (homoscedasticity)
  - Measured values are normally distributed

Cannot reject homoscedasticity

Dependent Variable	Anderson-Darling p-value	Brown–Forsythe p-value
Generational Distance	1.29814505086953E-007	0.9009324909
Architecture Complexity	2.88672812219819E-010	0.7207666486
Questionnaire Grade	0.635529605	0.7167840476

$\alpha=0.05$

The Quasi-Experiment

- Assumptions of parametric tests:
  - Data is taken from an interval or ratio scale
  - Observations are independent
  - Population variances are equal between groups (homoscedasticity)
  - Measured values are normally distributed

Can reject normal distribution

Dependent Variable	Anderson-Darling p-value	Brown–Forsythe p-value
Generational Distance	1.29814505086953E-007	0.9009324909
Architecture Complexity	2.88672812219819E-010	0.7207666486
Questionnaire Grade	0.635529605	0.7167840476

α=0.05

The Quasi-Experiment

- Assumptions of parametric tests:
  - Data is taken from an interval or ratio scale
  - Observations are independent
  - Population variances are equal between groups (homoscedasticity)
  - Measured values are normally distributed

Wilcoxon Signed-Rank Test

Dependent Variable	Anderson-Darling p-value	Brown–Forsythe p-value
Generational Distance	1.29814505086953E-007	0.9009324909
Architecture Complexity	2.88672812219819E-010	0.7207666486
Questionnaire Grade	0.635529605	0.7167840476

α=0.05

Paired t-test

The Quasi-Experiment

- Results of the statistical tests

H_0^i	Test	Criteria	Conclusion
1	Wilcoxon Signed-Rank	T(410) > T-critical(378)	Rejected
2	Wilcoxon Signed-Rank	T(367) > T-critical(361)	Rejected
3	Paired t-test	p-value=0.5488018266	Not Rejected

α=0.05

The Quasi-Experiment

- Threats to validity
  - Construct:
    - Inadequate preoperational explication of constructs
    - Hypothesis guessing
    - Objects representativeness
  - Internal:
    - Maturation
    - Instrumentation
  - External:
    - Students acting as subjects
  - Conclusion:
    - Experiment holders creating the experiment objects

Contributions of this paper

- To the best of our knowledge, SA:DuSE is the first effort in applying search-based approaches in the SAS domain
- To the best of our knowledge, this is the second controlled experiment in the SAS domain (the first one is [8])
- We got some empirical evidence about the benefit of search-based approaches when designing SASs
- The potential for leveraging knowledge acquisition still deserves further investigation

[8] Weyns, Iftikhar, and Soderlund. SEAMS. 2013.

Final Remarks

- A lot of research towards principled and systematic design of self-adaptive systems
- Effectiveness of systematic design spaces in capturing intricate solution spaces
  - However, qualitative approaches may be used to address another facets of architectural design
- Future work:
  - SA:DuSE expansion (NSGA-III)
  - More investigation on real prototypes
  - From design spaces to design theories

CBSoft2014
Congresso Brasileiro de Software: Teoria e Prática
28 de Setembro a 03 de Outubro de 2014
Maceió - Alagoas

Thanks ! Questions ?

Sandro S. Andrade and Raimundo J. de A. Macêdo

{sandros, macedo}@ufba.br

LaSiD UFBA