Measuring the Effort for Creating and Using Domain-Specific Models

Yali Wu
PhD Candidate

18 October 2010

Outline

- Introduction
- Related Work
- Motivation
- Contributions
- Effort classification
- Effort Metrics
- Applying Metrics
- Conclusion

Introduction

- DSMLs lead to increased productivity through domain specific abstractions[1]
- Industrial experiences have shown DSMLs to be 5-10 times more productive[1]
- Is there a systematic approach of measuring the claimed benefits of using DSMLs?

1: www.dsmforum.org

Related Work

- Current ways of measuring DSM productivity
  - Qualitative results as anecdotal evidence
  - Measuring reduced development time
  - Measuring reduced implementation effort
- Software model metrics
  - Model heterogeneity created challenges
  - Initially focus on model size

Motivation

- How do we **systematically measure** the effort involved in domain specific modeling in a way that:
  - Take into concern the multiple dimensions of the DSM process?
  - Does not depend on specific DSMLs?

Contributions

- A classification of the effort involved in realizing applications using DSMLs
- A set of metrics for measuring each category of the involved effort
- A case study showing how these metrics could be applied to various DSMLs

Effort Classification

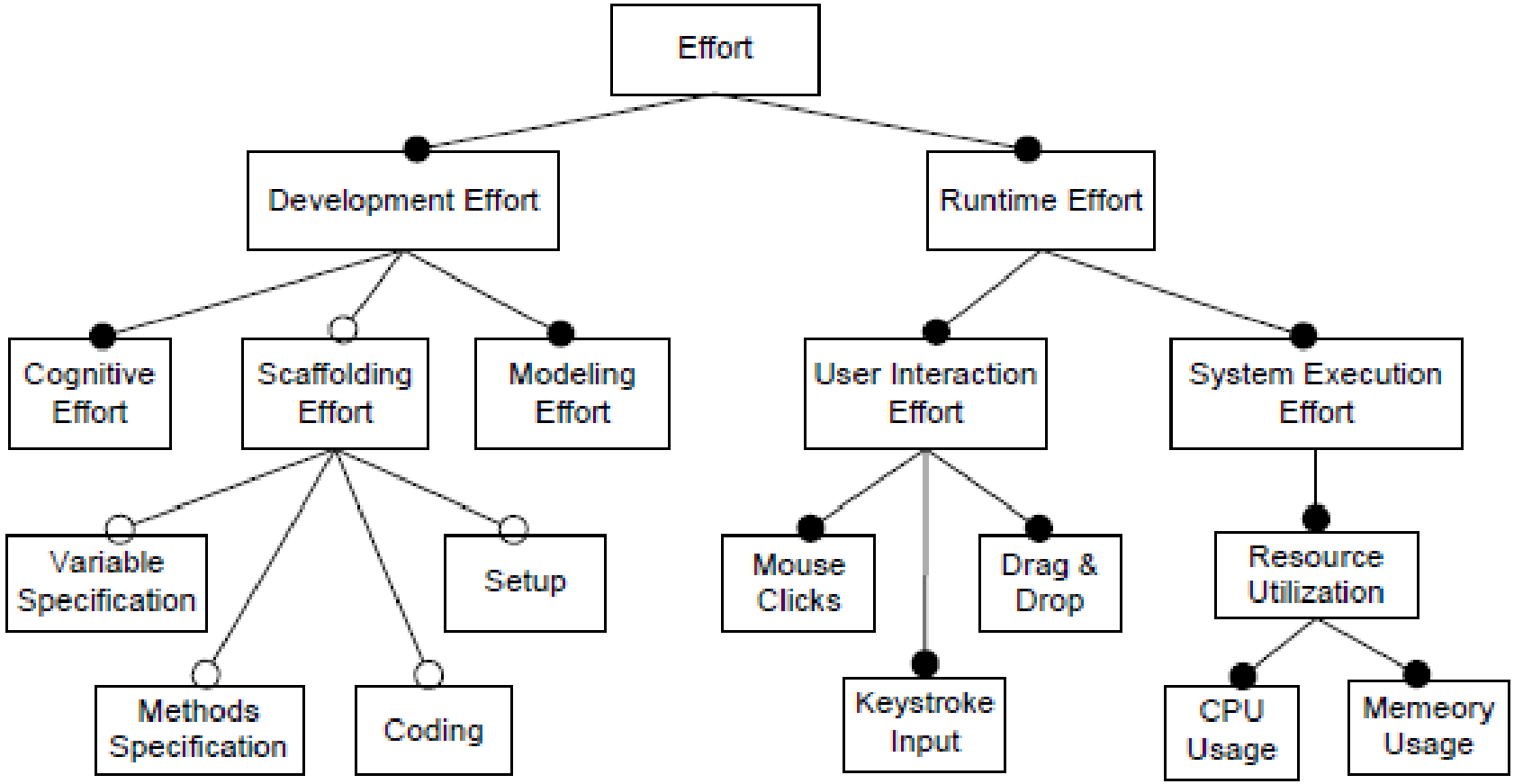


Figure 1: Feature Diagram to Classify Effort.

Development Effort

Effort Category	Effort Metrics	Explanation	Associated Property
Modeling Effort	Size Of Model	Effort required to	Conciseness
	Control Flow Complexity	del	
Cognitive Effort	Cognitive Weight	red to	Understandability Comprehensiveness
	Closeness of Mapping	ntal solutions to	
Scaffolding Effort	Addt. LOC	Additional effort required to create	Executablility
	Addt. Methods		

LOC Metric

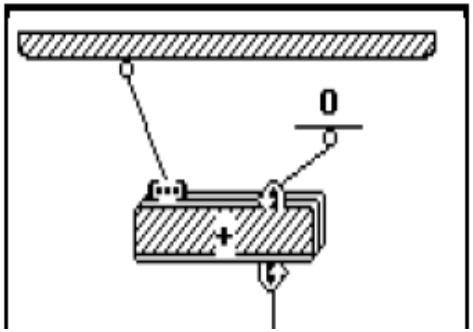
McCabe Metric

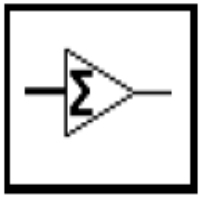
Cognitive Complexity

Usability Analysis

```
sum = 0;
for (i = 0; i < N;  i++)
{
   sum += x[i];
}
```

```
sum = 0
for i = 1 to N
   sum = sum + x(i)
next
```





=SUM(A1..A10)

Runtime Effort

Effort Category	Effort Metrics	Explanation	Measured Property
User Interaction Effort	# Mouse Clicks	Effort required to interact with the execution interface	Runtime Ease-Of-Use
	# Keyboard Input		
	# Drag'n'Drop		
System Execution Effort	CPU Utilization	Required system resource to map to executables at runtime	Runtime Resource Efficiency
	Memory Utilization		

Applying Metrics to DSMLs

- Model the same heathcare scenario using three different DSMLs
  - Yet Another Workflow Modeling Language [W.M.P. van der Aalst, 2005]
  - Windows Workflow Foundation[Microsoft 2010]
  - Workflow Communication Modeling Language [Wu et al. 2010]
- Collect metrics for realizing the DSML model

Results

Table 3: Development Effort

Modeling/ Cognitive	*SOM Top Level/Total*	*CFC*	*CW*	*COMR*
YAWL	21/21	9	61	14/17
WF	77/77	20	22	13/64
WF-CML	7/54	2	3	7/4
Scaffolding	*NALOC*	*NAV*	*NAM*	*NAC*
YAWL	857	34	3	38
WF	1265	77	1	58
WF-CML	0	0	0	0

Results -cont

Table 4: Runtime Effort

User Effort	*NMC*	*NDD*	*NKI*
YAWL	3	0	9
WF	15	0	1
WF-CML	3	5	1
System Effort	*MU (Page File)*	*CPUU*	
		Threads	*(milliseconds)*
YAWL	374	161	1909.8
WF	128	49	303
WF-CML	184	87	909.3

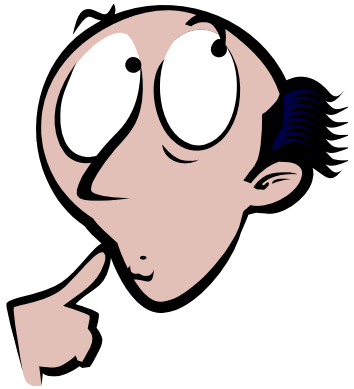
Discussions

- An initial attempt towards quantitative measurement of effort in using DSMLs
- Need more empirical studies to validate the metrics presented
  - Survey more DSMLs
  - Conduct extensive user studies for validation

Conclusion

- Investigate the mesurement of the effort to realize applications using DSMLs
- Present a classifcation of the effort and propose metrics for each category
- Multi-dimensional measurement provides a systematic and quantitative way of measuring the productivity of DSMLs

Thanks!
谢谢!
Gracias!
Merci!



So which language requires less effort using the metrics presented?