

27.01.2014	Advanced Software Engineering University of Vienna Forschungsgruppe Software Architecture	
<i>Kennzahl</i>	<i>Matrikelnummer</i>	<i>FAMILIENNAME</i> <i>Vorname</i>

6	12	12	12	12	6
1	2	3	4	5	6

Σ 60

Bitte lesen Sie diesen Abschnitt. Sie haben **60** Minuten Zeit die Fragen zu beantworten.

Hinweise:

- Füllen Sie Kennzahl, Matrikelnummer, Familienname und Vorname zuerst aus.
 - Während der Prüfung sind keinerlei Unterlagen erlaubt!
 - Technische Hilfsmittel wie Übersetzungscomputer, Taschenrechner, Mobiltelefone, etc. sind nicht erlaubt!
 - Wenn Sie Probleme beim Verstehen einer Frage haben, fragen Sie.
 - Sie können die Fragen sowohl auf Deutsch als auch auf Englisch beantworten.
 - Schreiben Sie mit dokumentenechten Schreibutensilien (kein Bleistift).
 - Sie können den Appendix (letzte Seite) von der Prüfung abtrennen.
 - Viel Erfolg!
-

Task 1 / Aufgabe 1 (6 Points / Punkte)

Explain the difference between external and embedded DSLs! Describe their advantages and disadvantages. Which alternatives or variants are possible?

Erklären Sie den Unterschied zwischen externen und eingebetteten DSLs! Beschreiben Sie ihre Vor- und Nachteile. Welche Alternativen oder Varianten sind möglich?

Task 2 / Aufgabe 2 (12 Points / Punkte)

VisTrails is an open-source scientific workflow and provenance management system that provides support for simulations, data exploration and visualization.

Document the architectural design decision that is described in the text below using the (simplified) AD template from Tyree / Ackermann.

VisTrails ist ein Open Source System zum Management von wissenschaftlichen Arbeitsabläufen das Simulationen, Visualisierungen und die Exploration von Daten unterstützt.

Dokumentieren Sie die Architectural Decision, die im Text unterhalb beschrieben wurde, mittels der (vereinfachten) Vorlage zur Dokumentation von Ads von Tyree / Ackermann.

Passing Data as Modules

One peculiar feature of VisTrails modules and their communication is that the data that is passed between VisTrails modules are themselves VisTrails modules. In VisTrails, there is a single hierarchy for module and data classes. For example, a module can provide itself as an output of a computation (and, in fact, every module provides a default "self" output port). The main disadvantage is the loss of conceptual separation between computation and data that is sometimes seen in dataflow-based architectures. There are, however, two big advantages. The first is that this closely mimics the object type systems of Java and C++, and the choice was not accidental: it was very important for us to support automatic wrapping of large class libraries such as VTK. These libraries allow objects to produce other objects as computational results, making a wrapping that distinguishes between computation and data more complicated.

The second advantage this decision brings is that defining constant values and user-settable parameters in workflows becomes easier and more uniformly integrated with the rest of the system. Consider, for example, a workflow that loads a file from a location on the Web specified by a constant. This is currently specified by a GUI in which the URL can be specified as a parameter. A natural modification of this workflow is to use it to fetch a URL that is computed somewhere upstream. We would like the rest of the workflow to change as little as possible. By assuming modules can output themselves, we can simply connect a string with the right value to the port corresponding to the parameter. Since the output of a constant evaluates to itself, the behavior is exactly the same as if the value had actually been specified as a constant.

AD shortname		AD name	
Problem Statement			
Decision drivers			
Alternatives			
Recommendation			
Decision Outcomes			
Decision Outcomes			
	Status		
	Chosen alternative		
	Justification		
	Consequences		
	Assumptions		

Task 3 / Aufgabe 3 (12 Points / Punkte)

- a) Name and explain the steps / domain engineering artifacts that are necessary to develop a Software Product Line!
 - b) Name and explain the application engineering artifacts that are necessary to develop a product in Software Product Line!
-
- a) Welche Schritte / Domain Engineering Artefakte sind zur Entwicklung einer Software Product Line notwendig? Beschreiben Sie diese im Detail!
 - b) Nennen und erklären Sie die Schritte / Artefakte die notwendig sind um im Rahmen einer Software Product Line ein Produkt zu entwickeln!

Task 4 / Aufgabe 4 (12 Points / Punkte)

- a) What is software architecture evolution and why is it relevant?
 - b) Name and explain the problems in software architecture evolution!
 - c) How can the problems in software architecture evolution be tackled?
-
- a) Erklären Sie den Begriff “Software Architecture Evolution”! Warum ist Software Architecture Evolution wichtig?
 - b) Nennen und beschreiben Sie die Probleme bei der Evolution von Software Architekturen auftreten!
 - c) Wie können die bei der Software Architektur Evolution auftretenden Probleme bekämpft werden?

Task 5 / Aufgabe 5 (12 Points / Punkte)

Create a valid instance of the XText grammar for the domain specific language that is described below. Use each grammar rule at least once.

For your convenience, you will find the XText grammar at the Appendix as well, which can be separated from this exam.

Erstellen Sie eine Instanz der folgenden Grammatik einer domänenspezifischen Sprache.

Verwenden Sie jede Grammatikregel mindestens einmal.

Zur leichteren Handhabung finden Sie die Code-Fragmente im Anhang, den Sie von dieser Prüfung abtrennen können.

```
grammar org.xtext.example.mydsl.FormModeler with org.eclipse.xtext.common.Terminals

generate formModeler "http://www.xtext.org/example/mydsl/FormModeler"

Form: 'Form' name = ID 'contains [' (section += Section | action += Action)*']';

Action:
    'Action' name = ID ":"
    'endpoint' endpoint = STRING
    'affecting' element = ID
    'method' ('send' | 'receive');

Section:
    'Section' name = ID ':'
    (item += Item)*;

Item: (Section | Label | AbstractInputField);

AbstractInputField:
    (TextField | PasswordField | DateTime | MultiOptionFields)
    ('labeled by' lab = Label)?;

MultiOptionFields:
    (ComboBox | CheckBox | List)
    'with entries ['
    (options += STRING)*
    ']';

List:
    'List' name = ID;

ComboBox:
    'ComboBox' name = ID;

CheckBox:
    'CheckBox' name = ID;

DateTime:
    'DateTime' name = ID;

PasswordField:
    'Password' name = ID;

TextField:
    'TextField' name = ID
    isMultiline?='multiline';

Label:
    'Label' name = ID
    'value' val = STRING;
```


Task 6 / Aufgabe 6 (6 Points / Punkte)

What are the criteria that have to be considered when choosing between a scripting language and a system programming language for a task at hand? Explain why each criteria will cause the use of a scripting language or a system programming language.

Welche Kriterien sind zu beachten, wenn Sie sich zur Entwicklung eines Programmes zwischen einer Skriptsprache und einer Systemprogrammiersprache entscheiden müssen? Erklären Sie für jedes Kriterium, warum dieses zum Einsatz einer Skriptsprache oder einer Systemprogrammiersprache führt.

Appendix

Task 5

XText Grammar

```
grammar org.xtext.example.mydsl.FormModeler with org.eclipse.xtext.common.Terminals
generate formModeler "http://www.xtext.org/example/mydsl/FormModeler"

Form: 'Form' name = ID 'contains [' (section += Section | action += Action)*']';

Action:
    'Action' name = ID ":"
    'endpoint' endpoint = STRING
    'affecting' element = ID
    'method' ('send' | 'receive');

Section:
    'Section' name = ID ':'
    (item += Item)*;

Item: (Section | Label | AbstractInputField);

AbstractInputField:
    (TextField | PasswordField | Date | Time | DateTime | MultiOptionFields)
    ('labeled by' lab = Label)?;

MultiOptionFields:
    (ComboBox | CheckBox | RadioButton | List)
    'with entries ['
    (options += STRING)*
    ']';

List:
    'List' name = ID;

ComboBox:
    'ComboBox' name = ID;

CheckBox:
    'CheckBox' name = ID;

RadioButton:
    'RadioButton' name = ID;

Time:
    'Time' name = ID;

Date:
    'Date' name = ID;

DateTime:
    'DateTime' name = ID;

PasswordField:
    'Password' name = ID;

TextField:
    'TextField' name = ID
    isMultiline?='multiline';

Label:
    'Label' name = ID
    'value' val = STRING;
```