

SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	Babeş Bolyai University	
1.2 Faculty	Faculty of Mathematics and Computer Science	
1.3 Department	Department of Computer Science	
1.4 Field of study	Computer Science	
1.5 Study cycle	Master	
1.6 Study programme / Qualification	Inginerie software	

2. Information regarding the discipline

2.1 Name of the discipline	Service oriented architecture						
2.2 Course coordinator	Lect. dr. Ioan Lazar						
2.3 Seminar coordinator	Lect. dr. Ioan Lazar						
2.4. Year of study	2	2.5 Semester		2.6. Type of evaluation	E	2.7 Type of discipline	

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	2
3.4 Total hours in the curriculum	48	Of which: 3.5 course	24	3.6 seminar/laboratory	24
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					20
Additional documentation (in libraries, on electronic platforms, field documentation)					10
Preparation for seminars/labs, homework, papers, portfolios and essays					23
Tutorship					7
Evaluations					20
Other activities:					
3.7 Total individual study hours	80				
3.8 Total hours per semester	128				
3.9 Number of ECTS credits	8				

4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> • Software Engineering
4.2. competencies	<ul style="list-style-type: none"> • UML basic knowledge

5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> • -
5.2. for the seminar /lab activities	<ul style="list-style-type: none"> • Laboratory with computers; high level programming language environment & CASE tools

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • Design techniques for service oriented applications. • Knowledge of important SOA frameworks
Transversal competencies	<ul style="list-style-type: none"> • Ability to apply SOA techniques to different real life problems • Improved programming & designing abilities

7. Objectives of the discipline (outcome of the acquired competencies)

7.2 Specific objective of the discipline	<p>At the completion of this course, the students must:</p> <ul style="list-style-type: none"> • have systematic knowledge on service oriented architecture • know basic elements of SoaML, BMM, and BPMN modeling languages • know basic patterns used in modeling SOA based systems • know basic elements of some concrete application frameworks for building services
--	---

8. Content

8.1 Course	Teaching methods	Remarks
1. RESTful services <ul style="list-style-type: none"> • Introduction to RESTful services • Java frameworks: JAX-RS, Spring/Jersey • Testing tools 	Exposure: description, explanation, examples, discussion of case studies	
2. RESTful services <ul style="list-style-type: none"> • Python frameworks for REST services • Testing tools 	Exposure: description, explanation, examples, discussion of case studies	
3. JavaScript services <ul style="list-style-type: none"> • Node.js • Modules • Testing tools 	Exposure: description, explanation, examples, discussion of case studies	
4. JavaScript services <ul style="list-style-type: none"> • Rhino • Comparison of JavaScript frameworks 	Exposure: description, explanation, examples, discussion of case studies	
5. RPC Services <ul style="list-style-type: none"> • Protocol Buffers • RPC 3rd party libraries 	Exposure: description, explanation, examples, discussion of case studies	
6. RPC Services <ul style="list-style-type: none"> • Apache Thrift • Comparison of RPC frameworks 	Exposure: description, explanation, examples, discussion of case studies	
7. Enterprise SOA <ul style="list-style-type: none"> • Service Component Architecture 	Exposure: description, explanation, examples, discussion of case studies	

8. Enterprise SOA <ul style="list-style-type: none"> • Web Services • WS-BPEL • BPEL4People 	Exposure: description, explanation, examples, discussion of case studies	
9. Enterprise Integration <ul style="list-style-type: none"> • Enterprise Integration Patterns • Enterprise Service Bus 	Exposure: description, explanation, examples, discussion of case studies	
10. SOA Patterns <ul style="list-style-type: none"> • Inventory, Definition • Implementation, Composition • Messaging 	Exposure: description, explanation, examples, discussion of case studies	
11. SoaML <ul style="list-style-type: none"> • UML profile for SOA • Service classification 	Exposure: description, explanation, examples, discussion of case studies	
12. SoaML and other models <ul style="list-style-type: none"> • SoaML and BMM integration • SoaML and BPMN integration 	Exposure: description, explanation, examples, discussion of case studies	

Bibliography

OMG Service oriented architecture Modeling Language (SoaML), 2009.

<http://www.omg.org/spec/SoaML/1.0/>

OASIS. Service Component Architecture (SCA), 2007. <http://www.oasis-open.org/sca>

OASIS. Web Services Business Process Execution Language Version 2.0, 2007. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

OASIS. BPEL for People, 2010. <http://www.oasis-open.org/committees/bpel4people/charter.php>

Martin Fowler. Domain Specific Languages. Addison-Wesley, 2010.

OMG Model-Driven Architecture, 2003. <http://www.omg.org/cgi-bin/doc?omg/03-06-01>

OMG Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0, 2008.

<http://www.omg.org/spec/SPEM/2.0/>

Apache Org. Injected POJO (iPOJO), 2008. <http://felix.apache.org/site/apache-felix-ipojo.html>

OMG Business Motivation Model (BMM) Version 1.1, 2010. <http://www.omg.org/spec/BMM/1.1/>

***. SOA Patterns, 2010. <http://www.soapatterns.org/>

8.2 Seminar / laboratory	Teaching methods	Remarks
1. RESTful services	Dialogue, debate, case studies, examples, proofs	
2. RESTful services	Dialogue, debate, case studies, examples, proofs	
3. JavaScript services	Dialogue, debate, case studies, examples, proofs	
4. JavaScript services	Dialogue, debate, case studies, examples, proofs	
5. RPC Services	Dialogue, debate, case studies, examples, proofs	
6. RPC Services	Dialogue, debate, case studies, examples, proofs	
7. Enterprise SOA	Dialogue, debate, case studies, examples, proofs	
8. Enterprise SOA	Dialogue, debate, case studies, examples, proofs	

9. Enterprise Integration	Dialogue, debate, case studies, examples, proofs	
10. SOA Patterns	Dialogue, debate, case studies, examples, proofs	
11. SoaML	Dialogue, debate, case studies, examples, proofs	
12. SoaML and other models	Dialogue, debate, case studies, examples, proofs	
Bibliography OMG Service oriented architecture Modeling Language (SoaML), 2009. http://www.omg.org/spec/SoaML/1.0/ OASIS. Service Component Architecture (SCA), 2007. http://www.oasis-open.org/sca OASIS. Web Services Business Process Execution Language Version 2.0, 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html OASIS. BPEL for People, 2010. http://www.oasis-open.org/committees/bpel4people/charter.php Martin Fowler. Domain Specific Languages. Addison-Wesley, 2010. OMG Model-Driven Architecture, 2003. http://www.omg.org/cgi-bin/doc?omg/03-06-01 OMG Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0, 2008. http://www.omg.org/spec/SPEM/2.0/ Apache Org. Injected POJO (iPOJO), 2008. http://felix.apache.org/site/apache-felix-ipojo.html OMG Business Motivation Model (BMM) Version 1.1, 2010. http://www.omg.org/spec/BMM/1.1/ ***. SOA Patterns, 2010. http://www.soapatterns.org/		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- | |
|--|
| <ul style="list-style-type: none"> • The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies; • The course exists in the studying program of all major universities in Romania and abroad; • The content of the course is considered the software companies as important for advanced programming skills. |
|--|

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	- know the basic principle of the domain; - apply the course concepts - problem solving	Written exam	50%
10.5 Seminar/lab activities	- be able to implement course concepts and algorithms - apply techniques for different classes of programming languages	-Practical examination -documentation -portofolio -continous observations	50%

10.6 Minimum performance standards

➤ At least grade 5 (from a scale of 1 to 10) at both written exam and laboratory work.

Date

30.04.14

Signature of course coordinator

Lect. dr. Ioan Lazar

Signature of seminar coordinator

Lect. dr. Ioan Lazar

Date of approval

Signature of the head of department

Prof. dr. Bazil Parv