

ERRATA LIST

Page number	ERRATA
37	<p>Second paragraph in subsection 3.1.3.2: A space character between “and” and “port types” is inserted.</p>
50	<p>Subsection 3.2.4: A new sentence discussing model transformations and a new citation to a new article is added.</p> <p><i>A concrete example of model refinement is a transformation of service models into executable state machines, which is also a model, is presented in [54].</i></p> <p>The new cited article is:</p> <p><i>[54] F.A. Kraemer, P. Herrmann. Transforming Collaborative Service Specifications into Efficiently Executable State Machines. In Electronic Communications of the EASST 6(2007).</i></p>
52	<p>Subsection 3.2.5: A new sentence states that a code generation method that aligns UML 2.0 and Temporal logic is added. For this, a new citation is also added at then end of the sentence.</p> <p><i>An example of code generation method that aligns UML state machines and Temporal logic to generate code from service models is introduced in [55].</i></p> <p>The new cited article is:</p> <p><i>[55] F.A. Kraemer, P. Herrmann, R. Bræk. Aligning UML 2.0 State Machines and Temporal Logic for the Efficient Execution of Services. In R. Meersmann and Z. Tari (eds.), Proceedings of the 8th International Symposium on Distributed Objects and Applications (DOA06), pages 1613-1632, LNCS 4276, Montpellier, Springer-Verlag 2006.</i></p>
54	<p>Subsection 3.3: A new sentence and its reference with regard to an example of method for developing embedded applications are added.</p> <p><i>In [56], reusable building blocks are used to construct embedded applications. A dedicated building block provides a mechanism to access platform-specific functionality of Sun SPOTs devices. The building blocks can be used in combination with other blocks realizing other functionalities such as communication protocols.</i></p> <p>The new cited article is:</p> <p><i>[56] F.A. Kraemer, V. Slåtten, P. Herrmann. Model-Driven Construction of Embedded Applications based on Reusable Building Blocks – An Example. In R. Reed, A. Bilgic, R. 3 Gotzhein (eds.), Proceedings of the 14th International SDL Forum 2009, pages 1-18, Bochum, LNCS 5719, Springer-Verlag, September 2009.</i></p>

58	<p>Subsection 3.5: Instead of article 65 (old version), two newest versions of the article are cited. The new cited articles are:</p> <p><i>[53] F.A. Kraemer, R. Bræk, P. Herrmann. Compositional Service Engineering with Arctis. In Teletronikk, Special Issue on Model-Driven Security - Integrating Availability in System Development, Telenor, (2009)1, 135-151</i></p> <p><i>[57] F.A. Kraemer, V. Slåtten, P. Herrmann. Tool Support for the Rapid Composition, Analysis and Implementation of Reactive Services. In The Journal of Systems and Software 82 (2009) 2068-2080.</i></p>
66	<p>Second paragraph: A new citation is added at the end of sentence:</p> <p><i>Software developers can use ARCTIS to specify, analyze, and verify software system using models[57].</i></p> <p>The cited article is:</p> <p><i>[57] F.A. Kraemer, V. Slåtten, P. Herrmann. Tool Support for the Rapid Composition, Analysis and Implementation of Reactive Services. In The Journal of Systems and Software 82 (2009) 2068-2080</i></p>
67	<p>First paragraph: The sentence “<i>Similarly, ARCTIS building block is also a model as it is instance of ARCTIS</i>” is edited as</p> <p><i>Similarly, an ARCTIS building block is also a model as it is an instance of ARCTIS that is based on UML 2.0 [53,57,65].</i></p>
80	<p>Section 4.5.2: A short comparison of PMG-pro and ARCTIS code generation is added. For this, article number 55 is cited.</p> <p><i>With regard to the ARCTIS code generation, PMG-pro code generator applies only a simple transformation process. It reads and converts directly activity nodes (with control and object flows) into code. So, it works only on a simple activity diagram due to the formalism. The reason is that the code generator in PMG-pro is developed only for the purpose of proof-of-concept. However, since the method is independent of languages and tools, PMG-pro can use any built-in code generator. In contrast, ARCTIS converts a collaboration activity diagram into a state machine which is executable. For this, cTLA is used for the reasoning [55]. Code is then, generated from the generated state machines. It works on more complex activity diagrams.</i></p>
92	<p>Section 5.4, forth paragraph: The article number 55 is also cited at the end of the sentence “<i>In the case of ARCTIS ... is instantiated</i>”.</p> <p><i>In the case of ARCTIS to Java code generation, from each building block, one object is instantiated [55].</i></p>
105	

	<p>Subsection 5.5.2.4, second paragraph: The article number 55 is also cited again at the end of the sentence “ARCTIS has a code generator for J2SE ... model”.</p> <p><i>ARCTIS has a code generator for J2SE that enables automatic code generation from ARCTIS model [55].</i></p>
106	<p>Subsection 5.5.3.3: The misspelled title is corrected. The correct one is <i>Use Case Scenario</i>.</p>
123	<p>Subsection 6.3.2: To make it clear what is discussed, the title is changed. The new title is:</p> <p>6.3.2 Discussion</p>
124	<p>Subsection 6.3.2, first paragraph: A short comparison to article number 38 is added.</p> <p><i>Model-driven development is considered effective if the transformation of abstract models to more detailed models is an automatic process. In [38], a method for an automatic transformation of flow-global choreography models (UML activity diagram) into localized choreography models (ARCTIS model) is proposed. The author proposes to use Attributed Graph Grammar System (AGG) as the graph transformation engine. Using this method, ARCTIS code generator is used to generate an executable code. In contrast, PMG-pro implements only a simple interpreter that reads and converts any connection of activity nodes (in the activity diagrams) into code, directly. This is of course a weak approach. However, it works for a simple activity diagram, where the automation of code generation can be achieved. For more complex activity diagram, PMG-pro can use built-in code generator</i></p> <p>The new cited article is:</p> <p><i>[38] Han F., S.B. Kathayat, Hien L., R. Bræk, P. Herrmann, Towards Choreography Model Transformation via Graph Transformation, in Proceedings of the 2nd IEEE International Conference on Software Engineering and Service Science (ICSESS 2011), pages 508-515, Beijing, IEEE Computer Society Press, July 2011.</i></p>
135	<p>References: The list of the references is updated due to the new added six citations and one citation is removed.</p>