Advanced Topics in Software Architecture
Lectures and Literature

Sune Chung Jepsen
Torben Worm

Tuesday 22nd August, 2023

Course Plan

This is the course plan for the course "Advanced Topics in Software Architecture". The course will use articles from journals and book chapters that will be added during the course. This is a preliminary plan, and it will be updated as the course proceeds. The most recent plan can always be found on itslearning in the "General Course Information" topic. Please refer to the course description for further information.

Week	Lec.	Part	Content	Literature
35	1	Introduction	1. Introduction 2. Software architecture and technology overview 3. Working with a complex system	[13, 14, 15, 16], Chapter 1, 2, 3 [24] Optional: [17, 19]
36	2	Tools and technologies 1	1. Software interfaces 2. Documenting software interfaces	Chapter 15 [4], Chapter 7 (attention on 7.3) [7]
37	3	Tools and technologies 2	1. Programming languages 2. Database systems	[21, 12], Optional: [23]
38	4	Tools and technologies 3	1. Message busses 2. Containerization tools	[20], Chapter 14 [6], [1]
39	5	Software architecture patterns 1	Overview of the pattern language, e.g. layered view, data flow view, data-centered view etc.	[3, 5]
40	6	Software architecture patterns 2	Creational, structural, behavioral patterns	81-87, 137-139, 221-223, 331-337 [11], [9]
41	7	Quality attributes 1	Stability and availability	Chapter [18] TBA
42-43	No lecture			
44	8	Quality attributes 2	1. I4.0 quality attribute model 2. Multiple levels of interoperability	[2, 22]
45	9	Software architecture patterns 3	Architecture and agile practices	[26]

Continued on next page

Table 1 – continued from previous page

Week	Lec.	Part	Content	Literature
46	10	Formal software architecture	Introduction to architecture descriptions languages	[8], Appendix C [7], Optional [10]
47	11	Evaluating software architecture	Design experiment and evaluation	Chapter 6 [25]
48	12	Wrapping up	Summary	

References

[1] Charles Anderson. Docker [software engineering]. *IEEE Software*, 32(3):102–c3, 2015.

[2] Pablo Oliveira Antonino, Rafael Capilla, Patrizio Pelliccione, Frank Schnicke, Daniel Espen, Thomas Kuhn, and Klaus Schmid. A quality 4.0 model for architecting industry 4.0 systems. *Advanced Engineering Informatics*, 54:101801, 2022.

[3] Paris Avgeriou and Uwe Zdun. Architectural patterns revisited - A pattern language. In Andy Longshaw and Uwe Zdun, editors, *EuroPLoP' 2005, Tenth European Conference on Pattern Languages of Programs, Irsee, Germany, July 6-10, 2005*, pages 431–470. UVK - Universitaetsverlag Konstanz, 2005.

[4] L. Bass, P. Clements, R. Kazman, and an O'Reilly Media Company Safari. *Software Architecture in Practice, 4th Edition*. SEI series in software engineering. Addison-Wesley Professional, 2021.

[5] Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt. Past, present, and future trends in software patterns. *IEEE Software*, 24(4):31–37, 2007.

[6] Emiliano Casalicchio. *Container Orchestration: A Survey*, pages 221–235. Springer International Publishing, Cham, 2019.

[7] Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, M. Little, Paulo Merson, Robert Nord, and Judith Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley Professional, 2nd edition, 2010.

[8] P.C. Clements. A survey of architecture description languages. In *Proceedings of the 8th International Workshop on Software Specification and Design*, pages 16–25, 1996.

[9] João Tiago Duarte Maia and Filipe Figueiredo Correia. Service mesh patterns. In *Proceedings of the 27th European Conference on Pattern Languages of Programs*, EuroPLop '22, New York, NY, USA, 2023. Association for Computing Machinery.

[10] Peter Feiler, David Gluch, and John Hudak. The architecture analysis & design language (aadl): An introduction. Technical Report CMU/SEI-2006-TN-011, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2006.

[11] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition, 1994.

[12] Jing Han, Haihong E, Guan Le, and Jian Du. Survey on nosql database. In *2011 6th International Conference on Pervasive Computing and Applications*, pages 363–366, 2011.

[13] Sune Chung Jepsen, Thomas Ingemann Mørk, Jakob Hviid, and Torben Worm. A pilot study of industry 4.0 asset interoperability challenges in an industry 4.0 laboratory. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 571–575, 2020.

[14] Sune Chung Jepsen, Torben Worm, Aslak Johansen, Sanja Lazarova-Molnar, Mikkel Baun Kjærgaard, Eun-Young Kang, Jonas Friederich, Juan Esteban Heredia Mena, Reza Soltani, Sune Lundø Sørensen, and Jens Hjort Schwee. A research setup demonstrating flexible industry 4.0 production. In *2021 International Symposium ELMAR*, pages 143–150, Zadar, Croatia, 2021. IEEE.

[15] Sune Chung Jepsen, Torben Worm, Thomas Ingemann Mørk, and Jakob Hviid. An analysis of asset interoperability for i4.0 middleware. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 707–710, New York, NY, USA, 2021. Association for Computing Machinery.

[16] Sune Chung Jepsen, Torben Worm, Thomas Ingemann Mørk, and Jakob Hviid. Industry 4.0 middleware software architecture interoperability analysis. In *2021 IEEE/ACM 3rd International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT)*, pages 32–35, Madrid, Spain, 2021. IEEE.

[17] P. Kruchten, H. Obbink, and J. Stafford. The past, present, and future for software architecture. *IEEE Software*, 23(2):22–30, 2006.

[18] Michael T. Nygard. *Release It! Design and Deploy Production-Ready Software*. Pragmatic Bookshelf, Raleigh, NC, 2 edition, 2018.

[19] M. Shaw and P. Clements. The golden age of software architecture. *IEEE Software*, 23(2):31–39, 2006.

[20] P. Sommer, F. Schellroth, M. Fischer, and J. Schlechtendahl. Message-oriented middleware for industrial production systems. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 1217–1223, 2018.

[21] Walter F. Tichy. Programming-in-the-large: Past, present, and future. In *Proceedings of the 14th International Conference on Software Engineering*, ICSE '92, page 362–367, New York, NY, USA, 1992. Association for Computing Machinery.

[22] Andreas Tolk, Saikou Y Diallo, and Charles D Turnitsa. Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. *Journal of Systems, Cybernetics, and Informatics*, 5(5), 2007.

[23] A.I. Wasserman. Software tools: past, present, and future. In *Proceedings International Conference on Software Methods and Tools. SMT 2000*, pages 3–6, 2000.

[24] Anthony Weston. *A rulebook for arguments*. Hackett Publishing, 5th edition, 2017.

[25] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[26] Chen Yang, Peng Liang, and Paris Avgeriou. A systematic mapping study on the combination of software architecture and agile development. *Journal of Systems and Software*, 111:157–184, 2016.