Lab 7: Big Picture and Transition

Table of Content

Context and Motivation . . . 1
Learning Objectives . . . 2
Steps Overview . . . 2
Starting Position (Baseline, Initial Position; dt. “Ausgangslage”) . . . 2
Step 1: Recap of Labs 1 to 6 . . . 2
Step 2: From OO analysis and domain design to application architecture . . . 3
Step 3: From CoMo to software engineering practices, object-oriented programming . . . 5
Step 4: From CoMo to Business Process Modeling (BPM) . . . 6
Step 5: From CoMo To Database Modeling and Management . . . 6
Step 6: Application to larger projects and term theses (SA, BA) . . . 6
Step 7: Alternatives and Outlook . . . 6
Research and development at OST in the topic areas of the CoMo Learning Lab. . . . 7
Summary and Conclusions . . . 7
Concepts Revisited . . . 7
Reflection and Call to Action . . . 7
Repetition Questions . . . 8
More Information . . . 8
Frequently Asked Questions (FAQ) . . . 8
References . . . 9

Context and Motivation

We have seen six CoMo practices in action so far, and looked at the transition to Object-Oriented and Domain-Driven Design (OOD, DDD), and Object-Oriented Programming (OOP) as well. This seventh and final CoMo lab takes a step back to return to the big picture and connect CoMo with other topics (taught in other modules at OST):

- Why bother about collaboration and models in analysis and design?
- How to the lab parts relate to each other, when to use which CoMo practice?
- How to make “these things” (domain storytelling, event storming, story maps) my/your own?
- Which other modules/courses continue the analysis, design and development journey we have embarked on?
- How to map the concepts from the CoMo lab to the topics in those other courses?

Note (to students, to instructors): Not all modules mentioned in this lab are available in all "Studiengänge". Moreover, existing modules with matching (or overlapping) content might have different objectives and content (e.g., compare Object-Oriented Programming (OOP) in "Informatik" and in "Wirtschaftsinformatik"). They keep on evolving too.

Learning Objectives

Having completed this lab, participants are able to:

- Compare and select different modeling approaches.
- Be able to map CoMo Concepts to those in other courses/modules.
- Discuss the pros and cons, the motivation, the pitfalls with peers, with other stakeholders.

Steps Overview

This lab exercise has the following steps:

1. Recap of Labs 1 to 6
2. From OO analysis and domain design to application architecture
3. From CoMo to software engineering practices, object-oriented programming
4. From CoMo to Business Process Modeling (BPM)
5. From CoMo To Database Modeling and Management
6. Application to larger projects and term theses (SA, BA)
7. Alternatives and outlook
8. Discuss role of Artificial Intelligence (AI) in CoMo, today and tomorrow.

Starting Position (Baseline, Initial Position; dt. "Ausgangslage")

We have elaborated the product vision in six steps. Let's recapitulate and discuss what comes next in this final CoMo Lab 7.

Step 1: Recap of Labs 1 to 6

Task: Study the Figure "CoMo Learning Lab: Seven Labs, Product Vision to Big Picture Modeling" in the Module Overview.

Hints:

- Domain Storytelling concepts are introduced in "Quick-Start Guide: What you need to know about Domain Storytelling to get started.".
- See "EventStorming Glossary & Cheat sheet" for terminology and concept synopsis.

Questions:

1. Which of the steps in the lab overview figure did you perform?
2. Do you agree on the interfaces between the steps?
3. What's next, where do the various outputs of the steps go?

Step 2: From OO analysis and domain design to application architecture

Task: Map the CoMo Concepts to software/application architecture concepts (Fowler 2002).

Hints: If you have not yet taken any architecture course, or do not remember the coverage of the topic in those (or other) courses, feel free to review the sample solution straight away.

You may want to provide your answers in table form (Table 1, Table 2, Table 3, Table 4). Markdown sources for these four tables are available upon request.

Table 1 is the template for Domain Storytelling mappings.

Table 1: Domain Storytelling concepts and practices to be mapped and leveraged

CoMo Concept	Analog	Usage	Comments
Domain Storytelling			
Actor	…	…	…
Work object	…	…	…
Activity	…	…	…

Table 2 is the template for EventStorming mappings.

Table 2: EventStorming concepts and practices to be mapped and leveraged

CoMo Concept	Analog	Usage	Comments
EventStorming			
Domain event	…	…	…
Actor/agent	…	…	…
System	…	…	…
Pivotal event	…	…	…
Swim lanes	…	…	separated by pivotal events
Policy	…	….	reaction that says “whenever X happens, we do Y”
Command/action	…	…	…
Read model/views and forms	…	…	aka query model/information in ddd-crew glossary

Table 3 is the template for user story related mappings:

Table 3: User story mapping/splitting concepts and practices to be mapped and leveraged

CoMo Concept	Analog	Usage	Comments
User Story (Role-Feature-Benefit template)			
Role	…	…	…
Feature	…	…	…
Benefit	…	…	…
Story Mapping			
Horizontal axis of story map	…	…	…
Vertical axis of story map	…	…	…

CoMo Concept	Analog	Usage	Comments
Story Splitting			
Splitting patterns: workflow	…	…	…
Splitting patterns: operations	…	…	…
Splitting patterns: business rule	…	…	…
Splitting patterns: user input channel	…	…	…
Splitting patterns: data variation	…	…	…
Remaining four splitting patterns	…	…	…

Finally, Table 4 is the template for VDAD mappings:

Table 4: VDAD concepts and practices to be mapped and leveraged

CoMo Concept	Analog	Usage	Comments
Value-Oriented Analysis and Design (VDAD)			
Stakeholder	…	…	…
Value-impact map	…	…	…

Step 3: From CoMo to software engineering practices, object-oriented programming

Task: Perform the concept mapping outlined in Step 2 of this lab, this time for general software engineering and OOP rather than application architecture.

> *Hints:* There is no need to be complete or agree on correctness here; the lab step is designed to provide an outlook on the transition; multiple approaches exist and are practiced. What works in one project content and for one team is not necessarily suited well elsewhere. You might want to provide a few answers and then read through the sample solution for inspiration.

Questions:

1. How does Domain Storytelling help with the elicitation of Architecturally Significant Requirements (ASRs)?

2. Which EventStorming concepts may yield Non-Functional Requirements (NFRs)? Can we expect them to be SMART already (specific, measurable, agreed upon, realistic, time-bound)?
3. Can story splitting be used for component identification?

Step 4: From CoMo to Business Process Modeling (BPM)

Task: a) Map the CoMo Concepts to those in Business Process Modeling (BPM) and, specifically, language concepts in BPMN. Use the same four tables as in Step 2 and Step 3. b) Create a BPMN model/diagram that represents the event storming output, the domain event timeline in particular.

> *Hints:* The level of precision and rigor varies deliberately. We do not expect you to produce executable BPMN models here. That said, Context Mapper is able to generate Sketch Miner scripts that in turn can be transformed into BPMN.

It is possible to turn event storming results into BPMN process models; Context Mapper and Sketch miner can support such effort. See https://contextmapper.org/docs/application-and-process-layer/ and https://contextmapper.org/docs/bpmn-sketch-miner/ for more information.

Step 5: From CoMo To Database Modeling and Management

Task: Perform a concept mapping again, this time for data modeling and database design. To do so, please fill out the four tables proposed in Step 2 (Markdown source available upon request).

> *Hints:* This task and its sample solution are suggestive not normative; just like the ones in the previous steps, they report on heuristics that work in certain contexts.

Step 6: Application to larger projects and term theses (SA, BA)

Task: Apply selected concepts from the CoMo Learning Lab to a completed or ongoing project of yours, for instance engineering project, term projects (“Studienarbeit”) and bachelor theses (SA, BA) (in computer science “Studiengang”).

Step 7: Alternatives and Outlook

Task: Think about and/or research alternatives to the presented practices.

If you study part-time and are involved with design and development (software or other products): how are production vision, end user requirements, system dynamics, business rules and their variations (or other constraints) elicited and elaborated upon on your projects, in the organizations you work for? How do these practice sand notations compare to those presented in the CoMo LL?

> *Hints:* You can perform this task in small groups (self organized), as all other ones in the CoMo LL. Feel free to ask for help too (assuming supervised learning).

Questions:

1. What are the benefits that are mentioned?
2. Are drawbacks also pointed out ("when not to use"?)
3. How does your personal analysis and design toolbox look like after the CoMo LL? Has it changed?

Research and development at OST in the topic areas of the CoMo Learning Lab.

Have a look at the IFS CAL pages, as well as those of other institutes involved with CoMo.

Summary and Conclusions

We have returned from product vision to business-level organization and enterprise-wise system/software architecture. This lab prepared the handover to other courses/modules in your "Studiengang".

Concepts Revisited

We mapped concepts from all previous labs to:

- Application architecture
- Software engineering practices, object-oriented programming
- BPM and BPMN
- Database modeling and management concepts were mapped to too

Architecture diagramming concepts and notations such as C4, UML were touched upon.

Reflection and Call to Action

Please review and recapitulate what you take away from this lab. Where and when can you apply the taught concepts?

Repetition Questions

1. Why and when should we model?
2. Why collaborate when modeling?
3. Are the CoMo practices and their combined usage featured in CoMo LL easy and fully repeatable and non-ambiguous?

More Information

Read "Critically Engaging with Models" and/or consult the Architecture Modeling activity in DPR (Zimmermann and Stocker 2024).

Vaughn Vernon's books are good reads for those of you with a deeper interest in DDD and its architectural relevance (Vernon 2013),(Vernon and Jaskula 2022).

arc42 and Q42 are go-to-places when it comes to crafting and documenting software architectures, beginning with NFRs.

Frequently Asked Questions (FAQ)

- *Where can I find more information on decision making and recording? Do the software architecture template also work for business decisions, implementation decisions, evaluation decisions?*
  Answer: As long as you make a decision in a context that has positive and negative consequences and addresses certain conflicting desired quality properties, templates such as Y-statements work well. For more information and discussion, see "ADR = Any Decision Record? Architecture, Design and Beyond".
- *How does CoMo relate to (or differ from) data modeling, Data Flow Diagrams in particular?*
  Answer: "A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself." (source: Wikipedia). Both domain stories and event storming outputs identify data that flows through a process or systems; they indicate the timing but make the flow less explicit than DFDs; hence the CoMo practices are well-suited as input to DFD creation. For more information on DFDs and their roots, please refer to https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/, https://www.ibm.com/think/topics/data-flow-diagram, https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/.
- *What is Enterprise Architecture Management (EAM)?*
  Answer: EAM operates on the city planning level whereas software architecture pertains to the building level. ArchiMate is a rather rich language supported by tools such as Archi, TOGAF a

framework in use in large companies. John Zachman and his framework have their role to play in EAM too.

- *What is Event-Driven Architecture (EDA)? What is Event Sourcing? What is Command Query Responsibility Segregation (CQRS)? How to these concepts and terms relate to CoMo, EventStorming in particular?*
  Answer: See https://contextmapper.org/docs/event-sourcing-and-cqrs-modeling/ and https://interface-refactoring.github.io/refactorings/segregatecommandsfromqueries for background information. While an analysis-level focus on events might suggest that the EDA style and related patterns (e.g., event sourcing, CQRS) are a natural choice, this is neither sufficient nor necessary to justify such big architectural decisions; many other decision drives and ASRs exist.
- *What about interdisciplinary projects, which tools to use there?*
  Answer: Check out https://transdisciplinarity.ch and the toolbox "Methods and tools for co-producing knowledge".
- *What is the role of Artificial Intelligence (AI) today and tomorrow in the CoMo context?*
  Answer: See the blog post "Ten Resources about AI in Software Engineering You Do Not Want to Miss" for general pointers. We see AI tools in assisting roles, for instance to summarize or convert workshop results and intermediate/draft models; prompt quality matters and results have to be reviewed carefully. Time will tell whether the current excitement and momentum sustains; chances are that the next trend bandwagons are already preparing to get going. Have a look at this article for background information "The Effects of Hype in the Software Domain: Causes, Consequences, and Mitigations" (Broy and Selić 2025).

References

Broy, Manfred, and Bran Selić. 2025. "The Effects of Hype in the Software Domain: Causes, Consequences, and Mitigations." *IEEE Software* 42 (2): 98–102. https://doi.org/10.1109/MS.2024.3511732.

Fowler, Martin. 2002. *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Vernon, Vaughn. 2013. *Implementing Domain-Driven Design*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Vernon, Vaughn, and Tomasz Jaskula. 2022. *Strategic Monoliths and Microservices: Driving Innovation Using Purposeful Architecture*. Addison-Wesley Signature Series (Vernon). Addison-Wesley Professional.

Zimmermann, Olaf, and Mirko Stocker. 2024. *Design Practice Reference - Guides and Templates to Craft Quality Software in Style*. online: LeanPub. https://leanpub.com/dpr.