



## SS2018 – Model-driven Software Development in Technical Spaces

# Template Meta Programming with Beta

Professor: Prof. Dr. Uwe Aßmann

Tutor: Dr.-Ing. Thomas Kühn

### Task 1 Template Meta Programming with Beta

**BETA**<sup>1</sup> is an experimental language from Mjølner Informatics in Denmark that runs on the Mjølner runtime environment [1]. In BETA any component is a fragment that can be composed with other fragments. The execution of a BETA program revolves around instantiating, matching, and composing patterns. To better understand BETA, read Chapters 3–5 of “*Object-Oriented Programming in the BETA Programming Language*” [1]. Moreover, BETA comes with a powerful fragment language that allows arbitrary parts of a BETA program to be split out into separate files. The connection between such fragments is maintained through slots marking up certain non-terminals and declaring that those need to be filled with concrete code before execution of the program (cf. [1, Chapter 17 and 20]).

- a) What is the basic construct of BETA?
- b) How are objects and operations represented using this construct?
- c) How does fragment composition work in BETA?
- d) Describe BETA’s *component model*, *composition technique* and *composition language*.

---

<sup>1</sup><http://www.daimi.au.dk/~beta>

## Task 2 Programming with BETA

In 1972, David Parnas proposed the following problem [2]:

*The Key Word In Context (KWIC) index system accepts an ordered set of lines; each line is an ordered set of words, and each word is an ordered set of characters. Any line may be “circularly shifted” by repeatedly removing the first word and appending it at the end of the line. The KWIC index system outputs a listing of all circular shifts of all lines in alphabetical order.*

Within this exercise, you will be tasked to implement the main modules of the KWIC components within the **BETA** language, i.e., input, cycle, and output, and use the **Mjølnner** runtime environment to run the KWIC application.

- a) Download and install the **Mjølnner** runtime environment.
- b) Work through some of the tutorials. In particular, *MIA 99-40* and *MIA 94-24* (to get a feeling for OO development in BETA and for using the **Mjølnner** environment).
- c) Design the KWIC index system by implementing each *component* in the *do*-part of a separate pattern. Use **attribute**-definitions and **enter**- and **exit**-statements and assignments to represent the application’s architecture.
- d) Refactor your KWIC implementation, such that the main program fragment should only specify the architecture of your application, while the implementations should be split up into individual fragments’ *do*-part.

Hand in your solution as archive \*.zip before the next exercise.

### Additional Information

- **Mjølnner** runtime environment<sup>2</sup>
- **BETA** language reference manual<sup>3</sup>
- Tutorial on the use of **Mjølnner** (MIA 99-40)<sup>4</sup>
- Tutorial on fragment composition (MIA 94-24)<sup>5</sup>

## References

- [1] Ole Lehrmann Madsen, Birger Møller-Pedersen, and Kristen Nygaard. *Object-oriented programming in the BETA programming language*. Addison-Wesley, 1993. URL <http://www.daimi.au.dk/~beta/Books/index.html#betabook.download>. Reprinted by **Mjølnner** Informatics with permission from Addison-Wesley.
- [2] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Commun. ACM*, 15(12):1053–1058, 1972. ISSN 0001-0782. doi: 10.1145/361598.361623. URL <http://doi.acm.org/10.1145/361598.361623>.

<sup>2</sup>[http://daimi.au.dk/~beta/mjolner\\_system](http://daimi.au.dk/~beta/mjolner_system)

<sup>3</sup><http://www.cs.au.dk/~beta/Manuals/latest/beta-intro/index.html>

<sup>4</sup><http://www.cs.au.dk/~beta/Manuals/latest/mjolner-tut/index.html>

<sup>5</sup><http://www.cs.au.dk/~beta/Manuals/latest/tutorial/index.html>