# 2IO23 (DBL Software Specification and Games)

Project coordinators: Herman Haverkort
Tim Willemse
cs.Herman@Haverkort.net, T.A.C.Willemse@tue.nl

1 November 2012

## Guidelines Deliverables I

Below, we provide guidelines and criteria for writing and assessing the deliverable —the final report— for assignment I. This final report is marked with a *pass* or *fail* grade. A *pass* is a necessary condition for continuing the DBL project. The final report should contain at least clearly marked sections addressing:

1. An informal problem description, paraphrasing and analysing the assignment. Special attention should be paid to describing the ambiguities in the original problem description (if any), the way these are resolved and the impact these choices have on the final solution. This section should be written by the team members acting as the producer.

2. A formal specification of the problem description. The original problem description can be formalised by specifying the relation between input and output using a Z schema. A formal specification should be void of implementation details. Ensure that the formal specification can be understood and communicated to others by explaining it informally. This section should be written by the team members acting as the producer.

3. A written and well-motivated assessment of the formal specification. Pay particular attention to:

   - Consistency between the informal problem description and the formal specification
   - Transferability of the document: can the problem and the rationale behind its solution be understood and reconstructed on the basis of the document?
   - The quality of the formal specification. Is it clear what the pre-conditions are, and why these are the way they are? Can they be made more liberal, and, if so, why did the authors not do so? Is it clear what the post-conditions are? Can the post-condition be made stricter?

   This section should be written by the team members acting as the client. N.B.: Provide the assessment to the producer only when the producer has finished implementing the formal specification.

4. A selection of testcases. Provide a number of inputs that satisfy the preconditions of the Z schema's, and state what, according to the formal specification should be the output.

Note that any incongruences between specified output and desired output should be reflected in the assessment of the formal specification. For stating the testcases, the formal specifications are leading and not the informal description of the problem. Try to provide inputs for which you expect the *implementation* might give dubious outputs. This section should be written by the team members acting as the client.

5. An implementation. The specification may be implemented in a programming language of your liking; motivate your choice. Implement the **original** specification, handed to the client rather than a potentially improved version of the specification addressing issues raised by the client.[1] The actual code must be included in an appendix, but a section relating the code, and in particular, its interfaces, to the formal specification should be included in the main document. This section should be written by the team members acting as the producer.

6. A test report. Show the results of executing the testcases and explain whether the observed outputs are indeed in line with the specification and the informal description.

Each part of the document contributes equally to the final grade for this assignment. A *pass* is obtained when the report is graded 5.5 or higher on a scale of 1 to 10. Any grade below 5.5 will lead to a *fail*.

[1] This ensures that testcases, derived from the formal specification, are still meaningful and allow for detecting differences between the implementation and the specification (if present). Note that for assignment I, in contrast to assignment II, we do not assess the quality of the implementation.