



Preparator 2.0

Software Lab Project 2012

Students: Fatemeh Fakharian, Philipp Schneider

Supervisors: Gergő Dori, Simon Daum

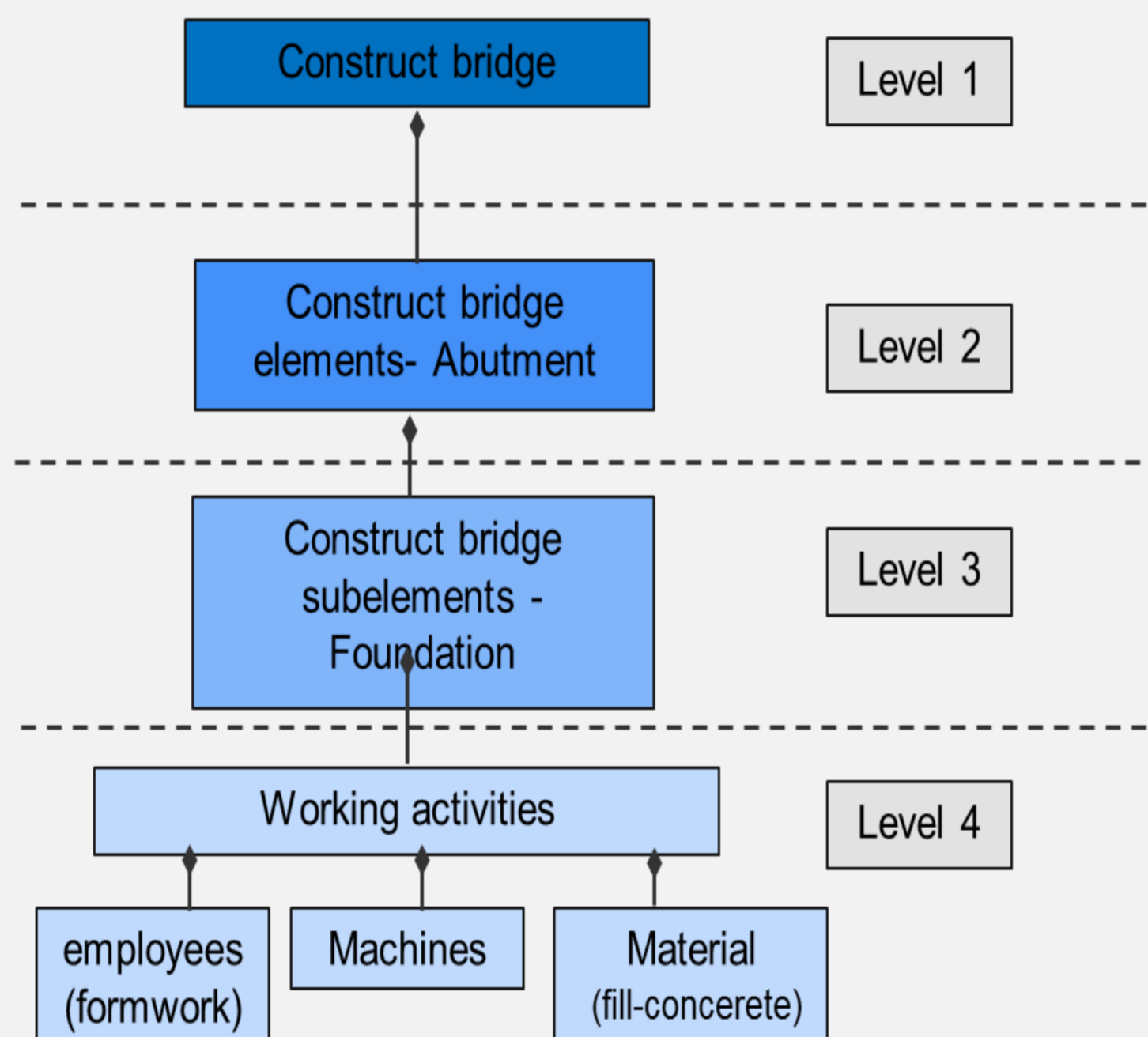
Abstract

This project was performed to bring production steps to economic order. Production chains can be set to discrete event structure to be able to apply optimization tools.

Motivation

Why production cycles should be improved

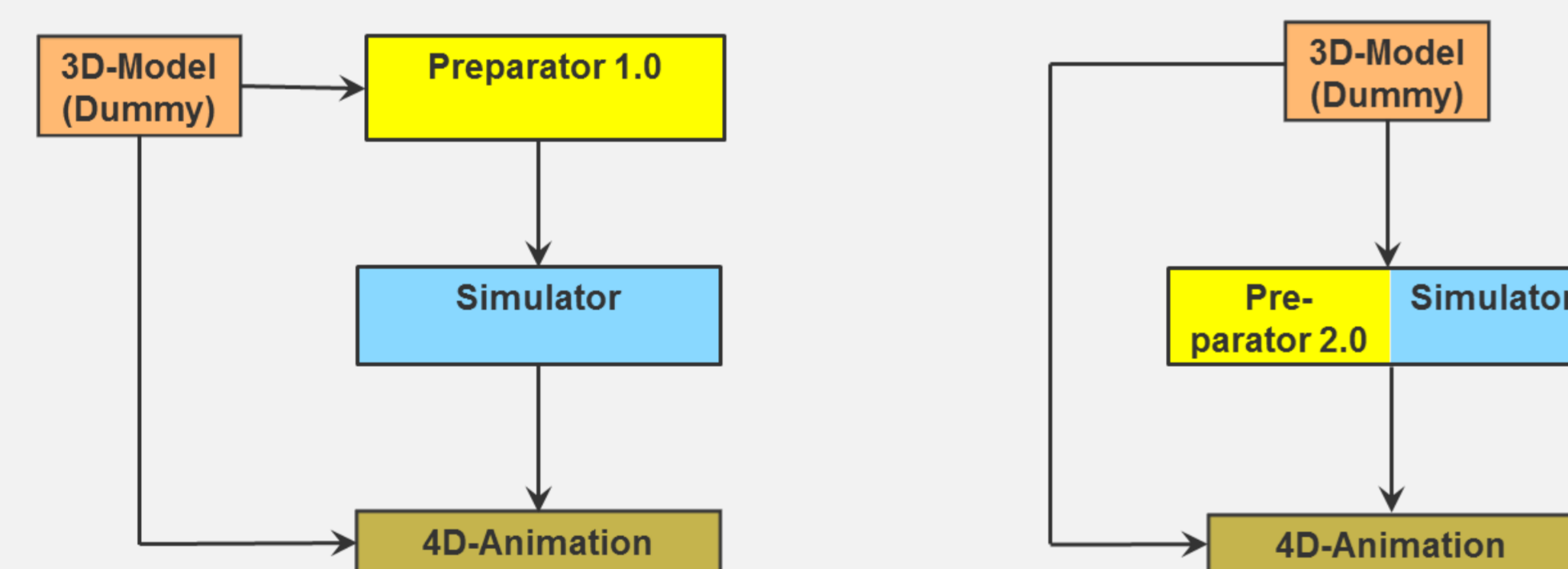
It is a tedious work to structure the order of working tasks within industrial projects. This results because of the high number of different tasks and their technological dependencies (e.g. precedence relationships, resources) such a project consists of. Often, the planning of work cycling steps is performed by the construction site manager by estimating the facts that influence the construction situation; this is – of course – combined with one's experience. By observing such work flow situations with respect to mathematical relations one can find out that there are possibilities to sum tasks to special classes ordered in a hierarchical structure according to a level of detail approach. The smallest element of the structure are called activities. They describe atomic construction tasks of the project. If certain attributes (material amount, quantity of workers, ...) are assigned to an activity, it becomes a work package. Several work packages can be united to one superior class, which represents here one physical item (i.e. a foundation of a bridge pier) called subelement.



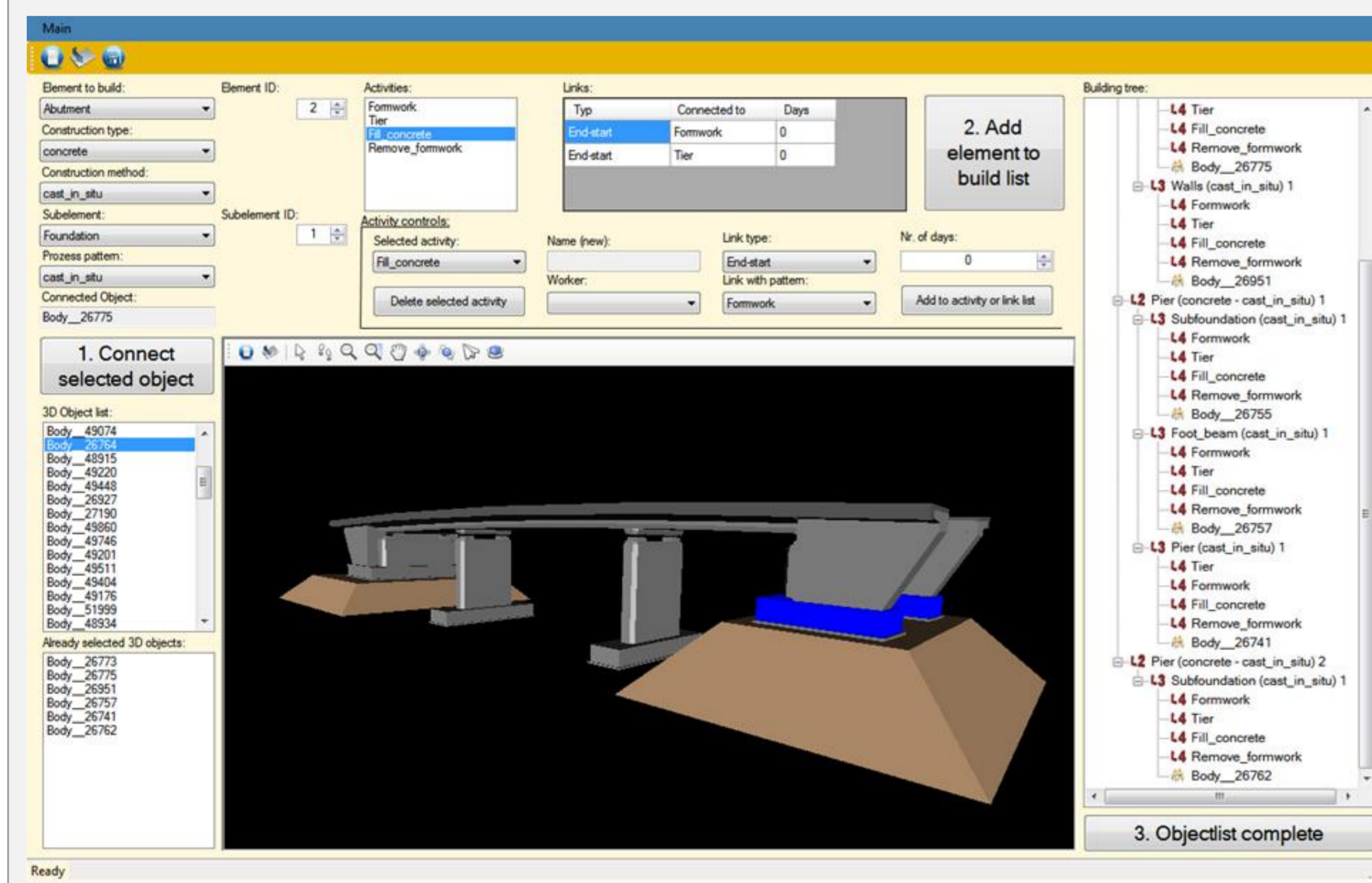
Once again, several subelements are to be unified to create super classes, called elements. An example element could be one complete bridge pier, consisting of the subelements pier foundation, pier shank and upper roller supports. So this pattern can be performed further such that on the highest level there is only one class left where all other classes are derived from. This could be here the whole building like a bridge.

Another point of view is that the complete building is regarded as a group of work packages. This makes sense if the problem shall be solved by mathematical methods.

Every work package is clearly defined by its task, location, mass, precedence constraints and required resources. In order to generate a feasible schedule from this work packages a simulation toll is used. In order to run the simulation properly, all the necessary above mentioned data for the work packages have to be predefined.



At the beginning of the project the core structure for these tasks existed already (module called "Preparator 1.0"). With this, one is able to structure all tasks according to their level. Also, all the tasks can be shared to activities. The scope of the project is to implement the Preparator 1.0 code in JAVA language to achieve the final goal that is merging Preparator 2.0 with the simulator engine. This is a tool that is able to generate a feasible schedule with respect to the precedence and resource constraints of all construction task.



Conclusions / Outlook

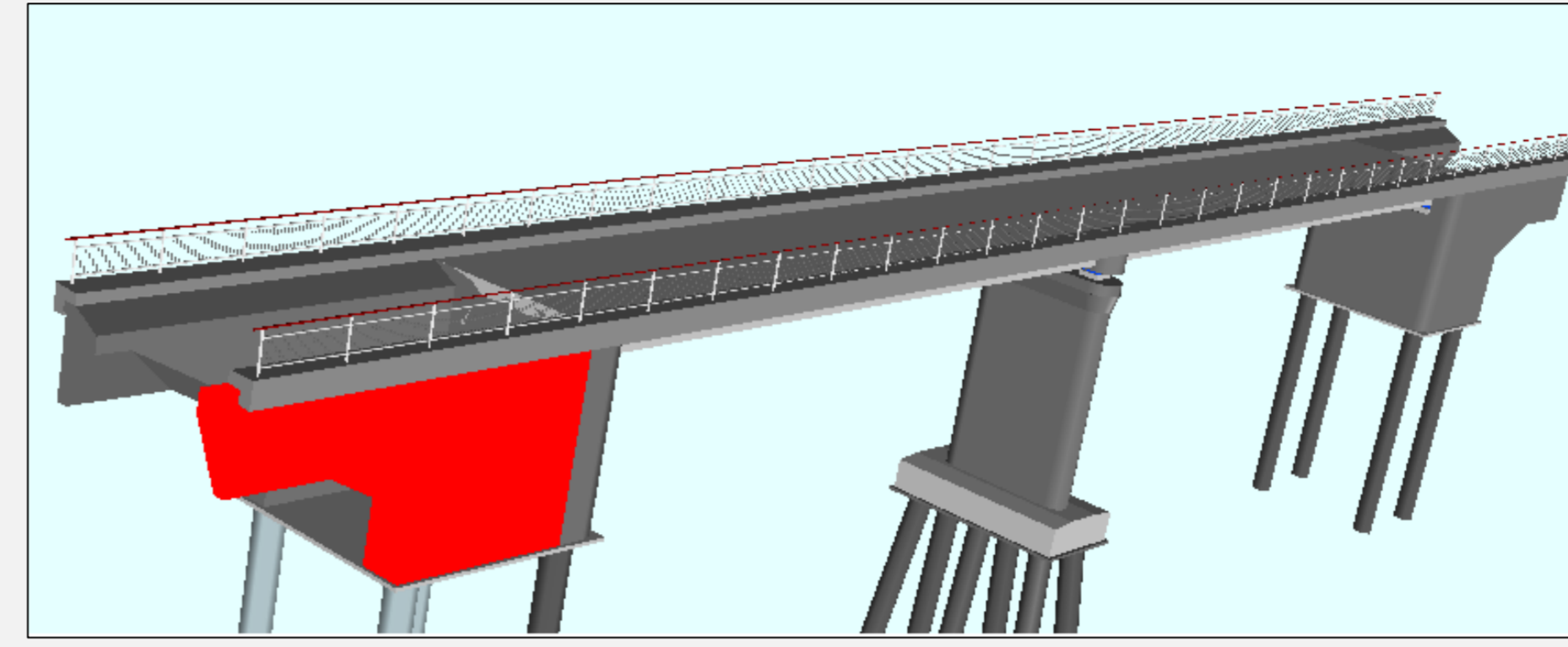
The newly developed Preparator 2.0 is able to create a hierarchical structure made of activities that are necessary to erect the complete building. It is able to determine geometrical properties of 3D objects, which can be connected and stored within a work package.

Up to now, certain important steps for performing the simulation have been accomplished.

As further steps, functionalities could be extended: First, more geometric properties of the bodies should be derived by the CAD input data (like volume and center of mass). Also one should be able to create an interface that can define precedence relationship and constraints.

Geometrical Properties

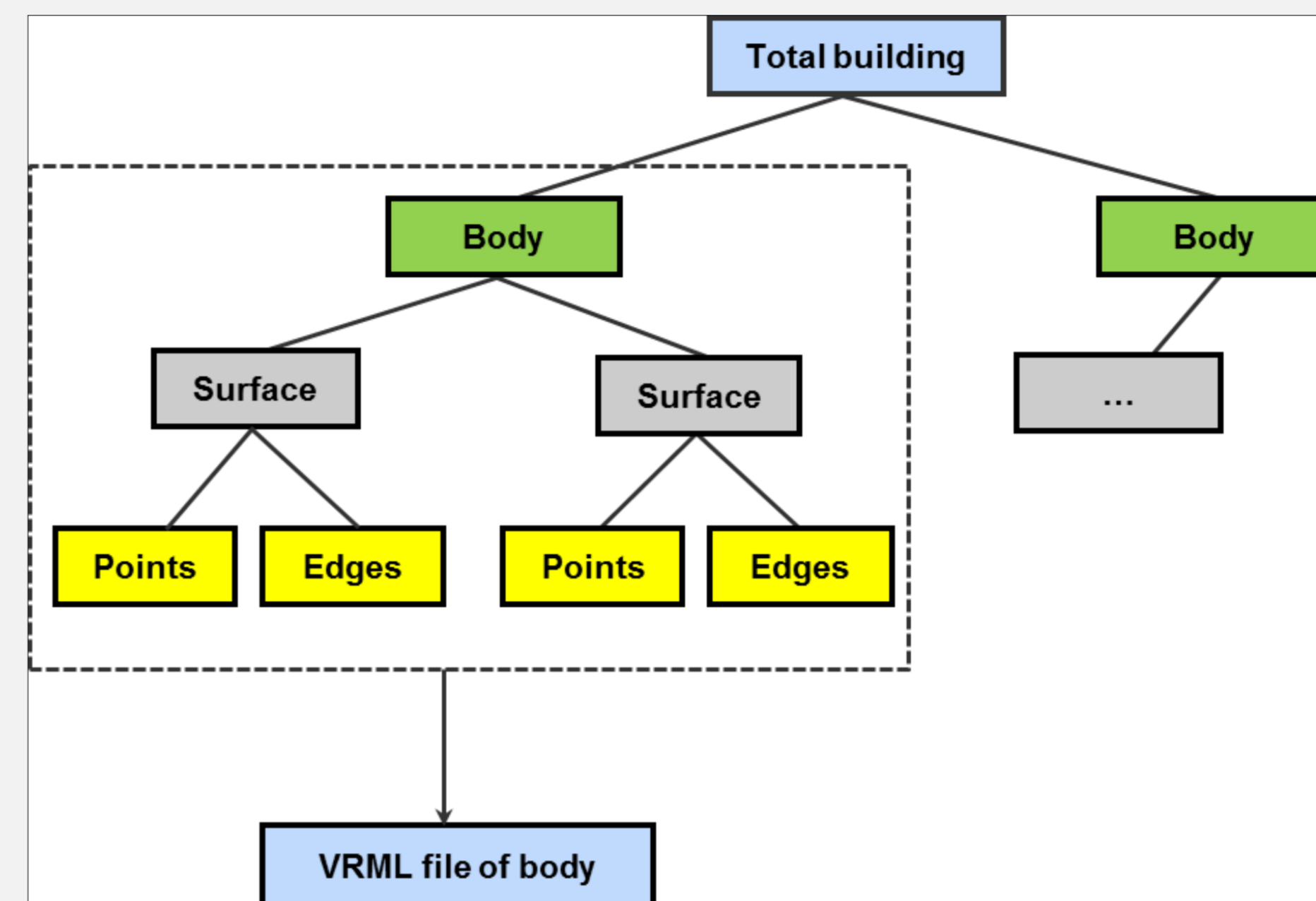
In order to determine geometrical properties (e.g. surface, volume etc.) for the construction tasks a connection of the corresponding work package to a 3D object is necessary. Geometry is read into Preparator via CAD-format, in our case via VRML. With this file, the complete building is visualized. Controlling and manipulating the model is performed via mouse steering.



VRML represents the geometry in a hierarchically built structure consisting of points and edges (first level), surfaces (second level), bodies (third level) and complete building structures (fourth level).

The code has been developed such that a mouse picking process delivers the underlying body to the user, which corresponds to the subelement level of the construction task hierarchy.

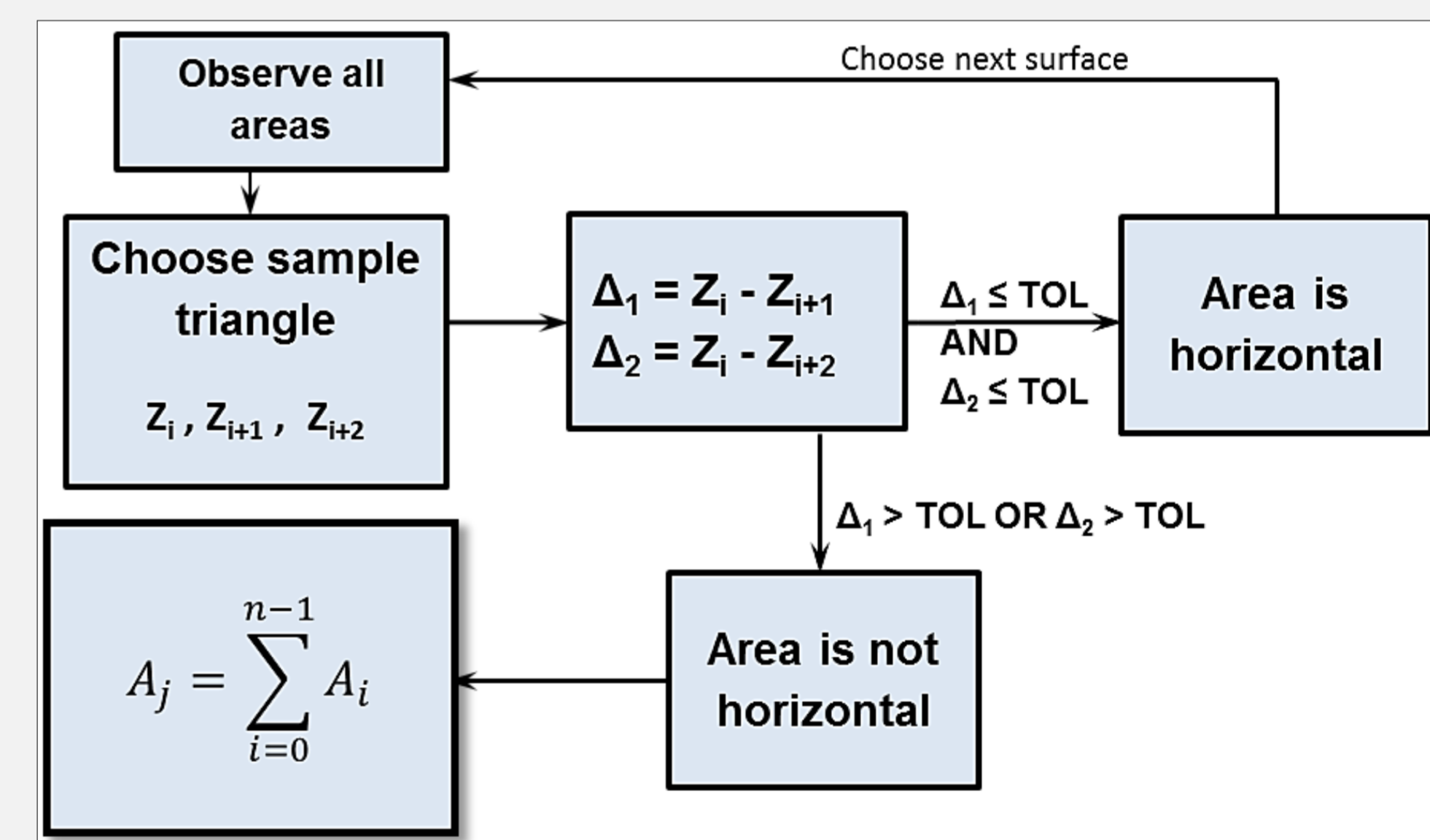
Normally, JAVA3D provides only an element of the surface level by a picking process. Therefore, a special method was included that gets the superior class of the picked surface, and determines which object it belongs to. After this, one is able to determine the geometrical properties of a single selected body and by connecting it to a work packages save these data as the attributes of the activity.



In order to calculate the geometrical properties for the work packages the program reads all the edges and points the picked body consists of. However, because of one body consists of several surfaces, the program reads all points and all edge definitions of all the surfaces. Once, one area has been recorded, the next surface is searched and all entities read. The procedure is performed sequentially till every surface within the body is stored.

Calculating non-horizontal area for formwork structures:

An algorithm has been implemented that is able to compute the area of all surfaces a body consists of which are not in horizontal position. Out of this ability, one is able to obtain the size of the requested formwork for concrete objects.



Achievements

Parallel to using the geometrical data of the bridge, predefined process patterns are used for different construction methods in order to accelerate the designing process of the work packages. This information are provided by XML files. To create a level of detail oriented structure of the construction tasks the above mentioned four level of classes such as „Bridge“, „Element“, „Sub-Element“ and „Processes“ are implemented.

After parsing the XML data the user can select one of the hierarchical ordered predefined work package and connect it to a 3D object of the 3D model of the construction project, so that the identification, localization and calculation of geometrical properties for the work package is achievable.

After the implementation of the predefined process patterns to a new hierarchical class structure, a GUI is made to provide a user interface dealing with the different elements, sub-elements and activities. Also, the user can create precedence relationships between activities defining a sequence between them.

When all the necessary precedence relationships and resource constraints of the activities of a subelement are created, the element can be added to the building tree of the bridge, which represents all the necessary construction tasks to erect the bridge.

Activities:	Links:
I	Type: End-Start, Days: 4, Linked: B
J	Type: Start-start, Days: 4, Linked: G
K	Type: End-Start, Days: 12, Linked: I
L	

PreParator 2.0

Element Name: Pier Element ID: 0

Construction type: cast_in_situ SubElement ID: 0

Construction Method: concrete

Sub-Element: Pier Selected N-H Surface Area: 16.074207473727768 m²

Process pattern:

References

- [1] G. Dori, A. Borrmann; Automatic Generation of Complex Bridge Construction Animation Sections by Coupling Constraint-based Discrete-event Simulation with Game Engines; Proc. of CONVR - International Conference on Construction Applications of Virtual Reality, Weimar, Germany, 2011
- [2] G. Dori, A. Borrmann; Determination of float time for individual construction tasks using constraint-based discrete-event simulation; Proceedings in Winter Simulation, Conference 2012
- [3] I-Chen Wu, Andre Borrmann, Ulrike Beißert, Markus König, Ernst Rank; Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation.; Advanced Engineering Informatics, V 24, pp(379-388); 2010.
- [4] I.C. Wu, A. Borrmann, E. Rank, U. Beißert, M. König; A Patter-Based Approach for Facilitating Schedule Generation and Cost Analysis in Bridge Construction Projects.
- [5] <http://docs.oracle.com/javase/> state: November 01-12
- [6] <http://www.java3d.org/> state: November 01-12