Medical Simulation Markup Language

A tutorial introduction

Nicolai Schoch * Markus Stoll † Stefan Suwelack ‡
Alexander Weigl §

November 7, 2014

Abstract

1 Introduction

We first mentioned the idea from an unified and mighty workflow system for biomechanical engineering in [1]. In the meantime we realize the Medical Simulation Markup Language (MSML).

You can see MSML as a build tool, like *GNU Make*, except the offered operations are settled in the pre- and postprocessing of three dimensional data after the biomechanical simulation. The simulation is the heart of every workflow and is outsourced to modern simulation frameworks, like SOFA[1] or Hiflow3[2], and MSML cares about the messy details of the simulation framework. You get an unified interface.

Content In this paper we show the various power of MSML in different scenarios (sections **??**). Before the tutorials we create a view on MSML in section 2. This chapter defines the ground terms and nomenclatures in MSML. Openness is one power of MSML. You can extend it in various kinds, such topics are discovered in section 5.

Acknowledgement MSML was developed in the SFB TRR 125 Cognition Guided Surgery project, funded by the DFG.[3]

*nicolai.schoch@iwr.uni-heidelberg.de IWR EMCL
†m.stoll@dkfz-heidelberg.de, DKFZ
‡suwelack@kit.edu, Humanoids and Intelligence Lab, Karlsruher Institute for Technology
§Alexander.Weigl@student.kit.edu, Humanoids and Intelligence Lab, Karlsruher Institute for Technology

[1] http://www.sofa-framework.org/
[2] http://hiflow3.org
[3] *weigl:* add english name here

2 MSML Architecture

We start with the terms, give an overview of the pipeline and end this section with the MSMLFile datastructure.

Terms An *Operator* is a function, that can be used within the workflow. The function can be written in C/C++ or Python. We maintain additional meta data of the arguments and output in a separate XML file. If an Operator is used, it get instantiated with the given references to other instances or variables. We call instances of Operator a *Task*. *Elements* define additional features, like materials or constraints, to your objects in the scene. The *Alphabet* contains both, Elements and Operators, with their XML meta data. You can add new Operators or Elements to MSML by creating new Alphabet entries. The simulation is outsourced to several simulation frameworks. The interface between MSML and the frameworks is done within several specific *Exporter*. Currently we support Sofa, Hiflow, FeBio and Abaqus.

Pipeline Figure 1 gives an overview about the process. MSML takes a data structure as shown in figure 2. The data structure can allocated from XML or with Python. Defining in Python offers more control over the pipeline and is more flexible, as you can use parameterized the creation. XML is easy, understandable but more fixed.

The first step analyze the data structure for finding errors and creating a build graph. The build graph is a directed acyclic graph[4],

Data structure The MSMLFile gather all information for making a simulation. It contains the Workflow, that contains the Task to be executed, the environment with parameters for the simulation framework, variables, and scene objects. Every scene object is a node in the simulation and have constraints, material regions or output.

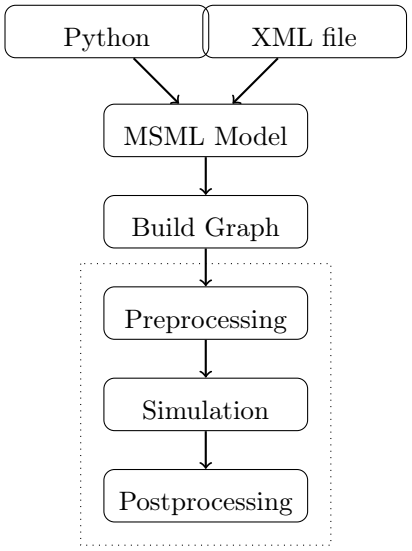

Figure 1: *MSML Pipeline*

3 Everyone has the bunny, we too.

4 Have someone say: Optimization?

5 Extend MSML

Operator

Element

Exporter

[4] it has to be

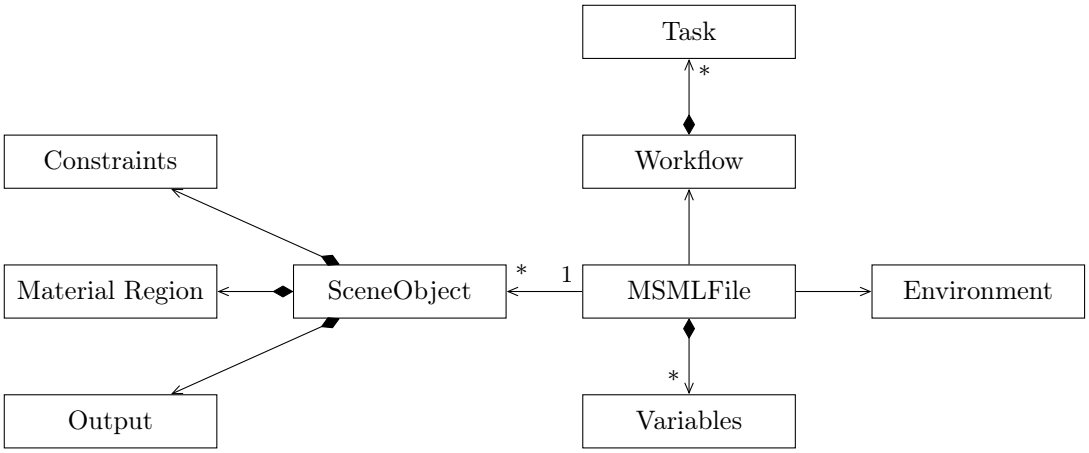

Figure 2: *MSML Model*

References

[1] Stefan Suwelack, Markus Stoll, Sebastian Schalck, Nicolai Schoch, Rüdiger Dillmann, Rolf Berndl, Vincent Heuveline, and Stefanie Speidel. The medical simulation markup language - simplifying the biomechanical modeling workflow. *Studies in Health Technology and Informatics*, 196:396–400, April 2014.