Static Dimensional Analysis in Modelica

David Broman
Linköping University
davbr@ida.liu.se

1. Research Project Description

Physical dimensions and units are normally used in science and engineering when modeling complex physical systems. When models are implemented in programming languages, physical quantities are typically implemented as real numbers with no notation of the units or dimensions. Due to the fact that programming languages generally lack this support, units and dimensional analysis cannot be used to detect potential errors caused by mismatch of dimensions.

While such errors might seem insignificant, unit mismatches are not uncommon and can lead to catastrophic safety consequences. For example, in September 1999, the NASA Mars Climate Orbiter Mission lost contact with the spacecraft during the Mars orbit maneuver. This failure was eventually traced back to a software flaw when converting between English and metric units [4].

There are many attempts of formulating dimension checking in programming languages, such as Ada Package [2] and Template based C++ approaches [5]. However, these solutions have the characteristics of ad hoc language features, rather than being part of a language's general type system. Recently, approaches have been investigated in how to incorporate dimension and unit analysis in object-oriented languages, such as Java and C#, based on nominal type systems [1].

Dimensions and units are today part of the Modelica language specification [3], which enables the engineer to specify units during modeling. However, it exists no sound strategy of how to analyze and verify dimensional integrity in arbitrary language constructs of Modelica.

The research problem is to investigate if and how dimension and unit checking can be enforced statically, i.e. before runtime, in a language such as Modelica, which has a structural type system with the unusual property of being a declarative equation based object-oriented language that combines side-effect free functions with imperative algorithm sections.

Our expected contribution is twofold: i) A sound design proposal of how the Modelica language can be extended to support full static checking of dimensions and units, and ii) a formal and proven type system which guarantees safety. In addition to theoretical investigations of the proposed research solution, a full scale solution will be implemented in the OpenModelica environment, to enable evaluation and verification. Hence, the aim of the project is to provide concrete and useful results, which can be of direct practical usage in the commercial industry.

References