An advice on megamodeling

Ralf Lämmel, Software Languages Team, University of Koblenz-Landau

16 April 2016

Objective

One wants to model a scenario of technology and language usage.

Tasks to accomplish

- Give a *name* to the “scenario”.
- Identify relevant *documentation*.
- Identify *illustrations* (e.g., source code or diagrams).
- *Extract megamodel elements*:
  - Entity types
  - Entities
  - Relationships
- Verify the ('notational') *well-formedness* of the megamodel.
- *Link* entity types and entities to authoritative online resources, e.g., Wikipedia pages.
- Compile a *presentation* of the megamodel.

Some of the tasks are discussed in more detail below.

Strategies for the extraction of megamodel elements

One can apply **one or both** of the following strategies.

Documentation-based strategy

- Given a document (documentation), identify the most relevant sections:
  - Is a *technology* characterized in terms of classifiers or concepts?
  - Is a *data flow* (some input/output behavior) described?
  - Is an *artifact* characterized in terms of classifiers or concepts?
- Map technical terms to megamodel elements:
  - A noun could proxy for an *entity.*

    - A noun could proxy for an *entity type* used for classification.
    - A verb could proxy for a *relationship*.
- Translate informal illustrations to “formal” megamodels, to the extent possible.

Code-based strategy

- Add megamodel elements for each file of the source code distribution:
  - Add an 'element of' relationship to declare the language.
    - For instance, the file extension may hint at the language.
  - Add a 'uses', 'facilitates', or 'implements' relationship for a concept.
    - For instance, an imported package may hint at a concept.
- Based on the assumed data flow and behavior, identify transient artifacts.
  - For instance, data flow may be represented in a build script.
  - Also, any sort of service may imply requests, responses, or messages.
- Add megamodel elements for each transient:
  - Add an 'element of' relationship to declare the language.
  - Add a 'uses', 'facilitates', or 'implements' relationship for a concept.
- Go over all possible relationship types and pairs of entities.
  - Research whether the relationship holds for the entities.

Verification of the well-formedness of the megamodel

- Use only predefined entity types or newly defined subtypes ('<').
- Use only predefined relationship types or newly defined ones (if you really need to).
- All entities ('operands of relationships') have an assigned entity type (':').
- All artifact entities have an assigned language ('element of').
- Data flow relies on function applications. Functions must be introduced as follows:
  - A function may be defined or implemented by an artifact.
  - A function may also be implemented by a technology or a system.
- All relationships have operands that are of the permitted entity types for the relationship.

Compilation of a presentation of the megamodel

- Use a graphical illustration to sketch the scenario.
  - Label the figure with the name of the scenario.
  - Aim at brining up the key entities as 'nodes' in the figure.
  - Aim at hinting at the key relationships as 'edges' in the figure.
  - Make sure the notation of your figure is reasonably clear.
- Modularize the megamodel into groups of declarations, e.g.:
  - the languages involved and the decomposition of a technology;
  - a function with its application and the source and target artifacts.