TNO ESI and Océ: Architecting support using knowledge graphs and expert systems

Participants: Richard Doornbos (industry representative), Paolo di Francesco, Klaas Andries de Graaf (academic leader), Ana-Maria Oprescu, Alexander Serebrenik (academic leader), Wouter Tabingh Suermondt, Roberto Verdecchia

Context and problem

The complexity in high-tech systems is increasing rapidly, therefore development of these systems is becoming a huge challenge. In particular, development of those systems is an inherently multi-disciplinary multi-stakeholder endeavour putting increasingly high demands on system architects. To address these demands TNO-ESI has advocated *knowledge graphs* (Fig. 1) as means of supporting the decision making through externalisation of shared knowledge of the system architects and stakeholders. Application of the knowledge graphs in the industry is however challenged by the high costs associated with creating them as well as insufficient clarity on return on investment for system architects. During the ICT with Industry workshop participants have focused on addressing those challenges.

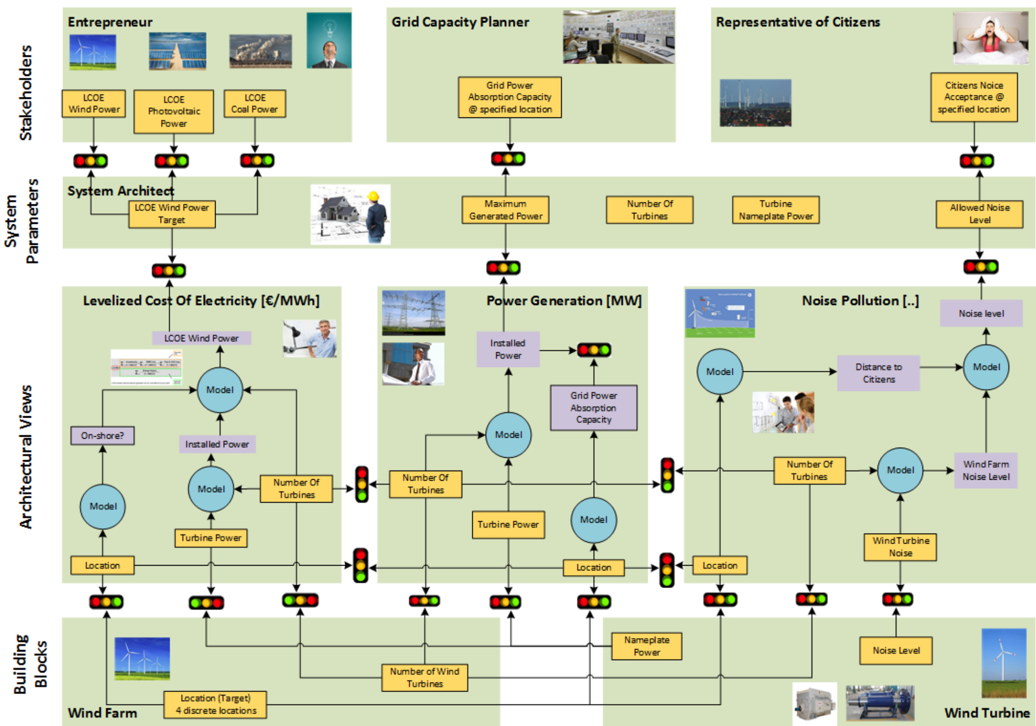

Figure 1 Knowledge graph of the windmill case

Research approach

Knowledge graph

The knowledge graph consists of responsibility domains such as shown in the Wind Farm example (Figure 1). Each of these domains involve multiple variables and relations between them. The unique feature of the knowledge graph are the *semaphores*, representing shared concerns of different domain owners or interfaces between different responsibility domains. Semaphores reflect constraints over one or more variables: the semaphore turns red when the constraint is violated (represented as traffic lights in Figure 1). For example, all stakeholders of the windmill case should agree on the location of the windmill (with the location preference of different stakeholders being variables involved).

Constructing knowledge graphs

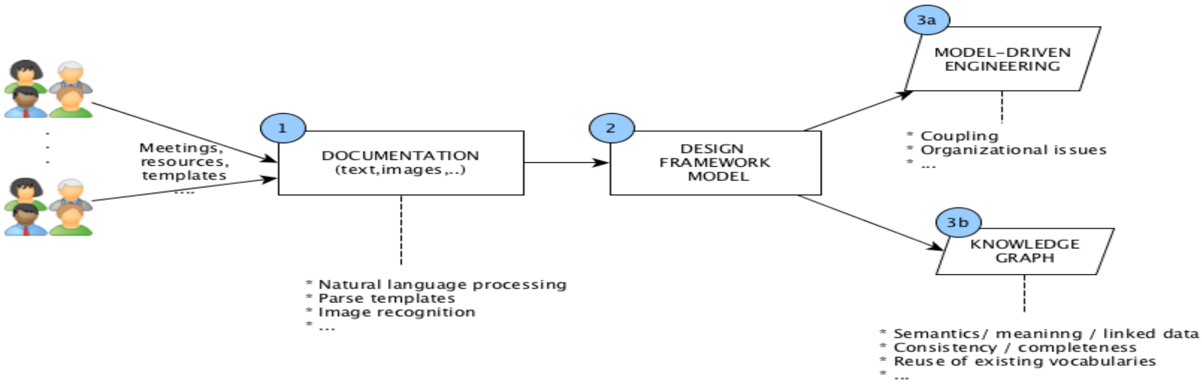

Figure 2 The process of creating a knowledge graph

As identified by TNO one of the main challenges related to adoption of the knowledge graph in the industry pertains to the effort required to construct it (see Figure 2). Indeed, architectural knowledge is often tacit and externalisation of such tacit knowledge is an important step in creation of knowledge. We propose two main lines of improving the knowledge graph construction: improving the organisation of the construction process, and automation of individual steps during this process. Improved organisation of the knowledge graph construction process can facilitate automation of the individual follow up steps, e.g., automated information extraction can benefit from standardised templates for meeting minutes and automated analysis of audio recordings---from predefined statements indicating beginning/end of the discussion phase/topic. In terms of automation of the information extraction process multiple techniques are readily available dependent on the information source: e.g., automatic document summarisation/multi-document summarisation techniques are available for textual document, while multiple research papers and patents propose summarisation of audio recordings. Applicability of these techniques to architecture information extraction should be evaluated in the follow-up study.

> In the TNO-Océ case the architects represent informal models as box-and-arrows schemes on an A3 paper. Restriction of the page size forces architects to focus on the principal elements of the solution and facilitates communication between them and the stakeholders. The box-and-arrows schemes are Microsoft Visio documents and as such can be stored as XML files amenable for further automatic processing.

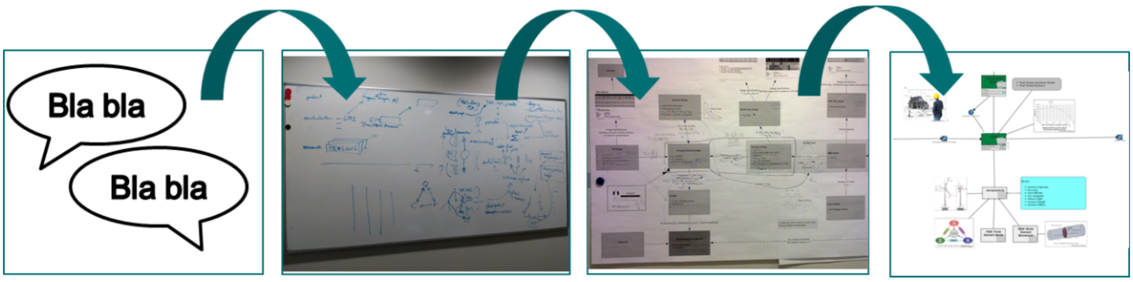

Figure 3 Early architecting process: conversations, informal modeling, A3 architecture models and knowledge graphs

Using the knowledge graphs

The knowledge graph can be used as a basis for several metrics/analyses that in their turn can be used to guide the discussions/prioritise architectural decisions. For instance, using genetic algorithms one can explore the design space and aim at finding variable assignment ensuring that all semaphores are green (all constraints are satisfied); the number of responsibility domains hosting incoming arrows per semaphore can be seen as a complexity metrics reflecting architectural hotspots; comparison of the knowledge graph with the organisational structure of the company or ecosystem of companies provides means of detecting socio-technical (in)congruence and assessing whether the responsibilities have been appropriately separated.

Most elements of the current knowledge graph constructed in the Design Framework software of TNO-ESI are of a basic 'block' type. This gives a lot of freedom in modelling different domains, process, and systems. The blocks in the DF software can be assigned labels, to make their meaning and purpose more explicit, but they do not belong to a well-defined class hierarchy and semantic relationships defined in an ontology (domain model). An ontology would improve the primary goals of the DF software, namely, communication and documentation, as it makes the input, usage, and output of the DF software more explicit and understandable by providing a common reference and understanding of the meaning of the elements, types of elements, and relationships between elements in a domain. Moreover, the instantiation of an ontology will result in a knowledge graph with meaning, which can be queried using SPARQL (the semantic web version of SQL), thereby answering relevant questions for intended users of the DF software (e.g., stakeholders, managers, and system architects) which, for example, need to know:

- Which requirements have been decided upon and are satisfied in a project?
- What is the rate of innovation in a project and what elements lead to the project goal?
- What is the separation of concerns and coupling between domains in a project?
- Is an architectural solution complete? What are the uncertainties or bottlenecks?

Representing knowledge in the DF software in an instantiated ontology (stored in e.g., RDF/TTL/OWL) also allows description of the same knowledge in the vocabulary ("jargon") of different disciplines, reuse of existing ontologies (e.g., specifying ISO standards), inference of new knowledge by logic and rules (e.g, via SWRL or class axioms), publishing the DF knowledge graph as linked data on the internet, and connecting to, referencing to, integrating and/or reasoning with existing linked data on the internet. Moreover, having a knowledge graph with semantics allows analysis and visualisation of the graph in more detail compared to only structural analysis (i.e., only nodes+edges without meaning).

Results

The main outcomes of this project are the identified knowledge graph design process (Fig. 2) and an ontology (domain model) to make the semantics of the knowledge graph explicit as well as the first steps towards automation of the information extraction and analyses supported by knowledge graphs. Furthermore, the discussions held during the workshop will serve as a basis for collaboration between industrial and academic partners. This collaboration can be supported through projects for PhD candidates, PDEng trainees or MSc/BSc students.

Future work

The following topics have been identified as topics for collaboration:

- Extraction of architecture-relevant information from technical documentation and stakeholder discussions: related research areas are text/audio summarisation, image processing, machine learning, software engineering.
- Design space exploration:
    - Automatic identification of variable assignments ensuring that all semaphores are green: possible techniques are genetic algorithms, constraint solving, linear optimisation.
    - Manual design space exploration with automated support: to obtain a better view on the design options, it is desirable not merely to generate a solution with all semaphores being green or all such solutions but a subset of such solutions that would be limited (to support manual inspection), representative and diverse enough (to make this inspection “complete” and “efficient”).
    - Change impact analysis should provide insights in the possible impact of changes of individual variables or entire responsibility domains.
- Identification of the ontology underlying the knowledge graphs: research topics include knowledge modeling and management, model-driven software engineering.
- Socio-technical analysis of knowledge graphs and their evolution: research domains relevant to this topic are social network analysis, repository mining. The outcome of this line of work should provide the answers to the management questions and architects’ questions.
- Tool support of the knowledge graph creation and use: at the moment the knowledge graph is supported by a website and is stored in an RDBMS. Richer knowledge representation, efficient data storage, improved UI and graph visualisation should be sought.