Interactive comment on "OWFgraph: A graph database for the offshore wind farm domain" by Erik Quaeghebeur et al.

Pawel Gancarski (Referee)

pgancarski@cener.com

The manuscript is a release of a graph-structured description of objects, procedures, attributes, variables, phenomenons (not done yet), models and relations between them in the offshore wind farm domain. The presented work is impressive in size (1219 nodes, 1718 edges) and has a great potential for becoming a key element in the further improvements to the maturing wind energy technology. The presentation of the idea should be improved. This is especially important when realizing the potential of the work and the fact that it will need a widespread adoption to be successful.

Issues: - Chapters 2 and 3 are 23 pages of documentation that, while very important, is mostly of use for someone working on the database. It doesn't necessarily provide an

interesting read for a broader audience. - Size of the graph should be provided in the abstract or introduction. It is important to stress out the ambition of this work early on in the paper, as skimming through the document might give a sensation of a chaotic, yet another "single-use spreadsheet". This might reduce the impact of the article with the "casual" reader. - Publishing the graph in the form of an attached .zip file is not the best way to start building a community around it. My suggestion would be to use GitHub and publish the repository via Zenodo. GitHub would work well for tracking changes proposed by the community in the .json file, while Zenodo would make it citable and would keep track of future releases (supports versioning). - While none "phenomenon" nodes have been defined yet, it would still be useful to provide some examples of what concepts you would like to see there in the future.

Comments: - The variable names don't seem to follow any existing naming convention. By defining over 800 of them it seems like an attempt to create a yet another standard, which I believe to be more harmful than helpful when it comes to developing standards and work on tool interoperability. This could be addressed by defining a new relation for variables "same_as". The mapping between naming conventions could be stored in a separate .json file (see next point), for optional use. - Dividing the .json file into smaller ones would make it easier to work with and to trace changes. The division could be made by discipline, or it could separate attachments to the core graph which might not be of interest for the whole community (institution specific nodes like "ECN install"). - APPEARS_IN relation feels like an unnecessary replacement of input and output relations, probably created for compatibility with the visualization software. If possible, I would consider using separate input and output relations or renaming, for clarity APPEARS_IN to INPUT_OUTPUT_TO.