W3C LBD Community Group
Minutes - Call 20/11/2023

Attendees:

- Alex Donkers (Eindhoven University of Technology, NL)
- Daniel Hernández (University of Stuttgart, Stuttgart, Germany)
- Eduardo Costa (University of the West of England, Bristol, UK)
- Jan-Iwo Jäkel (RWTH Aachen University,, GER
- Stefan Spreng (Züblin, Stuttgart, GER)
- Eva Heinlein (RWTH Aachen)
- Mathias Bonduel (Neanex)
- Aaron Wagner
- Katja Breitenfelder
- Christopher Krueger
- Georgios Triantafyllidis
- Dimitris Mavrokapnidis
- Philipp Hagedorn
- Theo Dunas
- Tim Rossi
- Zhancheng Qiang
- Xi Peng
- Jakob Martin (iabi e.V.)
- Klaus Linhard (iabi e.V.)
- Marin Ljuban
- Rui de Klerk
- Sebastian Seiss (Bauhaus-University Weimar, Germany)
- Diellza Elshani (University of Stuttgart, Germany)

Presentation slides

- slides: https://github.com/w3c-lbd-cg/lbd/blob/gh-pages/presentations/20231120_Presentation_DiellzaElshani.pdf
- minutes
- demo material

Date and time

- 20/11/2023, Monday, 14:00-15:30@UTC/ 16:00-17:30@CET/ 07:00-08:30@PDT

Moderators

1. Alex Donkers

Agenda

1. Introduction of new members
2. Diellza Elshani & Aaron Wagner: 'bhOWL: Federated Data Schemas and Semantic Web Technologies via Design Tools with a Focus on Grasshopper 3D'
3. Discussion

4. Further topics

Minutes

1. **Introduction of new members**
    - Eduardo Costa: university of West of England (Bristol), research on digital fabrication, applying ontologies in the process
    - Stefan Spreng: colleague of Diellza, works in construction company Züblin and presently at University of Stuttgart
    - Daniel Hernandez: post-doc at University of Stuttgart, working with Diellza, department of IA. Applying ontologies and Linked Data
    - Christopher Krueger: joining as product manager of Bluebeam from Texas
2. **Diellza Elshani & Aaron Wagner: 'bhOWL: Federated Data Schemas and Semantic Web Technologies via Design Tools with a Focus on Grasshopper 3D'**
    - context
        - cluster of excellence IntCDC > umbrella for different projects
        - co-design: design, analysis, manufacturing and construction from different disciplines
            - facts
            - rules
            - constraints
            - project specific data
        - project: RP20-1 Knowledge representation for multi-disciplinary co-design
            - data interoperability
            - knowledge representation
            - BIM
        - project context
            - which tools represent what and how => how exchanged
            - represent: building data can be tool dependent <> IFC <> building ontologies
            - exchange of datasets: IFC data (importers/exporters) <> plugins (Speckle, BHoM)
    - co-design interoperability paradigms
        - three existing methods
            - IFC data schema
            - Speckle: focus on connecting software (data) but less focus on representing
            - BHoM > compatibility with parametric design
        - distributed <> centralized
            - decentralized: data exchanges between different tools, no shared data schema nor database > current way of working in the AEC, challenges with versions
                - inefficient for coordination and clash detection
            - centralized: tools which are compatible to a shared data schema and database model
                - data loss between tools and shared data schema
                - e.g. IFC
            - federated: combination of both decentralized and centralized
                - shared data schema is less big compared to centralized approach
                - shared ontology is central
                - shared database model, e.g. RDF

        - every tool has its own graph => connect
        - BHoM already useable with existing design software (adapters), large library
            - separates functions from objects => rules and facts
        - similar approach with IFC <> plugins: designers don't need to learn new approaches nor tools + easier to extend
        - why BHoM instead of Speckle?
            - federated data approach
            - 1200 existing objects instead of starting from scratch
    - BHoM: initiative by Buro Happold => move together to a Linked Data ontology
        - graphs as database model > bidirectional exchange to design tools
            - rules and constraints are reusable
        - reuse classes across projects
        - BHoM ecosystem > many applications supported
        - components
            - the object model (oM) > classes
            - adapters
            - the engine
            - combined: toolkit
        - example oM "Wall"
            - C# framework: implementing composition over inheritance (interfaces)
            - C# graph: start and end point of two distinct objects can be linked
    - Conversion pattern for BHoM to RDF representation
        - mapped BHoM components to SWT counterparts in OWL/RDFS
        - data level
        - connecter to GraphDB to store RDF
    - application example: roof construction
        - architectural design
        - rules from manufacturer and graph
        - linking graphs
        - data validation with SHACL
        - rules/inferencing > find edge cassettes (only rough estimation)
        - querying weight using SPARQL based on individual measurements
    - application of the toolkit at Digital Futures Workshop (2022)
        - many inputs > improvements
        - videos and lectures are on vimeo and Youtube
    - application at CODEC hackathon
        - energy analysis
        - compare design options
    - live demo by Aaron Wagner
        - Grasshopper and Rhino CAD environment
            - Rhino geometry to graph
                - TTL adaptor
                - push component
                - result: individuals + classes
        - when applying the BHoM content as input > longer TTL
        - create custom BHoM objects
        - export to TTL <> GraphDB adaptor (API interactions)
        - not only objects but also interfaces stored in the graph
        - bidirectional exchange: also from GraphDB to Rhino (update geometry)

3. Discussion

- [Rui de Klerk] possible to do SPARQL queries from Grasshopper?
    - currently we query from GraphDB directly. Could make a component. Not sure if there's additional benefits (GraphDB has built-in query visualization)
    - [Rui] more flexibility? Prepare endpoints for specific use cases at GraphDB?
- [Tim Rossi] GitHub repository? Access to demo files presented?
    - entire project is on GitHub (BHoM RDF prototypes)
    - meeting is recorded > files of the demo can be shared
- [Rui de Klerk] size of the graph database (meshes)? How to handle that?
    - option 1: full serialization of geometry
    - option 2: no serialization of geometry in graph (only string) > base64
        - BHoM has it's own geometry format which mappable to the geometry formats of adaptors
    - currently no intermediate solution
- [Mathias Bonduel] how do you keep track of data coming from different sources (applications, users)? Which data is leading?
    - architect starts designing > named graph
    - structural eng > other named graph, but connected content to arch graph
    - options
        - detect problem > does not know who should solve
        - arch is leading > can change/limit struct design
        - struct is leading > can change/limit arch design
    - [Alex] GIT approach with project manager approving applied in the past > hard to put all requirements in graph
- [Tim] current challenges?
    - OWL vocabulary
    - geometry challenge (discussed above)
    - connecting elements together which are actually the same
    - delete actions > delete object => delete properties as well (cascading deletes)
- [Rui] next steps in project? How does your work relate to previous LBD work?
    - (above)
    - map to IFC and LBD work
- [Alex Donkers] interaction between deleting in GraphDB and deleting in Grasshopper: also possible to load/create content from the RDF?
    - [Aaron] many possibilities
    - [Alex] alignment between BHoM/BHOWL and BOT. Everyone can create their own ontologies/extensions. Link to bSDD content?
        - planning to publish BHoM on bSDD
        - classification using other systems
        - approach in BHoM: define classes per institute and publish in GitHub
- [Alex] how much tied to Grasshopper?
    - BHoM can be used through existing UI: Excel, Revit (Dynamo) and Rhino (Grasshopper)
        - needed for interaction with GraphDB
    - alternative: C# programming to BHoM API > needed for other integrations (e.g. structural engineering in Robot)
- [Rui] API for Python using BHoM?
    - Python toolkit exists for BHoM

4. Further topics

-

Next Call

- 18/12/2023, Monday, 14:00-15:30@UTC/ 16:00-17:30@CET/ 07:00-08:30@PDT

  Agenda: Philipp Hagedorn, Ruhr University Bochum

  We are interested in getting suggestions from the community about potential agenda items and **Elevator Pitches** for the following calls. Please send your suggestions to the chairs or to internal-lbd@w3.org, whether you have a short presentation to bootstrap the discussion, and an approximate duration you think the discussion will last.

Previous minutes

https://github.com/w3c-lbd-cg/lbd/tree/gh-pages/minutes