SupplyChain2Map

Software Requirement Specification

Mirco Sprung, Omid Najaei Nezhad, Mohammad Ali Ghasemi, Sina Mahmoodi

External Interface Requirements

- Hardware Interface Requirements
  - JS library needs browser => Device supporting display screen, can run browser
  - E.g. PC and handheld devices
  - Requires functioning network card connected to the internet
- Communication Interface Requirements
  - Two protocols between client and server, HTTP and SPARQL
  - SPARQL itself uses HTTP, HTTP relies on TCP/IP
  - Transferred data either in SPARQL query format, or JSON documents

External Interface Requirements *(cont.)*

- Software Interface Requirements
  - Browser API (or in our case Babel), e.g. user screen size, LocalStorage for cache
  - Leaflet API for visualization
  - SPARQL as communication protocol and query format
- Ontology
  - SCOR provides framework for talking about supply chains
  - SCORVoc materializes SCOR in concept hierarchies using RDF

Functional Requirements

Requirement ID	Requirement Statement	Must/Want	Comments
FR001	The website shall have a world map.	Must	
FR002	The map shall be interactable.	Must	Providing general map interaction like zooming or dragging.
FR003	SC data shall be visualized on the map.	Must	
FR004	The nodes on the map shall be selectable.	Must	Selection in order to get additional information about the node.
FR005	The website shall provide analysis functionality over the SC data.	Must	Given already existing algorithms and metrics.

Design Constraints

- Memory usage
  - JS library, runs in browser, doesn’t require a lot of memory
- Disk usage
  - The same reasons limit the disk capacity available for us by the library
  - Must load lazily, and cache most used data

Software System Attributes

- Modularity and maintainability
  - Final result is a library which could potentially be used in different projects
  - If open-source, community can work on it
  - If not, it needs to be maintained by the teams working on the aforementioned projects
- Security
  - Data might be on servers owned by the enterprises
  - Must use a secure authentication method, to prevent leaking enterprise data
- Interoperability
  - Enterprises might use different methods for keeping or exposing data
  - By incorporating different methods (RDF File, SPARQL endpoints, ...) should be as interoperable as possible
- Usability

Questions?