

35

Examples of Community platforms

- Task centered
- User centered
- Topic centered

F. Mäkelä - Virtual Communities

36

Task centric communities Collaborative ontology development

- Ontology (Gruber93) : formal specification of terms and relations in a domain
- Useful for organizing and categorizing data (used on Internet : Web of Data micro-format)
- Ontology development steps
 - Specify the domain
 - Initiate a list of terms of the domain, classes (hierarchy), relations between classes
 - Search for existing ontologies
 - Iterate on the list of terms till group members are satisfied
 - Populate the ontology (= create a knowledge base)

F. Mäkelä - Virtual Communities

37

Collaborative ontology development

- Why collaborative?
 - Ontologies represent a consensus on a domain
 - Different expertises: various points of views, domain experts, end-users, knowledge engineer
- List of requirements for collaborative ontology development
 - Proposed by Todorache 2008
 - Extended by Al Qawameh 2018
 - This list is useful to compare different methods/tools

F. Mäkelä - Virtual Communities

38

Collaborative ontology development Requirements 1/3

- R1 **Discussion**: sharing ideas, comments, queries
- R2 Support of various level of expressiveness: from high-level description to very **detailed** formalization of the domain
- R3 **User management and Provenance information**: users should be identified, actions should be traced
- R4 **Scalability, reliability, robustness**: ability of the tool to support big ontologies: and to offer trusted stable environment

F. Mäkelä - Virtual Communities

39

Collaborative ontology development Requirements 2/3

- R5 **Access control**: Read/write access on parts of the ontology granted to users
- R6 **Workflow support**: Definition and execution of ordering rules for tasks (example: edition step → validation step)
- R7 **Access type**: synchronous or asynchronous

F. Mäkelä - Virtual Communities

40

Collaborative ontology development Requirements 2/3

- R8 **User friendly**: easy to use for all kinds of users (not familiar with such tools)
- R9 **Tool's setup, open source code**: easiness of installing the tool; possibility of control the tool's content, and to extend the functionalities
- R10 **Ontology validation**: checking functions such as syntax and semantic validation
- R11 **Supported format**: existing and produced ontology may be in different formats: owl, rdf-turtle, rdf-N3, ...

F. Mäkelä - Virtual Communities

41

Collaborative ontology development Tool: Protégé

- From Stanford University <http://protege.stanford.edu/>
- Most famous tool for ontology edition
- Free, open source (Java), organized as a framework with plugins
- Features: graphic user interface, consistency validation, new information inferencing
- WebProtégé for online tool
 - Limited functions: multi-users, comments/discussion attached to objects <http://webprotege.stanford.edu>

P. Masek - Virtual Communities

42

Collaborative ontology development Tool: VoCol

- From University of Bonn and Fraunhofer IAS
- Features (integration of several components):
 - Versioning control system, based on github for lacking of evolutions**
 - Text editor with syntax validation (turtle) (easy to use for domain experts)
 - Checking of inconsistency/constraints
 - Editor free (theoretically) (format translation for a unique serialization service)
 - Documentation generation (HTML), visualization
 - Query service (through a SPARQL endpoint)
- Collaborative features fully rely on Github
- Difficult to setup; No freedom in the generated output; Bugs or missing functions (it is not a product!)

P. Masek - Virtual Communities

43

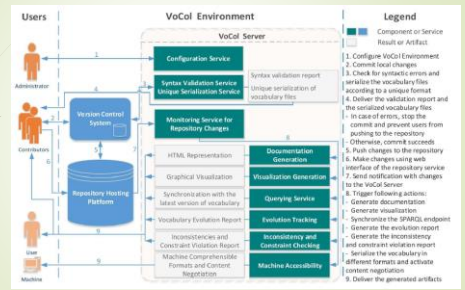
Principles of Github

- Dedicated to controlling versions of projects
- User identify into repositories (repo)
- Repository: contains all files related to a project
- Main concepts in the main page
 - Code:** structure & contents of the repo
 - Issues:** discussion topics, problems related to the repo
 - Pull Requests:** list of changes for review / acceptance
 - Commits:** all saved and documented changes to the repo
 - Fork:** personal copy of the repo → **Pull request:** transfer the changes to initial repo
 - Pull** to create local (offline) copy, **Push** to send changes to the forked repo

P. Masek - Virtual Communities

44

Vocal Architecture and workflow



45

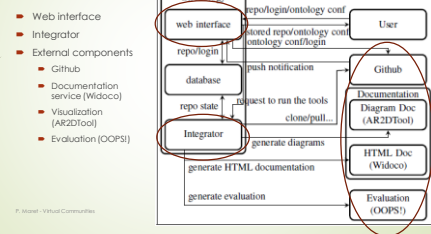
Collaborative ontology development Tool: Ontology

- From University Polytechnic of Madrid (Alobaid 2015)
- Based on Github (similarly to Vocol)
- External tools: Documentation, Visualization, Evaluation
- Role of Ontology (see architecture)
 - User's access to Github repositories: Setup of the ontologies (tools activated)
 - Launch tools: Production of diagrams, HTML documentation, evaluation report
 - Communicate with Github: submit documents (pull request) to be reviewed/merged by the maintainer; Add issues for the pitfalls generated by the evaluation tool

P. Masek - Virtual Communities

46

Ontology architecture



P. Masek - Virtual Communities

47

Evaluation of Task centric communities/ Collaborative ontology development

In-use experiment and comparison

- Difficulties: availabilities of experts, subjectivity of modeling, importance of the user interface

Compare tools based on a list of requirements

- Izquierdo 2016 : Analyse distributed ontology engineering tools through agile approach (continuous development, quick responses to change, collaborative)
- See next slide

P. Nouri - Virtual Communities

48

Comparison table of collaborative ontology development tools (from Alqawasmeh)

	Category 1					Category 2			Category 3		Comments
	R1	R2	R5	R10	R11	R3	R6	R7	R8	R9	
Domingue [10]	✓	✓	✓	NI	NI	✓	NI	both	NI	✗	-
Arpaz et al. [11]	✓	✓	✓	NI	NI	✓	✓	NI	NI	✗	Arch. for ontology dev.
Kosaki et al. [12]	✗	✓	✓	NI	NI	✓	NI	NI	NI	✗	-
Sure et al. [13]	✓	NI	NI	✓	✓	✓	✓	NI	NI	NI	-
Hayes et al. [14]	✗	✓	✓	NI	NI	NI	NI	NI	NI	✗	-
Konnala et al. [16]	NI	✓	✓	✓	✓	✓	✓	NI	NI	NI	Frame work
Amr et al. [17]	✓	✓	✓	✓	✓	✓	✓	NI	NI	NI	-
De Leenheer and Doherty [18]	✓	✓	✓	✗	NI	✓	✓	NI	NI	NI	-
Noy et al. [19]	✗	✓	✗	✓	✓	✓	✓	Synce	✓	✓	-
Tudorache et al. [21]	✓	✓	✗	✓	✓	✓	✓	Synce	✓	✓	-
Glukini et al. [22]	NI	✓	✓	NI	NI	✓	NI	NI	NI	NI	-
Tawfik et al. [23]	✓	NI	✓	NI	NI	✓	NI	Asynce	NI	NI	-
Alshaid et al. [24]	✗	✓	✓	✓	✗	✓	✗	NI	✓	✓	based on Github
Hallaj et al. [25]	✗	✓	✓	✓	✗	✓	✗	Asynce	✓	✓	based on Github

R1: Discussions, R2: Various levels of expressiveness, R3: Management and provenance of information
 R5: Access control, R6: Workflow support, R7: Access type, R8: User Friendly
 R9: Easy to Install/Configure- open source, R10: Ontology validation, R11: Supported ontology types
 Y: YES, N: No, NI: No Information

Vocal
Ontology