Ontopolis.net: Social-Semantic Web Application for Participative e-Democracy

Václav Belák

Department of Information and Knowledge Engineering
University of Economics, Prague

Seminar of **K**nowledge **E**ngineering **G**roup
December 10, 2009

Outline

1 Introduction

2 Ontopolis project achievements
- Ontology
- Design and implementation
- Similarity algorithms

3 Summary

Motivation

Tož demokracii bychom už měli, teď ještě nějaké ty demokraty. — T.G.Masaryk

- Present democracies are based on competition of political parties
- The competition is biased by
  - Personal relationships between parties' secretaries and elite politicians
  - High costs of entry, Advertisement
- It resembles oligopolistic competition
- Main objective of the Ontopolis.net is to allow people to collaboratively create solutions of political issues and self-organize around these solutions
- The system is based on knowledge technologies to achieve this goal

Motivation

Tož demokracii bychom už měli, teď ještě nějaké ty demokraty. — T.G.Masaryk

- Present democracies are based on competition of political parties
- The competition is biased by
  - Personal relationships between parties' secretaries and elite politicians
  - High costs of entry, Advertisement
- It resembles oligopolistic competition
- Main objective of the Ontopolis.net is to allow people to collaboratively create solutions of political issues and self-organize around these solutions
- The system is based on knowledge technologies to achieve this goal

Motivation

Tož demokracii bychom už měli, teď ještě nějaké ty demokraty. — T.G.Masaryk

- Present democracies are based on competition of political parties
- The competition is biased by
  - Personal relationships between parties' secretaries and elite politicians
  - High costs of entry, Advertisement
- It resembles oligopolistic competition
- Main objective of the Ontopolis.net is to allow people to collaboratively create solutions of political issues and self-organize around these solutions
- The system is based on knowledge technologies to achieve this goal

Why is Ontopolis.net different?

- There are several similar applications currently:
  - Localocracy.org, E-democracy.org, Whitehouse2.org, Openpolitics.ca, Zmenpolitiku.cz, Smartocracy.net, Facebook.com, . . .
- They are not designed for direct political action
  - They are *reactive*
- They do not use shared formal ontology
  - They are mutually incompatible
  - *Users are locked-up* in the system

Why is Ontopolis.net different?

- There are several similar applications currently:
  - Localocracy.org, E-democracy.org, Whitehouse2.org, Openpolitics.ca, Zmenpolitiku.cz, Smartocracy.net, Facebook.com, . . .
- They are not designed for direct political action
  - They are *reactive*
- They do not use shared formal ontology
  - They are mutually incompatible
  - *Users are locked-up* in the system

Why is Ontopolis.net different?

- There are several similar applications currently:
  - Localocracy.org, E-democracy.org, Whitehouse2.org, Openpolitics.ca, Zmenpolitiku.cz, Smartocracy.net, Facebook.com, . . .
- They are not designed for direct political action
  - They are *reactive*
- They do not use shared formal ontology
  - They are mutually incompatible
  - *Users are locked-up* in the system

Creating an artificial self-organizing system

- SO is a *spontaneous emergence of global structure out of local interactions[2]*
- *A key characteristic of an artificial self-organizing system is that structure and function of the system 'emerge' from interactions between the elements.* ***The purpose should not be explicitly designed, programmed, or controlled.[1]***
- The system consists of its:
  - users
  - data
  - software
  - ontology
- Each of this part has to be free (as in freedom).

Implications on the system design

- Ontology must not *a priori* conceive any particular political issue nor organization
  - It has to be as flexible as possible
- Free interactions namely between:
  - users with one another
  - users with the system
  - users with the ontology
- The system is free/libre open-source software
- Ontology is based on existing ones
- *All* data are represented in RDF.

Implications on the system design

- Ontology must not *a priori* conceive any particular political issue nor organization
  - It has to be as flexible as possible
- Free interactions namely between:
  - users with one another
  - users with the system
  - users with the ontology
- The system is free/libre open-source software
- Ontology is based on existing ones
- *All* data are represented in RDF.

Outline

1 Introduction

2 Ontopolis project achievements

- Ontology
- Design and implementation
- Similarity algorithms

3 Summary

Ontopolis' Ontology (OPOL)

- It plays a crucial role in the system:
  - all *data are represented* using OPOL
  - *data are* also *validated* using OPOL
  - it enable to *guarantee the freedom of data*
- OPOL re-uses several existing ontologies:
  - **FOAF** for representation of persons and their relationships
  - **SIOC** for representation of content and its relationships to users
  - **DOLCE** for representation of political plans and goals
  - **WordNet Basic** for disambiguation of descriptions of content
  - **DCTerms**, **Konfidi**

Ontopolis' Ontology (OPOL)

- It plays a crucial role in the system:
  - all *data are represented* using OPOL
  - *data are* also *validated* using OPOL
  - it enable to *guarantee the freedom of data*
- OPOL re-uses several existing ontologies:
  - **FOAF** for representation of persons and their relationships
  - **SIOC** for representation of content and its relationships to users
  - **DOLCE** for representation of political plans and goals
  - **WordNet Basic** for disambiguation of descriptions of content
  - **DCTerms**, **Konfidi**

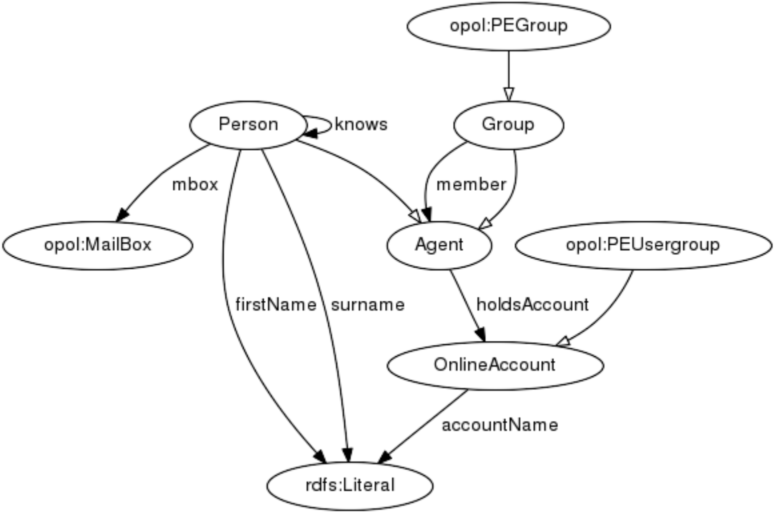
Ontopolis' Ontology (OPOL)

- It plays a crucial role in the system:
  - all *data are represented* using OPOL
  - *data are* also *validated* using OPOL
  - it enable to *guarantee the freedom of data*
- OPOL re-uses several existing ontologies:
  - **FOAF** for representation of persons and their relationships
  - **SIOC** for representation of content and its relationships to users
  - **DOLCE** for representation of political plans and goals
  - **WordNet Basic** for disambiguation of descriptions of content
  - **DCTerms**, **Konfidi**

Ontopolis' Ontology (OPOL)

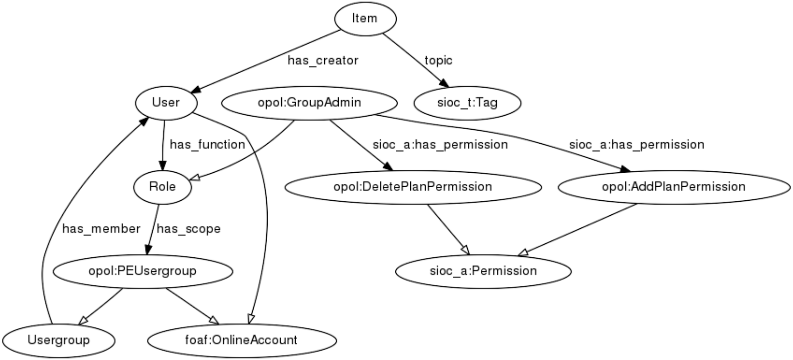
- It plays a crucial role in the system:
  - all *data are represented* using OPOL
  - *data are* also *validated* using OPOL
  - it enable to *guarantee the freedom of data*
- OPOL re-uses several existing ontologies:
  - **FOAF** for representation of persons and their relationships
  - **SIOC** for representation of content and its relationships to users
  - **DOLCE** for representation of political plans and goals
  - **WordNet Basic** for disambiguation of descriptions of content
  - **DCTerms**, **Konfidi**

Friend of a Friend

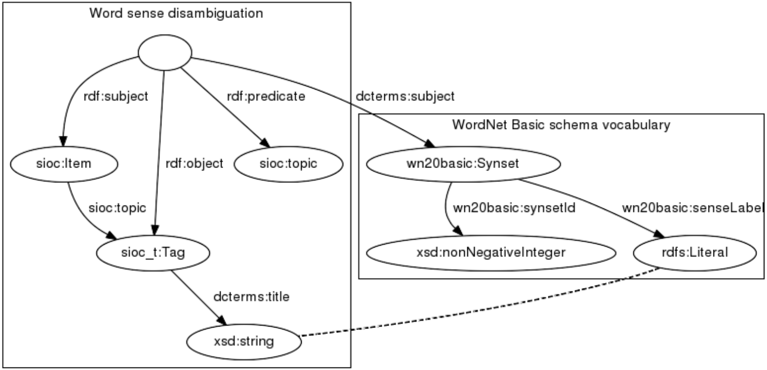


SIOC

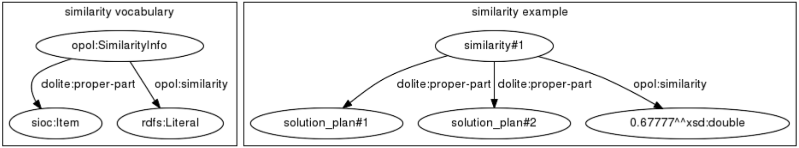
- **S**emantically-**I**nterlinked **O**nline **C**ommunities



Representing disambiguation



Representing similarity



Outline

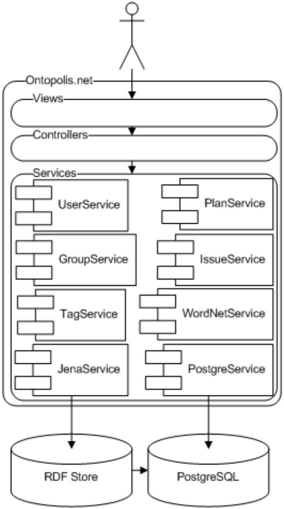
1 Introduction

2 Ontopolis project achievements

- Ontology
- Design and implementation
- Similarity algorithms

3 Summary

Global architecture



Key technologies

- Ontopolis.net is written using *Grails framework*
  - agile development
  - Groovy programming language
  - integrating Spring, Sitemesh, JUnit, ...
- Data are stored in *Jena*'s *SDB* RDF store with *PostgreSQL* at the backend
  - Jena's generic rules engine is used for real-time reasoning
- Data are validated using *Pellet* reasoner and its IC plug-in[6]

Key technologies

- Ontopolis.net is written using *Grails framework*
  - agile development
  - Groovy programming language
  - integrating Spring, Sitemesh, JUnit, ...
- Data are stored in *Jena*'s *SDB* RDF store with *PostgreSQL* at the backend
  - Jena's generic rules engine is used for real-time reasoning
- Data are validated using *Pellet* reasoner and its IC plug-in[6]

Key technologies

- Ontopolis.net is written using *Grails framework*
  - agile development
  - Groovy programming language
  - integrating Spring, Sitemesh, JUnit, . . .
- Data are stored in *Jena*'s *SDB* RDF store with *PostgreSQL* at the backend
  - Jena's generic rules engine is used for real-time reasoning
- Data are validated using *Pellet* reasoner and its IC plug-in[6]

Use-cases

- Creating an issue
- Creating a solution of an issue – political candidate role
- Declaring support to a candidate – political supporter role (follower)
- Creating group
  - group administrator
  - adding goals of the group
  - three roles in the group:
    - plain member
    - follower
    - candidate
  - the only way to share a plan

Use-cases

- Creating an issue
- Creating a solution of an issue – political candidate role
- Declaring support to a candidate – political supporter role (follower)
- Creating group
  - group administrator
  - adding goals of the group
  - three roles in the group:
    - plain member
    - follower
    - candidate
  - the only way to share a plan

Use-cases

- Creating an issue
- Creating a solution of an issue – political candidate role
- Declaring support to a candidate – political supporter role (follower)
- Creating group
  - group administrator
  - adding goals of the group
  - three roles in the group:
    - plain member
    - follower
    - candidate
  - the only way to share a plan

Use-cases

- Creating an issue
- Creating a solution of an issue – political candidate role
- Declaring support to a candidate – political supporter role (follower)
- Creating group
  - group administrator
  - adding goals of the group
  - three roles in the group:
    - plain member
    - follower
    - candidate
  - the only way to share a plan

Outline

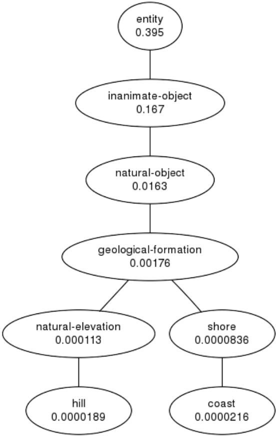
1 Introduction

2 Ontopolis project achievements

- Ontology
- Design and implementation
- Similarity algorithms

3 Summary

Fragment of WordNet



Lin similarity

Theorem

$$sim_{lin}(A,B) = \frac{\log P(common(A,B))}{\log P(description(A,B))}$$

For WordNet:

$$sim_{lin} = \frac{2 \times \log P(C_O)}{\log P(C_1) + \log P(C_2)}$$

Example

$$sim_{lin}(hill, coast) = \frac{2 \times \log(0.00176)}{\log(0.0000189) + \log(0.0000216)} \doteq 0.59.$$

Lin similarity

Theorem

$$sim_{lin}(A,B) = \frac{\log P(common(A,B))}{\log P(description(A,B))}$$

For WordNet:

$$sim_{lin} = \frac{2 \times \log P(C_O)}{\log P(C_1) + \log P(C_2)}$$

Example

$$sim_{lin}(hill, coast) = \frac{2 \times \log(0.00176)}{\log(0.0000189) + \log(0.0000216)} \doteq 0.59.$$

Disambiguating descriptions

- Issues and Plans can be tagged
- Tags are disambiguated before saving using WordNet
- We use similarity measure published by Lin[4]
- Pair-wise combinations of tags are considered and the highest similarity determines the choice of synsets

- Disambiguated tags are then used to compute similarity between two tagged items
- When a tagged item is about to be saved, possible similar items are determined and their mutual similarities are computed
  - possible similar plans are those plans, that have the same synset, tag or issue
- Only similarities above certain threshold are stored (currently 0.3)

Disambiguating descriptions

- Issues and Plans can be tagged
- Tags are disambiguated before saving using WordNet
- We use similarity measure published by Lin[4]
- Pair-wise combinations of tags are considered and the highest similarity determines the choice of synsets
- Disambiguated tags are then used to compute similarity between two tagged items
- When a tagged item is about to be saved, possible similar items are determined and their mutual similarities are computed
  - possible similar plans are those plans, that have the same synset, tag or issue
- Only similarities above certain threshold are stored (currently 0.3)

Item similarity algorithm

Example

Similarity between $tags_1 = \{$”*pickpocket*”,“*Praha*”$\}$ and $tags_2 = \{$”*criminality*”,“*Prague*”$\}$

1. Determine similarities between *picpocket* and $tags_2$:
   1. The similarity between *pickpocket* and *criminality* is 0
   2. The similarity between *pickpocket* and *Prague* is 0.059
   3. Maximal similarity for *pickpocket* is 0.059
   4. $os = 0 + 0.059$
2. Determine similarities between *Praha* and $tags_2$:
   1. The similarity between *Praha* and *criminality* is 0
   2. The similarity between *Praha* and *Prague* is 1
   3. Maximal similarity for *Praha* is 1
   4. $os = 0.059 + 1$
3. The similarity is $\frac{1.059*2}{4} \doteq 0.53$

Support of Self-Organization

- Intelligent recommendation based on similarity
  - Similar issues
  - Similar plans
  - Similar users
    - Users are not tagged, so users similar to a given users are those one, who share an issue, a group and/or a plan with the user
- Support is an implicit trust relationship
- Ordering plans and users by their count of support enables the emergence of authorities

Support of Self-Organization

- Intelligent recommendation based on similarity
  - Similar issues
  - Similar plans
  - Similar users
    - Users are not tagged, so users similar to a given users are those one, who share an issue, a group and/or a plan with the user
- Support is an implicit trust relationship
- Ordering plans and users by their count of support enables the emergence of authorities

Conclusion

- The competition of political parties is biased
- Internet enabled the opportunity to self-organize and collaborate, but we do not use its full potential in politics
- Present e-democracy applications are *reactive* and *mutually incompatible*
- Our ontology provides a way how to get *integrated overview* of citizens' opinions
- The ontology also helps to guarantee the *freedom of data*

Conclusion

- The competition of political parties is biased
- Internet enabled the opportunity to self-organize and collaborate, but we do not use its full potential in politics
- Present e-democracy applications are *reactive* and *mutually incompatible*
- Our ontology provides a way how to get *integrated overview* of citizens' opinions
- The ontology also helps to guarantee the *freedom of data*

Conclusion

- The competition of political parties is biased
- Internet enabled the opportunity to self-organize and collaborate, but we do not use its full potential in politics
- Present e-democracy applications are *reactive* and *mutually incompatible*
- Our ontology provides a way how to get *integrated overview* of citizens' opinions
- The ontology also helps to guarantee the *freedom of data*

Conclusion

- The competition of political parties is biased
- Internet enabled the opportunity to self-organize and collaborate, but we do not use its full potential in politics
- Present e-democracy applications are *reactive* and *mutually incompatible*
- Our ontology provides a way how to get *integrated overview* of citizens' opinions
- The ontology also helps to guarantee the *freedom of data*

What has been achieved

- Ontopolis.net has been built as a proof-of-concept of my research
- It is designed with specifics of an artificial self-organizing system in mind
  - free software
  - free data
  - free ontology
- Ontopolis.net represents an *active* approach to e-democracy

What has been achieved

- Ontopolis.net has been built as a proof-of-concept of my research
- It is designed with specifics of an artificial self-organizing system in mind
  - free software
  - free data
  - free ontology
- Ontopolis.net represents an *active* approach to e-democracy

What has been achieved

- Ontopolis.net has been built as a proof-of-concept of my research
- It is designed with specifics of an artificial self-organizing system in mind
  - free software
  - free data
  - free ontology
- Ontopolis.net represents an *active* approach to e-democracy

Future work

- Most important is a real-world use-case, but several implementation details has to be done:
  - pagination
  - updating of content
  - issue hierarchies
  - decentralization of group administration
  - argumentation support (incorporating argumentation ontology)[3]
- Other features are deserved/planned:
  - OpenID
  - OpenSocial API, maybe Facebook connect
  - Aligning the whole OPOL with DOLCE
  - Explicit trust and support for dynamically distributed democracy[5]
  - REST

Bibliography I

Carlos Gershenson and Francis Heylighen.
When can we call a system self-organizing?
In *ECAL*, volume 2801, pages 606–614, 2003.

Francis Heylighen.
Encyclopedia of Library and Information Sciences, chapter Complexity and Self-organization.
Taylor & Francis, 2008.

Mark Klein.
Achieving collective intelligence via large-scale on-line argumentation.
Internet and Web Applications and Services, International Conference on, 0:58, 2007.

Bibliography II

Dekang Lin.
An information-theoretic definition of similarity.
In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

Marko A. Rodriguez, Daniel J. Steinbock, Jennifer H. Watkins, Carlos Gershenson, Johan Bollen, Victor Grey, and Brad deGraf.
Smartocracy: Social networks for collective decision making.
Hawaii International Conference on System Sciences, 0:90b, 2007.

Bibliography III

Evren Sirin and Jiao Tao.
Towards integrity constraints in owl.
In Rinke Hoekstra and Peter F. Patel-Schneider, editors, *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009)*, 10 2009.