Web ontologies make up the Semantic Web, which allows data to be shared and reused across application, enterprise, and community boundaries. Ontologies have a wide range of applications, including such areas as biology, medicine, genetics, oil production, and information technologies.

This project aims at creating a system to allow performing version control and collaborative distributed development of web ontologies by integrating custom ontology diff and merge tools with existing distributed version control systems and issue trackers.

Advantages of the approach

- Is based on mature systems;
- Does not require persistent network connection;
- Allows developers to use ontology editing software they are accustomed to;
- Does not restrict web ontology language expressiveness.

Components of the System
Algorithms

Ontology = (E, A, N, I, P, d, f) where
- E ⊂ E' - entities;
- A ⊂ A' - axioms;
- N ⊂ N' - ontology annotations (n_p, n_v), where n_p - annot. property, n_v - annot. value;
- I ⊂ I' - imports (links to other ontologies);
- P ⊂ P' = S × S - prefixes (p_n, p_i), where p_n - prefix name, p_i - prefix value;
- d ∈ D - ontology identifier (r_n, r_i), where r_n - Ontology IRI, r_i - Version IRI;

Ontology Diff

\[ v_1 = (E_1, A_1, N_1, I_1, P_1, d_1, f_1) \]
\[ v_2 = (E_2, A_2, N_2, I_2, P_2, d_2, f_2) \]
C_A = δ_A(A_1, A_2) = \{ AddAxiom(a) | a ∈ A_2 \setminus A_1 \} \cup \{ RemoveAxiom(a) | a ∈ A_1 \setminus A_2 \}
C_N = δ_N(N_1, N_2) = \{ AddOntologyAnnotation(n) | n ∈ N_2 \setminus N_1 \}
∪ \{ RemoveOntologyAnnotation(n) | n ∈ N_1 \setminus N_2 \}
C_I = δ_I(I_1, I_2) = \{ AddImport(i) | i ∈ I_1 \setminus I_2 \} \cup \{ RemoveImport(i) | i ∈ I_2 \setminus I_1 \}
C_P = δ_P(P_1, P_2) = \{ AddPrefix(p_n, p_i) \mid (p_n, p_i) ∈ P_2 \setminus P_1 \}
∪ \{ RemovePrefix(p_n, p_i) \mid (p_n, p_i) ∈ P_1 \setminus P_2 \}
C_d = δ_d(d_1, d_2) = \{ SetOntologyIdentifier(d_1, d_2) \mid d_1 ≠ d_2 \}
∪ \{ SetOntologyIdentifier(d_1, d_2) \mid d_1 = d_2 \}
C_f = δ_f(f_1, f_2) = \{ SetOntologyIRI(f_1, f_2) \mid f_1 ≠ f_2 \}
∪ \{ SetOntologyIRI(f_1, f_2) \mid f_1 = f_2 \}

Changes C_A \cup C_P can be divided into:
- Matching changes C_m = C_A∩C_P;
- Conflicting changes of user 1: C_m1;
- Conflicting changes of user 2: C_m2;
- Other changes of user 1: C'_m1 = (C_m1 \setminus C_m) \cup C_m1;
- Other changes of user 2: C'_m2 = (C_m2 \setminus C_m) \cup C_m2;

Conflict Detection

Axios: C_A1 = \{ c ∈ C_A \mid \sigma(c) \cap \sigma(C_A \setminus C_m) ≠ ∅ \};
C_A2 = \{ c ∈ C_A \mid \sigma(c) \cap \sigma(C_A \setminus C_m) = ∅ \};
Prefixes: C_P1 = \{ c ∈ C_P \mid \sigma(c) \cap \sigma(C_P \setminus C_m) ≠ ∅ \};
C_P2 = \{ c ∈ C_P \mid \sigma(c) \cap \sigma(C_P \setminus C_m) = ∅ \};
Identifier: C_I1 = \{ c ∈ C_I \mid \sigma(c) \cap \sigma(C_I \setminus C_m) ≠ ∅ \};
C_I2 = \{ c ∈ C_I \mid \sigma(c) \cap \sigma(C_I \setminus C_m) = ∅ \};
Format: C_F1 = \{ c ∈ C_F \mid \sigma(c) \cap \sigma(C_F \setminus C_m) ≠ ∅ \};
C_F2 = \{ c ∈ C_F \mid \sigma(c) \cap \sigma(C_F \setminus C_m) = ∅ \};

Project Proposals

- Collaborative ontology development tools
- Security of ontological knowledge bases
- Information systems based on Semantic Web technologies
  - Decision support system
  - E-library system

Publications
