Bridging heterogeneous, autonomous, dynamic knowledge at runtime

M-ZONES

Declan O'Sullivan
Ruaidhri Power

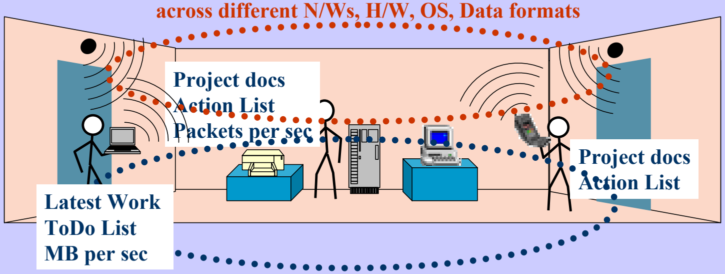
Knowledge & Data Engineering Group, Trinity College Dublin
http://kdeg.cs.tcd.ie

Outline

- The Problem
- Characteristics of Environment and Solution
- Context Information Service
- Semantic Matching Utility
- Conclusions

Example – Project Meeting

Syntactic Interoperability:
Create communication between applications
across different N/Ws, H/W, OS, Data formats



Semantic Interoperability:
Create understanding across different applications
across different conceptual models, reference systems,
and knowledge structures

Characteristics of Environment

- In traditional environments, integration involves
  - Dealing with KS model heterogeneity at design time
  - "Integration aware" autonomy of KS
  - Static deployment of knowledge sources
- Our target environment requires
  - Runtime bridging of KS model heterogeneity
  - "Integration unaware" autonomy of KS
  - Dynamic deployment of knowledge sources

Use Cases

- Two major application interoperability use cases require bridging heterogeneous, autonomous, dynamic knowledge at runtime

1. Sharing of Context
   - To support ad hoc and mobility
     - E.g. “Lets collaborate”
   - To support service adaptivity
     - E.g. meeting room being shared?, meeting to be minuted?
2. Creating understanding based on matching of semantics
   - To support Task Decomposition
     - User concepts versus system understood concepts
     - E.g. “Display the minutes of the last meeting”
   - To support Service Composition
     - Integration of externally defined services
     - E.g. “Do a literature search”

Characteristics of Solution

- Metadata and Ontology Based
  => to cope with model heterogeneity
- Loosely Coupled
  => to cope with autonomy
- Runtime Based
  => to cope with dynamism
- Two Experiments:
  - Context Information Service
  - Semantic Matching Utility

Context Information Service: The Problem

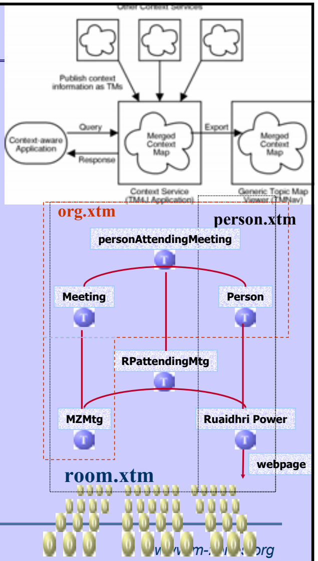
- How operate over wide diversity of context sources
  - Aggregate/derive upon demand?
  - Cache?
  - Periodic duplication?
- Range of quality of service requirements
  - Accuracy
  - Timeliness
- Cannot assume a single standard or API
- Range of client interaction styles envisaged
  - Querying
  - Publish/Subscribe

The Context Information Service: Characteristics of Environment

- Heterogeneous models
  - Raw context drawn from a range of sources (e.g. sensors, databases, applications, environment) and aggregated context generated
- Autonomous context control
  - E.g. user information controlled by user, organisational info controlled by organisation
- Dynamism
  - Within our target environment users/apps/devices which produce/consume context come and go, are in range/out of range, etc.

Initial Experiment TM4CM

- Topic maps potentially provide us a way to create a map that overlays the context information sources
  - Topic maps are mergeable
  - Two topics will be merged if
    - They share one identical base name in the same scope
    - They have the same subject identity
- Demonstrated dynamic creation of “meeting room” context topic map from different sources
- Source topic maps under control of different authorities
    - Personal details: maintained by person
    - Meeting details: maintained by organisation
    - Room usage details: maintained by room at runtime



Initial Results

- Ontology and Metadata Based
  - Metadata: topics and associations reasonable way to express concepts and their relationships
  - Ontology:
    - A Topic Map can be seen as form of ontology
    - Published Subject Indicators way to express common concepts
- Loosely Coupled
  - Successfully demonstrated that individual Topic Maps can be under control of autonomous context source authorities
- Runtime Based
  - Successfully manipulated and merged Topic Maps at runtime through the use of Application Programmer Interface (TM4J) and Query Language (Tolog)

Semantic Matching Utility: The Problem

- Our target environment assumes a wide range of applications/services interoperating seamlessly
- Reaching agreements on application/service interoperability standards will be impossible
- Services themselves will need to utilise knowledge encoded about meaning of terms (through ontologies) at runtime
- … even in circumstances where ontologies themselves are heterogeneous
  - Language
  - Authors

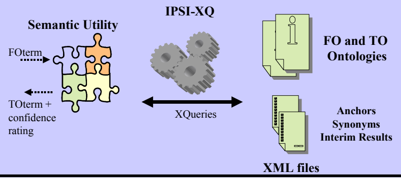
Semantic Matching Utility: Characteristics of Environment

- Heterogeneity of ontology models
  - Language Level Mismatches
  - Modelling Level Mismatches
- Autonomous
  - Within our target environment ontologies are developed and maintained independently from each other
- Dynamic deployment
  - Within our target environment:
    - Applications may require dynamic mediation of concepts/terms
    - The ontologies where concepts/terms defined will evolve

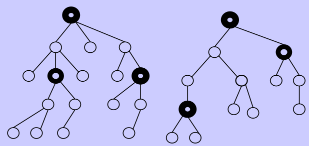
Initial Experiment SMUv1

- Given a term and a pointer to a Foreign Ontology (FO)
  - Return either matching Target Ontology (TO) term with confidence measure
- Does not support completely ad-hoc matching but on the other hand our approach does not require complete mapping between the ontologies
- Screening Guidelines
  - Ontologies amenable for processing by SMU?
  - Guidelines for SMU configuration info creation for the ontologies
    - Anchors, naming conventions etc.
- Implementation
  - Uses combination of term and ontology structure matching
  - Structure matching uses subclassing, associations and attributes to identify full or partial matches

Architecture



Anchor identification



Initial Results

- Ontology and Metadata Based
  - Metadata: need properties of an ontology exposed
    - Non-functional properties
      - e.g. comprehensiveness of ontology
    - Structural properties
      - E.g. density of relations used in the ontology
  - Ontology:
    - We can cope with heterogeneity of ontologies both at language and modelling mismatch levels
- Loosely Coupled
  - Demonstrated through the use of “Anchor Points”, XQueries and Synonym Database
- Runtime Based
  - Once the ontologies are successfully configured for use with the SMU

Future Work

- Context Information Service
  - Architecture
    - Hybrid of query driven and publish/subscribe?
  - Query processing
    - Processing queries given incomplete/possibly inaccurate information
    - Rapidly changing environment
- Semantic Matching Utility
  - Refinement of properties of ontologies, guidelines and tools that will allow for runtime semantic matching

Conclusions

- Our target environment requires us to bridge heterogeneous, autonomous, dynamic knowledge at runtime
- We believe that solutions that are ontology based, loosely coupled and runtime based are required
- Our TM4CM experiment has shown that the Topic Maps approach
  - exhibit these characteristics
  - has potential as the basis of the Context Information System
- Our SMU experiment has shown that
  - Semantic matching at runtime based on heterogeneous ontologies is feasible but requires some prescreening of the ontologies