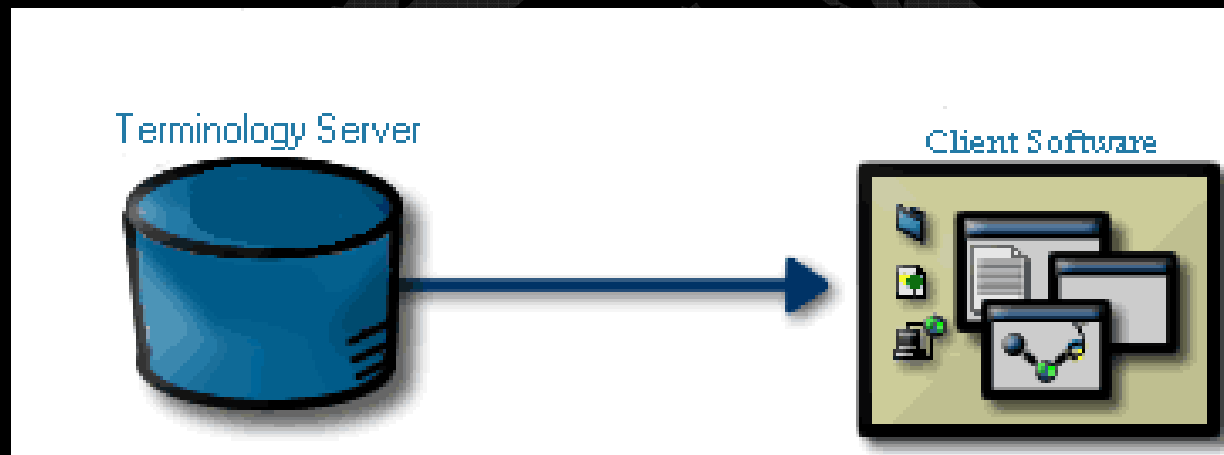


Deploying terminologies with a terminology server



Mike Hogarth, MD , Oracle Healthcare Applications

Overview

- Why a server
- Requirements
- Common information model
- jTerm – open source
- Oracle's ETS
- Lessons learned with ETS
- Open issues

Why a server?

■ Scalability

- UMLS, SNOMED RT, Read v3 = 3.4 million rows of data

■ Terminology management

- Versioning
- “Institutional” aliases (“local synonymy”)

■ Terminology support for thin devices?

- PDA, Pocket PC, Embedded PC's – monitoring equipment, Browser/Web clients

Terminology Server Requirements

■ *“enterprise friendly”*

- Ready for a networked enterprise (server)

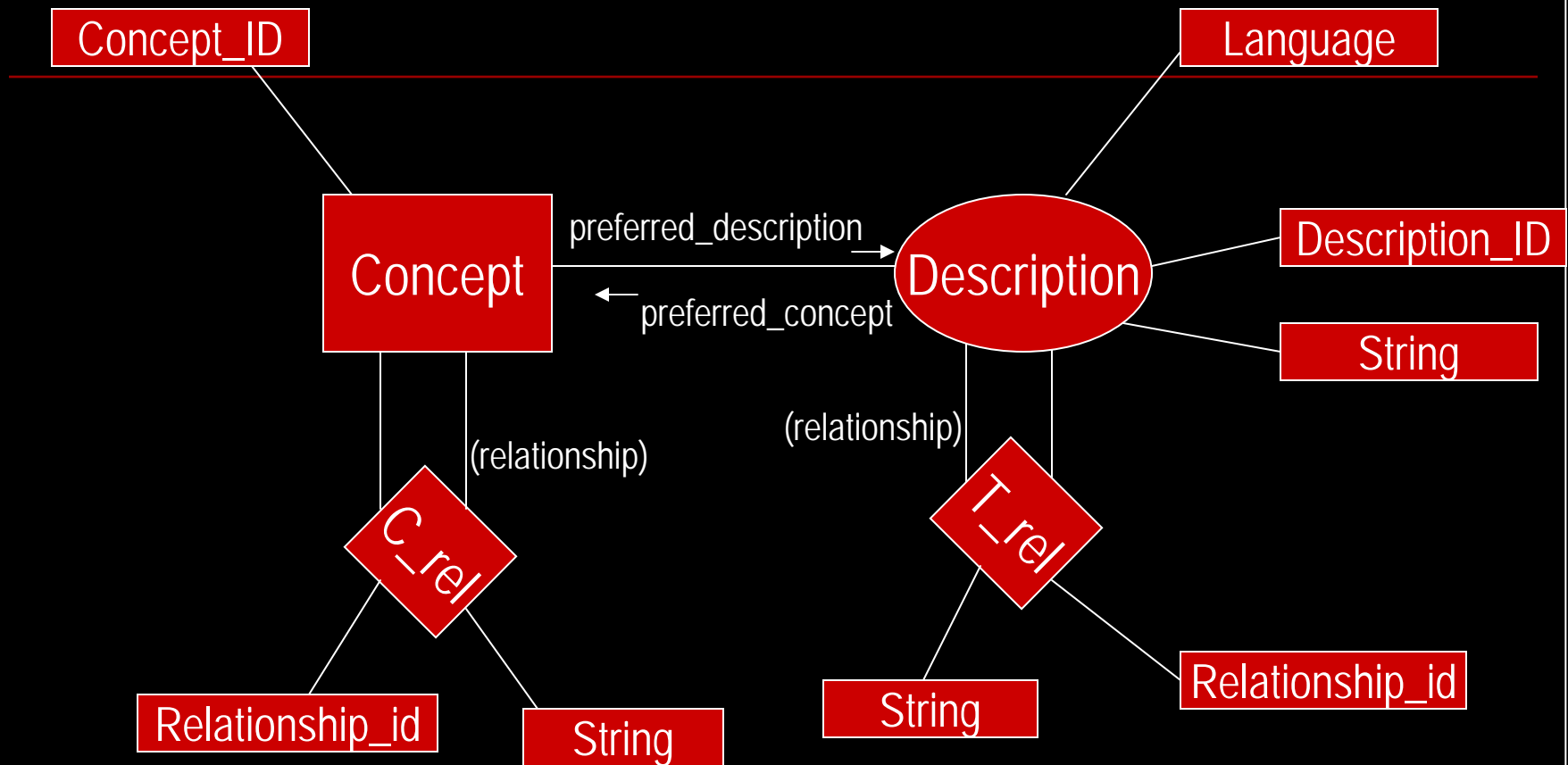
■ *“programmer friendly”*

- provides desired functionality in a simple way
- uses abstraction to reduce required knowledge of terminology system internals/specifics

■ *“application friendly”*

- High availability
- Results format is easily decomposed by the application

Start with an information model:



jTerm's Generic Terminology Information Model

Mappings to the generic model

Jterm Object Model	SNOMED-RT	NHS CT	ICD-9-CM
Concept	Concept	Concept	○
ConceptID	SnomedID	read_code	ICD Code
Description	Description	term_30	ICD text description
DescriptionID	DescriptionID	term_id	○
Relationship (multiple)	Relationship (multiple)	is_a only (hierarchical)	implied is_a
RelationshipID	RelationshipID	○	○

jTerm: An open source terminology server

- Uses a common terminology model
- Multi-platform, scalable (J2EE, multi-threaded)
- Open source under GNU LPL.
- Simple:
 - HTTP-based access mechanism
 - Any RDBMS as storage (can be remote)
 - Auto-loaders for raw terminology data (can load remotely)

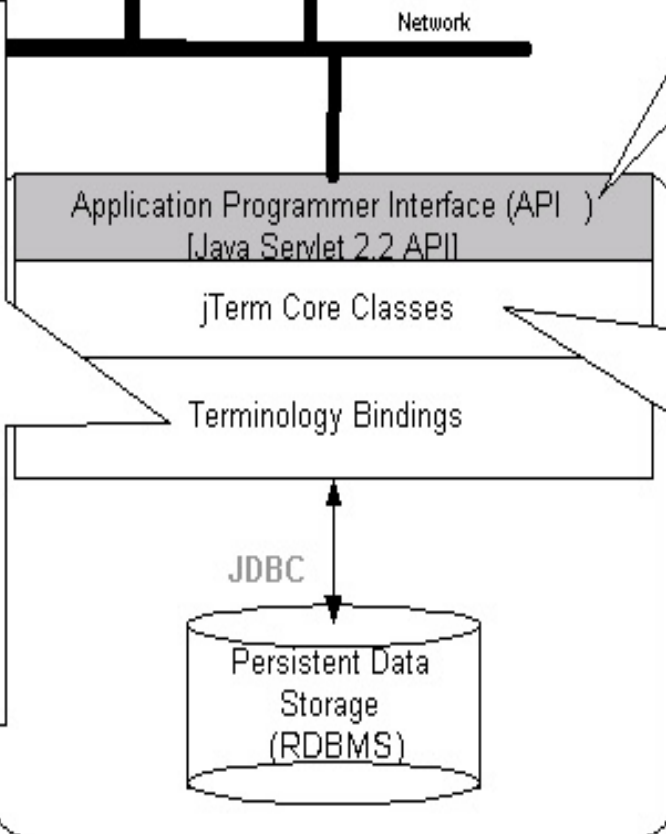
jTerm Architecture

Applications (external systems)



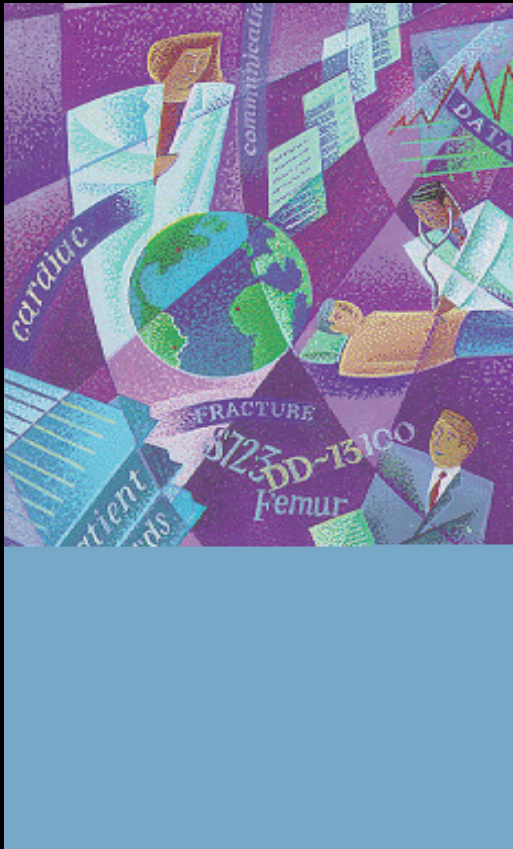
```
http://www.jterm.org/server/
handler?command=getTerm&param1=EXACT&param2
=Asthma
```

```
Snomed{
public Vector findTermString(
int search_type,
String SearchString)
throws SnomedException
{
Concept c = new Concept();
ConceptID cid = new
ConceptID();
if(search_type==Pattern.EXACT){
this.SQL = "SELECT * FROM
+DESCRIPTIONS_TABLE+
" WHERE Term
="+SearchString+"";}
}
.....
}
```



```
public class Term{
private TermID tid;
private ConceptID cid;
public String toXML(){
XMLRepresentation =
"<term>\n"+
"\tterm_id>"+
tid.toString()+"
</term_id>\n"+ "\n"
...etc...
this.RestrictionLevel+"</
term_restriction_level>\n"+
"</term>";
return this.XMLRepresentation;
}
```

jTerm Demonstration...



- Installer & Loader
- Term expansion -- synonymy
- Hierarchy navigation
- Conversion among coding systems (translation)
- Look at raw output

Using jTerm

- <http://www.jterm.org>
- What you need:
 - Java servlet runner (supporting v2.2 API)
 - At least 10 free ones
 - What I use: NT version of Apache's Jakarta (jakarta.apache.org)
 - REQUIRES support or 2.2—"web apps"
 - RDBMS + JDBC driver
 - What I use: NT version of MySQL (\$79), freeware MySQL JDBC driver
 - Terminology systems you want to use
 - Supported loaders: SNOMED RT, NHS CT, UMLS, LOINC

Enterprise Terminology System (ETS)

- A support component for Oracle's Healthcare Application Suite (*Oracle Healthcare Online*)
- Hosted model(ASP) or Intranet deployment
- Multi-terminology enabled (16 systems to date)
- Term/Concept queries, Ontology navigation
- *Domain/Topic functionality
 - Intersects with HL-7 v3 model
 - Provides query selection by "subject" rather than terminology system
- *Robust terminology system management
 - Loaders, verification, "quarantine", "activation"
 - Does not assume "good terminology practices" by terminology publishers

Lessons learned with ETS:

- Robust deployment architecture (jTerm vs. ETS)
- Which terminological systems are needed?
 - Functional requirements of the system – must be identified
 - billing
 - clinical documentation
 - linking to biomedical sources
 - decision support
 - workflow rules for processes
- Which systems are available in a practical sense?
 - Cost (CPT = \$10/user... who is a user?)
 - Licensing terms
 - lump sum fee/unlimited,
 - Is a metric required? – adds admin. and system overhead
- Updates
 - Variable terminology updates
 - NHS = monthly, ICD = annual, SNOMED = per version, etc...

Open issues

- Should “common functionality” and common output be defined rather than defining a common information model for terminological systems?
- Terminology system licensing is quite difficult
 - Possible solutions:
 - “single source” licensing (SNOMED RT + mappings)
 - Terminology vendor as Value Added Reseller (VAR).
- Terminology – HL7v3 Interface
 - To compose or pre-coordinate?
 - Terminology coverage “gaps”