University of Southern California

CiSoft

Chevron

avanade

Design and Implementation of a Semantic Web Solution for Real-time Reservoir Management

Ram Soma[2], Amol Bakshi[1], Kanwal Gupta[3],
Will Da Sie[2], Viktor Prasanna[1]

[1]University of Southern California, Los Angeles

[2]Chevron Corp.

[3]Avanade Inc.

About CiSoft

CiSoft: Center for Interactive Smart Oilfield Technologies
http://cisoft.usc.edu

Chevron

avanade

- USC-Chevron Center of Excellence for Research and Academic Training on Interactive Smart Oilfield Technologies
- **Established**: December 2003
- **Disciplines**: Petroleum Engineering, Chemical Engineering, Material Science, Physics, Computer Science, Electrical Engineering, Industrial Engineering
- **MS Degree** in Petroleum Engineering with emphasis on Smart Oilfield Technologies (SOFT)

RESEARCH AREAS	
• Integrated Asset Management	• Reservoir Management
• Well Productivity Improvement	• Data Management Tools
• Robotics and Artificial Intelligence	• Immersive Visualization
• Embedded and Networked Systems	

Outline

- Integrated Asset Management
  - Objectives
  - Role of semantic web
  - Software development methodology
- IAM Ontology
  - Ontology design
  - Change management and dirty queries
- Remarks
  - Lessons learnt
  - Areas of interest

CiSoft Integrated Asset Management (IAM)

Chevron

avanade

- What is IAM?
  - A **comprehensive transformation** approach to **integrated** oilfield operations
  - A software application that can help asset team members simulate decisions before making them
- Objectives
  - Increase integration between different functions
  - Enable asset level "what if" scenarios
  - Create a knowledge base of activities and decisions
  - Reduce risk and uncertainty in decision making
- Challenges
  - Data silos are not interoperable
  - Data is semi-structured
  - Multiple organizations and classes of users

CiSoft What IAM provides to users

Chevron

avanade

- Efficient access to data and information
  - Reduces time spent looking for data
  - Answers complex queries across semi-structured data sets
- Consistent view of information
  - Reconciles different views of the same information
  - Creates shared "situational awareness" of the asset
- Context of information creation and usage
  - Leads to more meaningful interpretation of data
  - Acts as organizational memory for the workflow
- Non-functional: Non-disruptive, extensible, scalable, usable, etc.

CiSoft

The IAM “Metacatalog”

Chevron

avanade

- Problem
  - Simulation models embody different realizations of uncertainty and development strategies for an asset
  - Models are created by different user groups at different times; it is difficult to maintain consistency of assumptions
  - No intuitive search functionality available to domain experts (“Show me most recently history matched model”)
- Solution: The IAM Metacatalog
  - Metadata repository at the core of the IAM application
  - Focus on answering “What does the data mean”? (vs. “How do I access the data”)
  - Key parameters and assumptions from various models are extracted and stored in the metacatalog
  - Also stores relationships between data objects and their provenance

Why Semantic Web Technologies

- Expressivity and richness of data model
- Organic growth capability for domain models/knowledge
- Inferencing and Rule Based Reasoning
- Flexibility of querying
- Ease of domain expert to understand and contribute to domain models
- Standards based (No vendor lock-in)
- Promoted by W3C

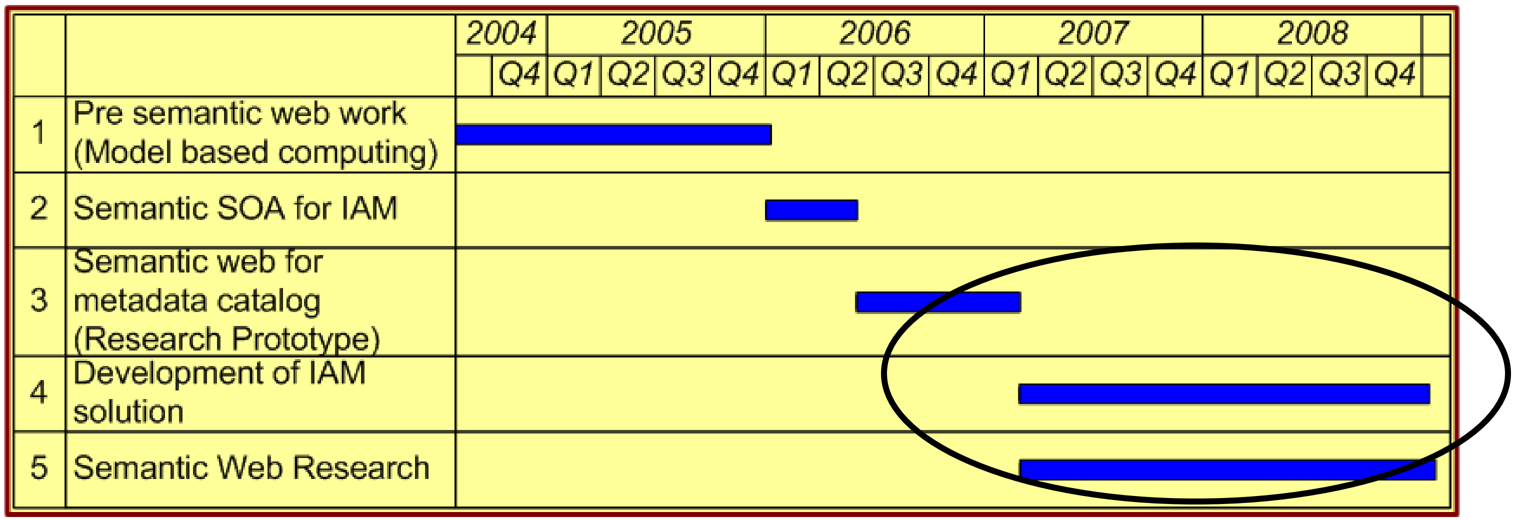
University of Southern California

CiSoft

IAM R&D Timeline

Chevron

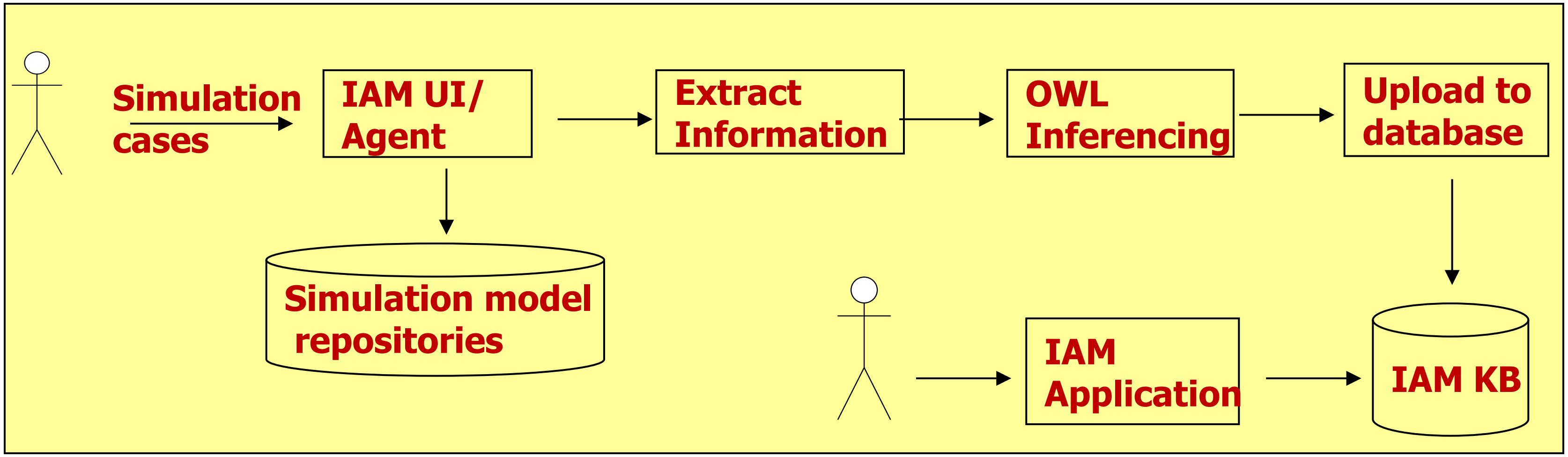
avanade



CiSoft

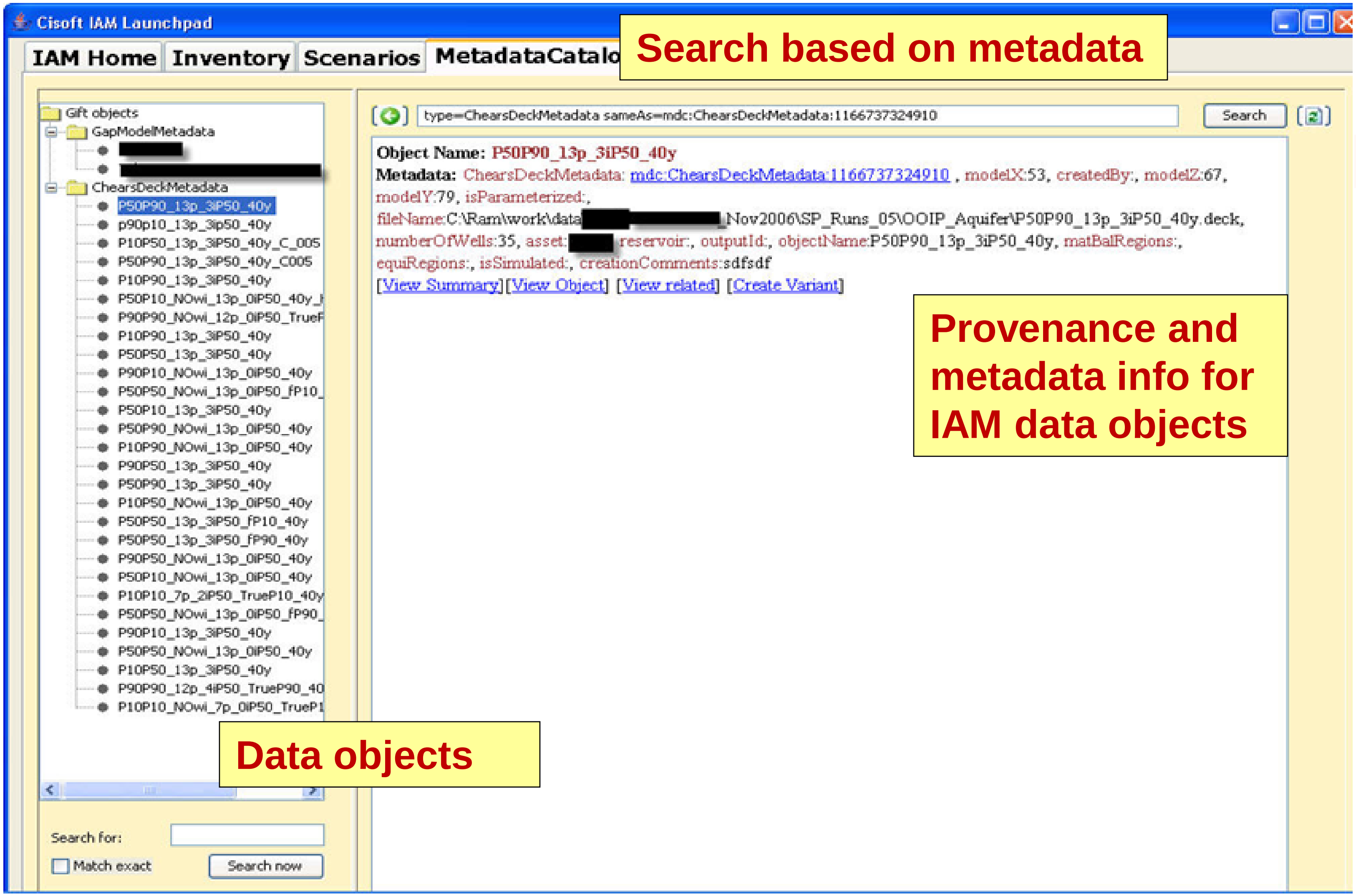
Populating the Metacatalog

- Most of the metadata is captured offline
- Metadata extraction by custom built parsers



CiSoft

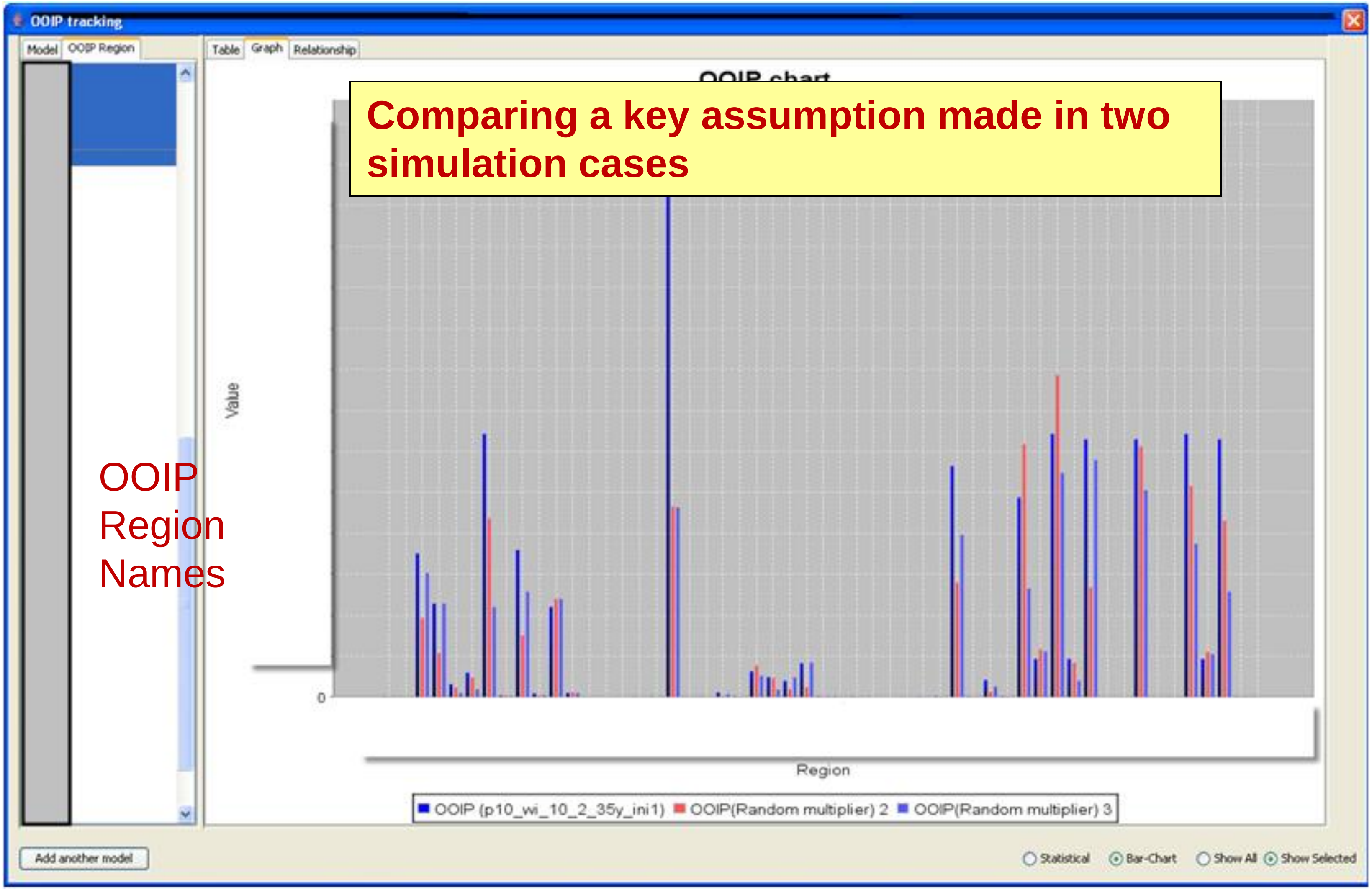
Example 1: Browse And Search



Chevron

avanade

Example 2: Comparing Assumptions



Comparing a key assumption made in two simulation cases

OOIP Region Names

Software Development Methodology

- Agile development using Scrum
  - Iterative software development in “Sprints”
  - Close collaboration with customer
    - Reviews/demos after each sprint
    - Flexible prioritization at sprint boundaries
    - “Product Owner” role represents the stakeholders
  - Less focus on formal documentation

CiSoft

Phases

- Development in sprints

Requirements → Planning → Ontology Spec/ Refactoring ⇄ Review → Application Dev

- Observation: Ontology frequently modified
  - Techniques for change management make methodology more successful

CiSoft Miscellaneous

- Addressed key risks of an OWL-based solution
  - Performance - Benchmarking
  - Limited tool support – Web service interfaces for KB
  - Ongoing evaluation of alternatives
- Tech transfer to software developers
  - Code and documentation
  - Demos and training
- Development
  - Ontology design was done with the assistance of domain experts and end users
  - CiSoft researcher acting as “Product Owner” for Scrum team moved research into deployment

Outline

- Integrated Asset Management
  - Objectives
  - Role of semantic web
  - Software development methodology
- IAM Ontology
  - Ontology design
  - Change management and dirty queries
- Remarks
  - Lessons learnt
  - Areas of interest

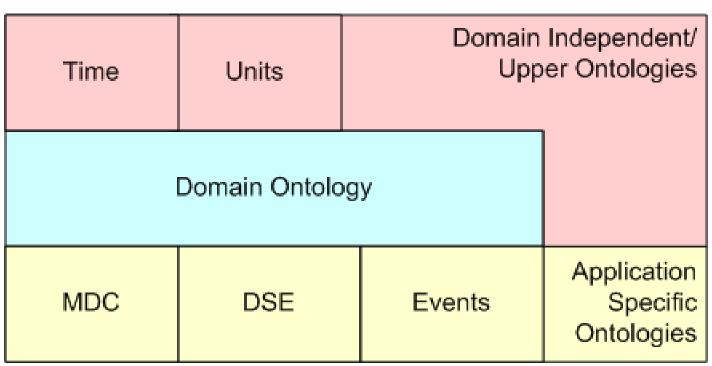
CiSoft

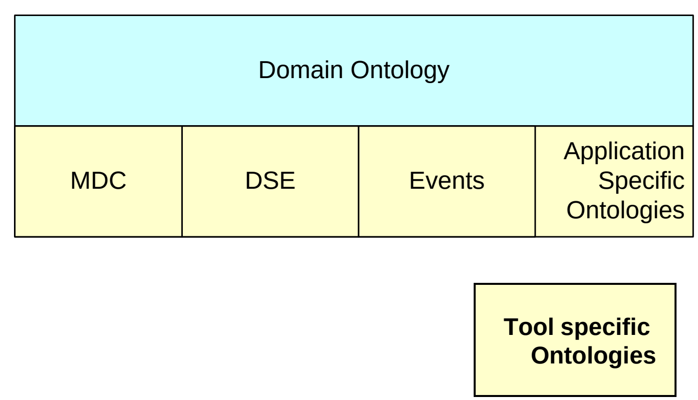
Ontology Design

Chevron

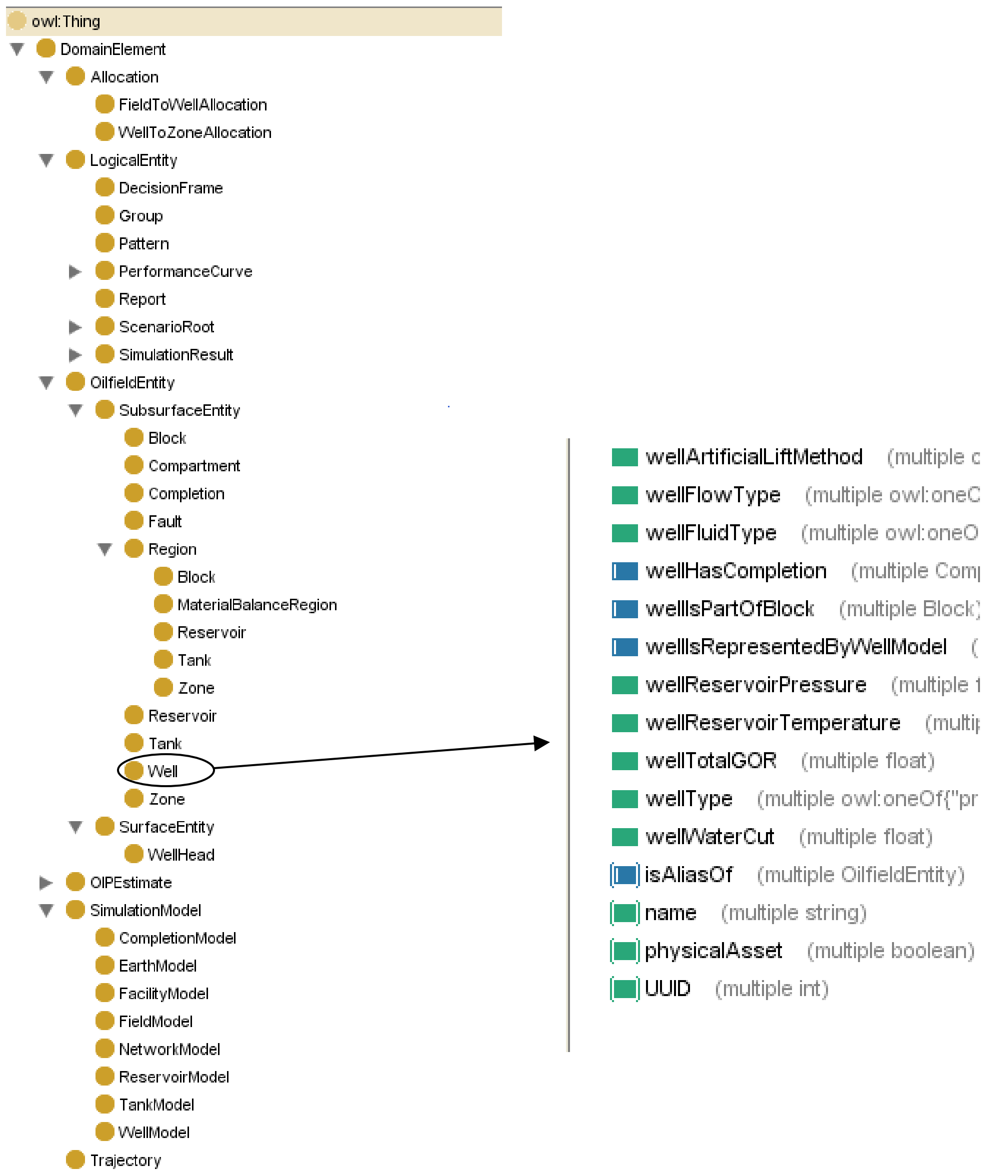
avanade

- Ontology design divided into three levels to improve modularity
- Domain independent/Upper ontologies
  - Concepts common to all ontologies like time, units etc.
- Domain ontology
  - Model of the elements in the asset
  - Uses elements from upper ontologies
- Application specific ontologies:
  - Elements specific to a given application or workflow
  - Uses elements from upper and domain ontologies





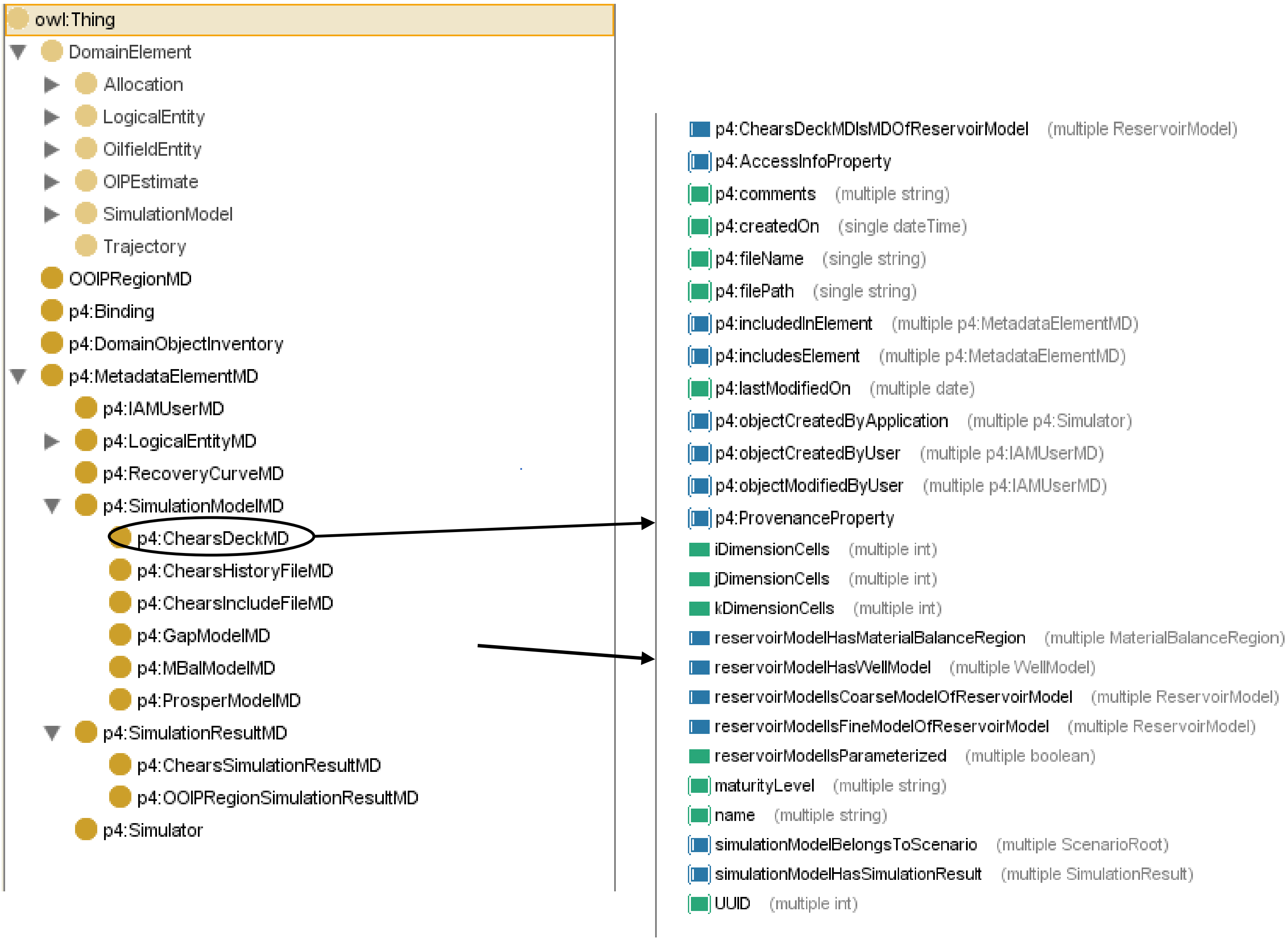
CiSoft IAM Ontologies: Domain Ontology



Chevron

avanade

CiSoft IAM Ontologies: Metadata ontology

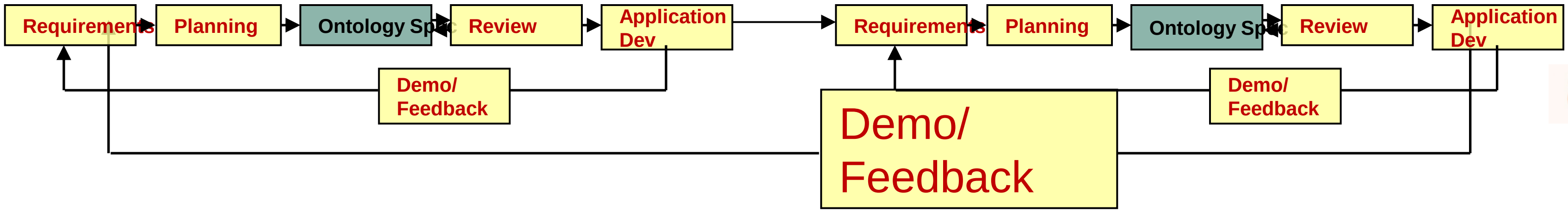


CiSoft

Implementation

- OWL data store + SPARQL querying
- Current implementation uses Jena OWL API
  - Two reasoners
    - Rule based reasoner (Jena)
    - Tableaux reasoner (Pellet)
  - OWL data stored in Jena RDBMS, file system
- Web service API to abstract data store (Apache Axis2)
- Various applications that use MDC

Supporting Iterative Development



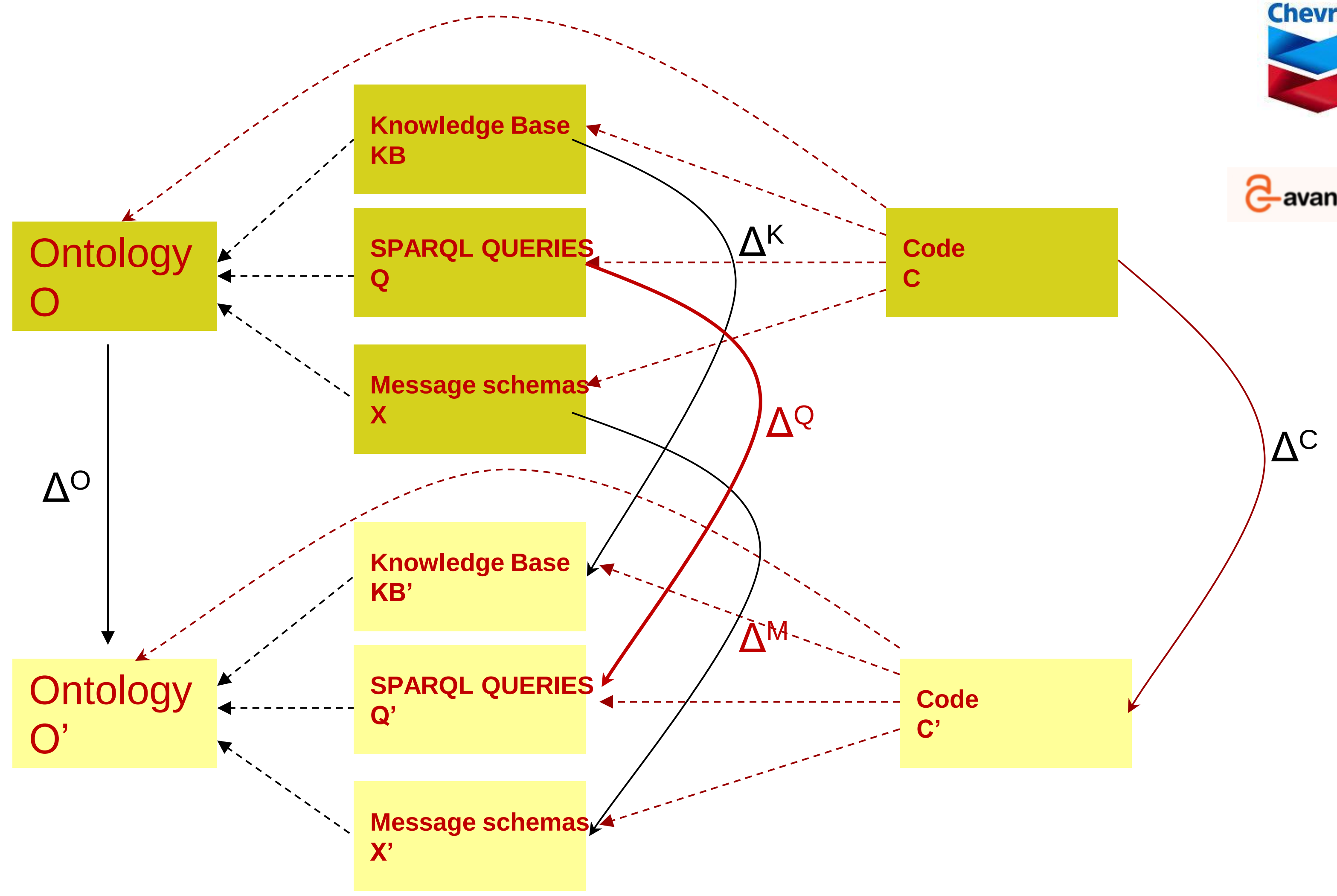
- Ontologies are modified in every sprint

CiSoft

Change Management Problem

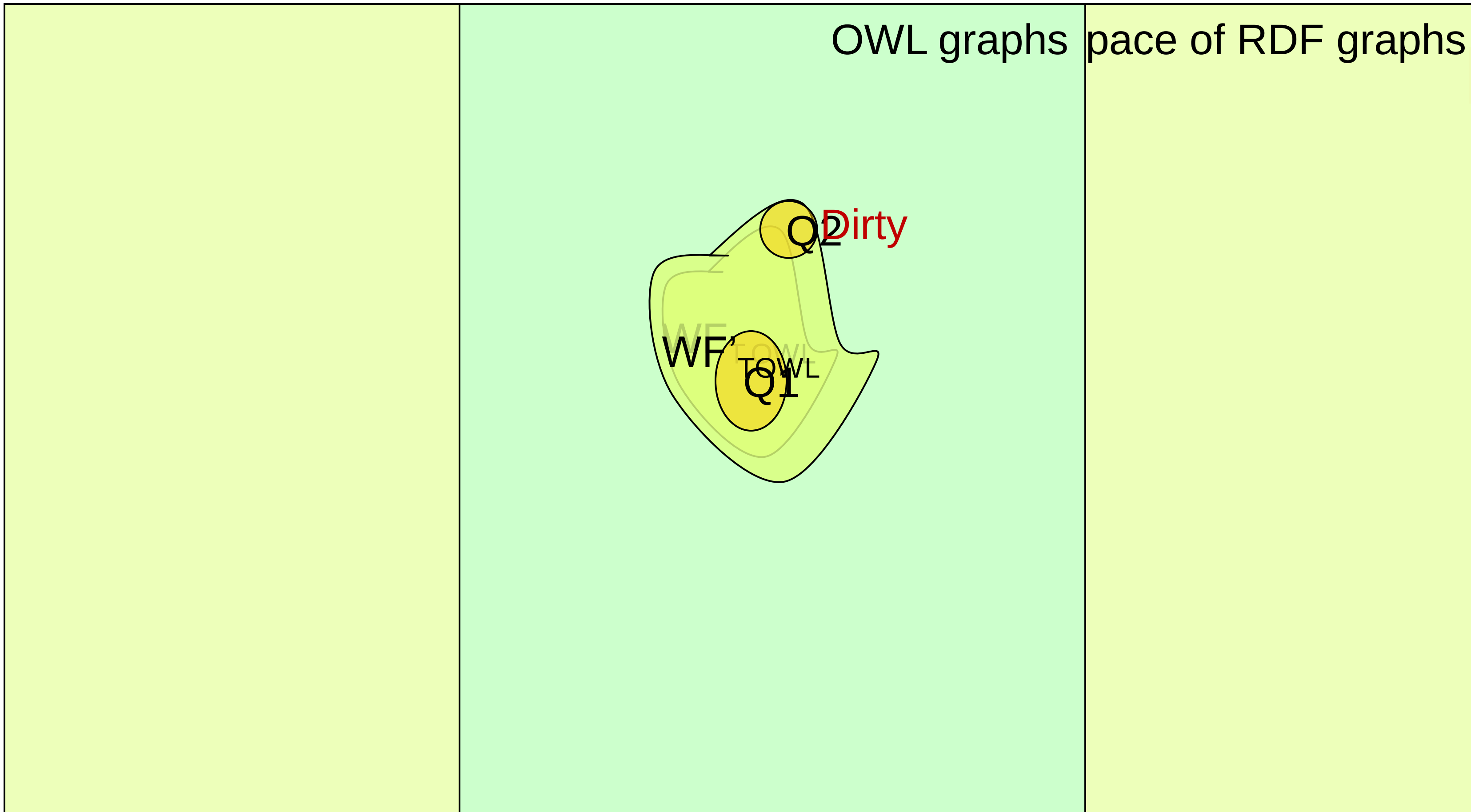
Chevron

avanade



Detect **dirty queries** that are invalidated when an ontology is modified

Dirty Queries



$$\mathrm{EXT}_{T'}(Q) \cap (WF'_{T,OWL} \setminus WF_{T,OWL}) != \Phi$$

$$\vee\ \mathrm{EXT}_{T}(Q) \cap (WF_{T,OWL} \setminus WF'_{T,OWL}) != \Phi$$

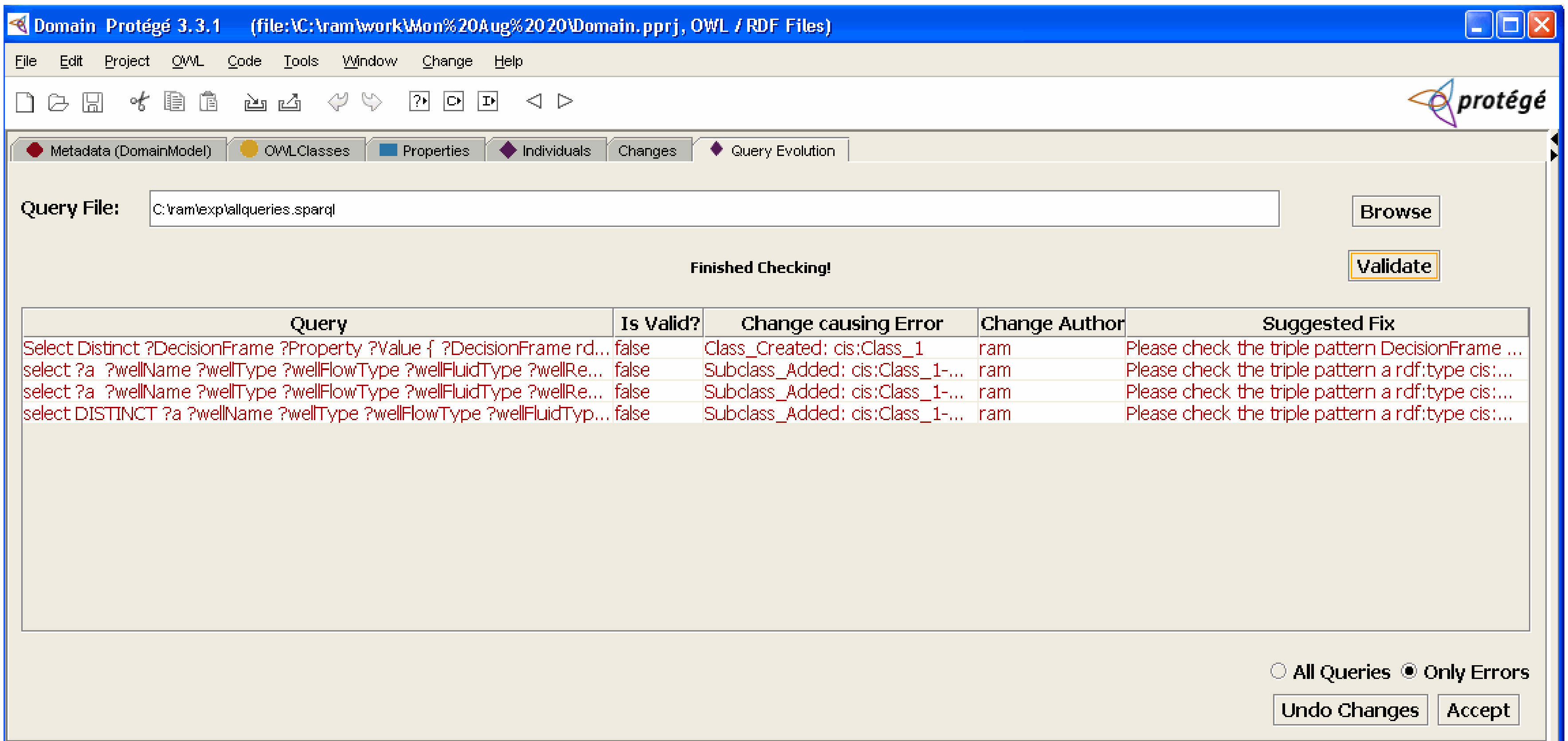
Change handling

- Detect ontology changes
- Evaluate Query, EXT(Q)
- Compute the impact/semantics of changes, $WF'_{T,OWL} \setminus WF_{T,OWL}$
- Match query and changes

Capture Change → Query Evaluation → Semantics of Change → Detect Dirty Queries

Implementation

- Protégé plugin
  - Jena, Pellet, SPARQL parser



CiSoft Outline

- Integrated Asset Management
  - Objectives
  - Role of semantic web
  - Software development methodology
- IAM Ontology
  - Ontology design
  - Change management and dirty queries
- Remarks
  - Lessons learnt
  - Areas of interest

CiSoft

Lessons learnt

- Ontology design
  - Plan schema changes carefully and do not change schema often
  - Keep OWL ontology small and modular; use OWL imports
- Performance
  - Track performance through product development cycle
  - Consider performance enhancing components (caching) in architecture
- Be cognizant of OWL features your tool supports
  - Very few are fully compliant with standards
- Design for change
  - Use SPARQL querying
  - Separate KB querying components from business logic and UI
  - Active area of work- expect big improvements soon

Features we missed

- SPARQL
  - Rollup/aggregation queries. E.g. get the aggregate of OOIP for region as sum of OOIPs of contained regions
  - Results as triples
  - XPath like expressions. E.g. get sub-tree under X
- Updating materialized OWL knowledge bases
  - *Solved problem* in research
- Better XML-OWL/RDF interoperability
  - SPARQL-XML (?)
  - OWL/RDF- XML (Gloze)

Areas of interest

- Ontology extension
  - Modeling events
  - Capturing data provenance
- Performance improvements
  - Developing representative benchmarks
  - Evaluating various RDF triple stores
  - Algorithms for parallel OWL inferencing
- Change management
  - Managing evolution of schema and instance data
  - Efficient techniques to track changes to OWL KBs

Some of our publications

- R. Soma, Viktor Prasanna, Detecting dirty queries during iterative development of OWL-based applications, 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2008), Monterrey, Mexico, Nov 11 - 13, 2008.
- R. Soma, Viktor Prasanna, Parallel Inferencing for OWL Knowledge Bases, International Conference on Parallel Processing (ICPP-2008), September 2008.
- R. Soma, Viktor Prasanna, A Data Partitioning Approach for Parallelizing Rule Based Inferencing for Materialized OWL Knowledge Bases, International Conference on Parallel and Distributed Computing and Communication Systems (PDCCS), September 2008.
- R. Soma, Amol Bakshi, Viktor Prasanna, W. DaSie and B. Bourgeois, Semantic-web technologies for Oil-field Management, SPE Intelligent Energy Conference and Exhibition, April 2008.
- R. Soma, Amol Bakshi, Viktor Prasanna, A Semantic Framework for Integrated Asset Management, Proceedings of The Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid), 2007
- R. Soma , A. Bakshi, V. K. Prasanna, and W. Da Sie, A Model-Based Framework for Developing and Deploying Data Aggregation Workflows, 4th International Conference on Service Oriented Computing (ICSOC), December 2006.

Backup

University of Southern California

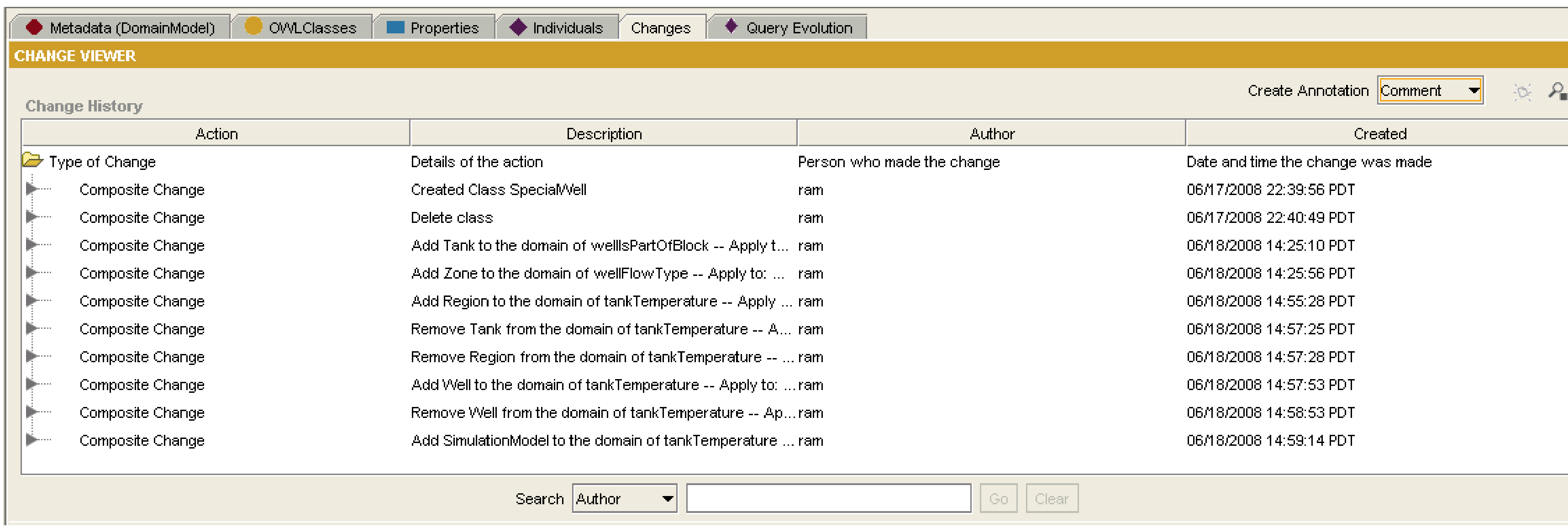
CiSoft

Change capture

- Well studied problem
  - All changes to OWL, representation, capture..
- Use Protégé plugin

Chevron

avanade



Action	Description	Author	Created
Type of Change	Details of the action	Person who made the change	Date and time the change was made
Composite Change	Created Class SpecialWell	ram	06/17/2008 22:39:56 PDT
Composite Change	Delete class	ram	06/17/2008 22:40:49 PDT
Composite Change	Add Tank to the domain of wellIsPartOfBlock -- Apply t...	ram	06/18/2008 14:25:10 PDT
Composite Change	Add Zone to the domain of wellFlowType -- Apply to: ...	ram	06/18/2008 14:25:56 PDT
Composite Change	Add Region to the domain of tankTemperature -- Apply ...	ram	06/18/2008 14:55:28 PDT
Composite Change	Remove Tank from the domain of tankTemperature -- A...	ram	06/18/2008 14:57:25 PDT
Composite Change	Remove Region from the domain of tankTemperature -- ...	ram	06/18/2008 14:57:28 PDT
Composite Change	Add Well to the domain of tankTemperature -- Apply to: ...	ram	06/18/2008 14:57:53 PDT
Composite Change	Remove Well from the domain of tankTemperature -- Ap...	ram	06/18/2008 14:58:53 PDT
Composite Change	Add SimulationModel to the domain of tankTemperature ...	ram	06/18/2008 14:59:14 PDT

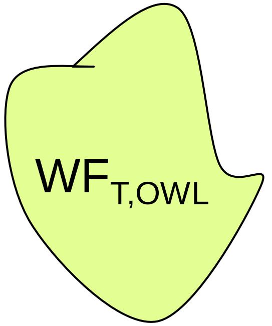
CiSoft

Query evaluation

- Evaluate triple patterns (TP)
  - "Projecting" TP to $WF_{T,OWL}$
  - Observations:
    - All OWL statements are either *type*, *property* or *identity* assertions
    - Triple pattern can have variable or constant in each of its 3 places: 2*2*2= 8 types of triple patterns
- Evaluate graph pattern
  - Based on semantics of connectors

CiSoft Semantics of change

- Not all changes modify WF
  - Lexical Changes: Names of entities, properties, easy to handle
  - Extensional: Modifies WF
  - Assertional: Does not change WF but adds rules
  - Cardinality: Does not change WF but adds/removes constraints
- Determine $WF'_{OWL} \setminus WF_{OWL}$ from changes
  - About 50 kinds of changes to OWL ontology



Object	Operation	Argument(s)	Semantics of Change
Ontology	Add_Class	Class definition (C)	$IOC \neq IOC'$
Ontology	Remove_Class	Class ID (C)	$IOC \neq IOC'$, $CEXT(SC) \neq CEXT'(SC)$ $CEXT(Dom(P)) \neq CEXT'(Dom(P))$, $CEXT \neq CEXT'(Ran(P)) \forall P \mid C \in Dom(P)$ or $Ran(P)$
Class (C)	Add_SuperClass	Class ID (SC)	$CEXT(SC) \neq CEXT'(SC)$
Class(C)	Remove_SuperClass	Class ID (SC)	$CEXT(SC) \neq CEXT'(SC)$
Property (P)	Set_Transitivity	Property ID	- (Assertional Change)
Property (P)	UnSet_Transitivity	Property ID	- (Assertional Change)

Matching

- Aggregate changes
- Handle Lexical change: String search/replace
- Compare extension of query with semantics of change
  - If they have some element in common ➔ dirty
  - E.g. EXT(Q) = P (ALL_Persons X rdf:type X Person) U P(ALL_Persons X IOP X I U L)
  - Sem(ch) = {ALL_Persons’ != ALL_Persons}