(Meta) Meta Model Extensions for Managability of Large Scale Collaborative Modeling

Graham McLeod
June 2008
at EMMSAD'08

CTO **+ PROMIS** We deliver Solutions!

Principal **inspired!**
IT ● Consulting ● Training ● Research ● Tools

Abstract

There is ample evidence to suggest that collaborative modeling offers significant advantages over modeling carried out by individuals. Collaborative modeling can be achieved by workshops and other interactive techniques. Recently there has been increasing interest in supporting collaborative modeling with web and repository based tools, especially where the desired participants are separated by distance and time zones and potentially language.

This paper introduces and formalises some constructs and extensions to meta models (and meta meta models) which have been found useful in enhancing the usefulness of large scale collaborative modeling tools and the manageability of the models employed in support of enterprise architecture management.
Issues addressed include: subject domain, ownership, authority, context, time, version, status, multiple opinions, user groups/roles, multiple languages and avoiding information overload.

We Will Look At

- Collaborative Modeling
- The Context for Our Work
- Challenges Encountered
- Potential Solutions
- Process
- Experience and Findings
- Resulting Model
- Remarks and Future Work
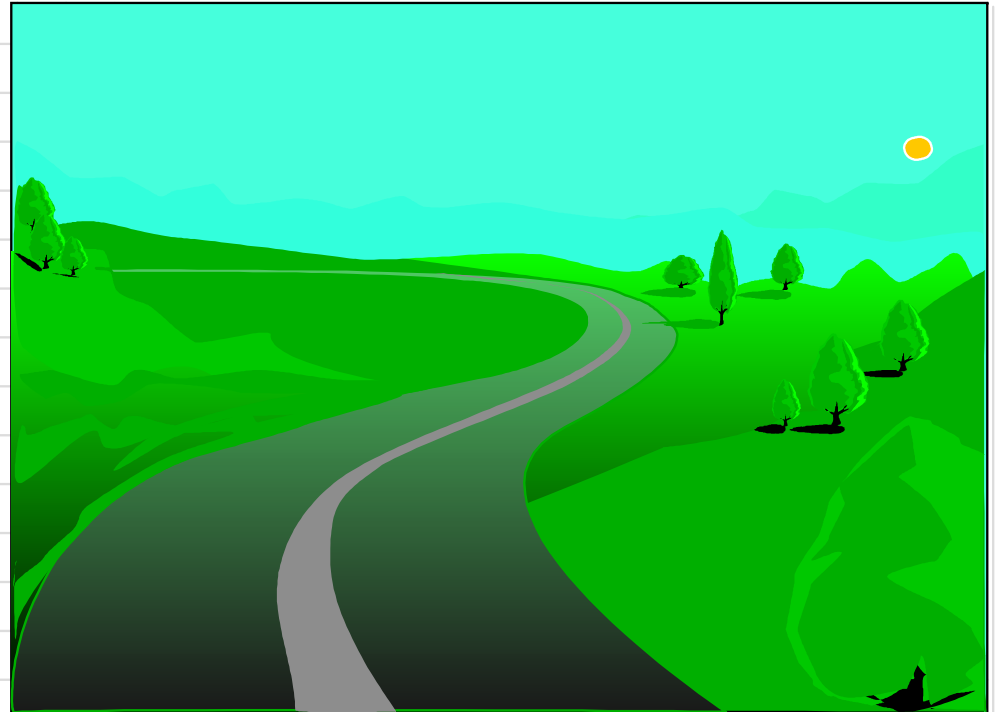
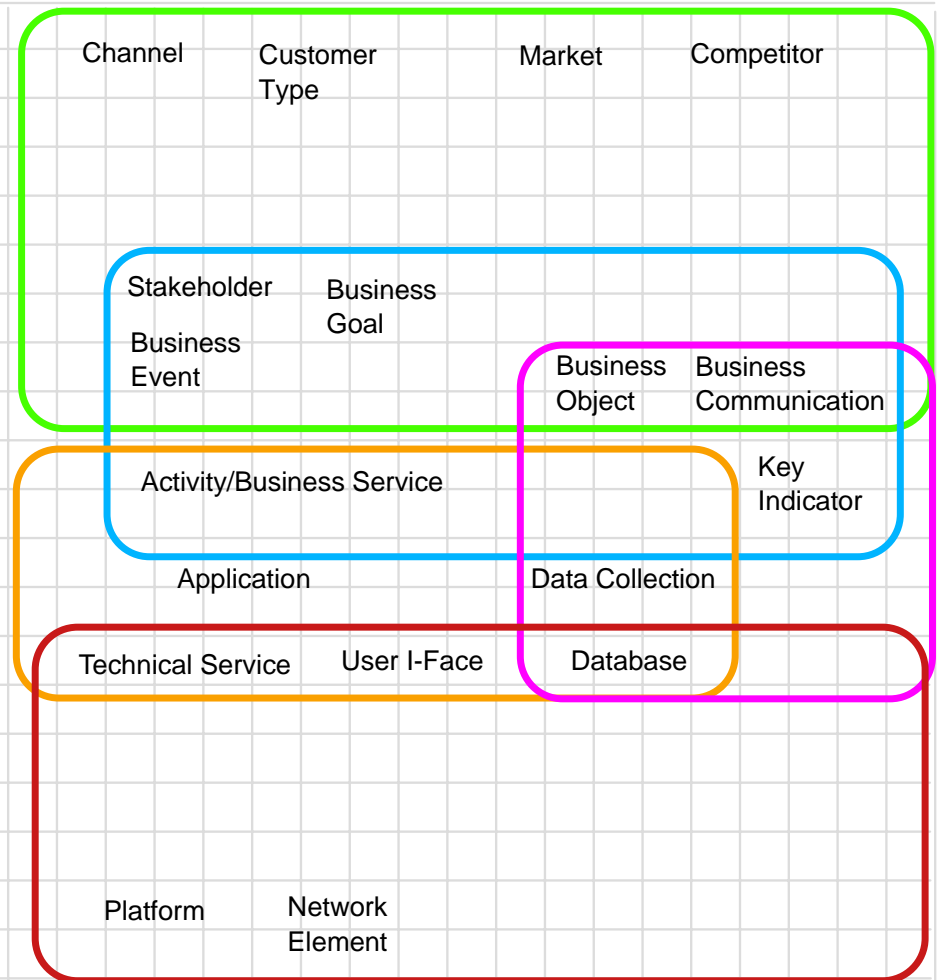Note: This is a very pragmatic study done in an industry setting

inspired!
IT ● Consulting ● Training ● Research ● Tools

Collaborative Modeling

- Where the end model is produced
  - ► By more than one modeler
  - ► Over a period of time
  - ► Possibly by
    - – Modelers who are separated by distance, language, or time
    - – Communities who share different representations or viewpoints but generally similar semantics, at least where their worlds overlap
- The full model may be composed of several overlapping models of intersecting concerns
  - ► Different stakeholders may have different rights to elements of different type within their models

Coverage of Meta Model for EA

- **Domains**
  - ▶ Business
  - ▶ Process
  - ▶ Application
  - ▶ Information
  - ▶ Technology
- **Cross Cutting Concerns**
  - ▶ Risk
  - ▶ Cost
  - ▶ Quality
- **Housekeeping**
  - ▶ Security and Permissions
  - ▶ Timeframe and Status
- **Model Types / Viewpoints**
  - ▶ Audience
  - ▶ Content
  - ▶ Notation

Channel Customer Type   Market   Competitor

Stakeholder   Business Goal

Business Event

Business Object   Business Communication

Activity/Business Service

Key Indicator

Application   Data Collection

Technical Service   User I-Face   Database

Platform   Network Element

inspired!
IT ● Consulting ● Training ● Research ● Tools
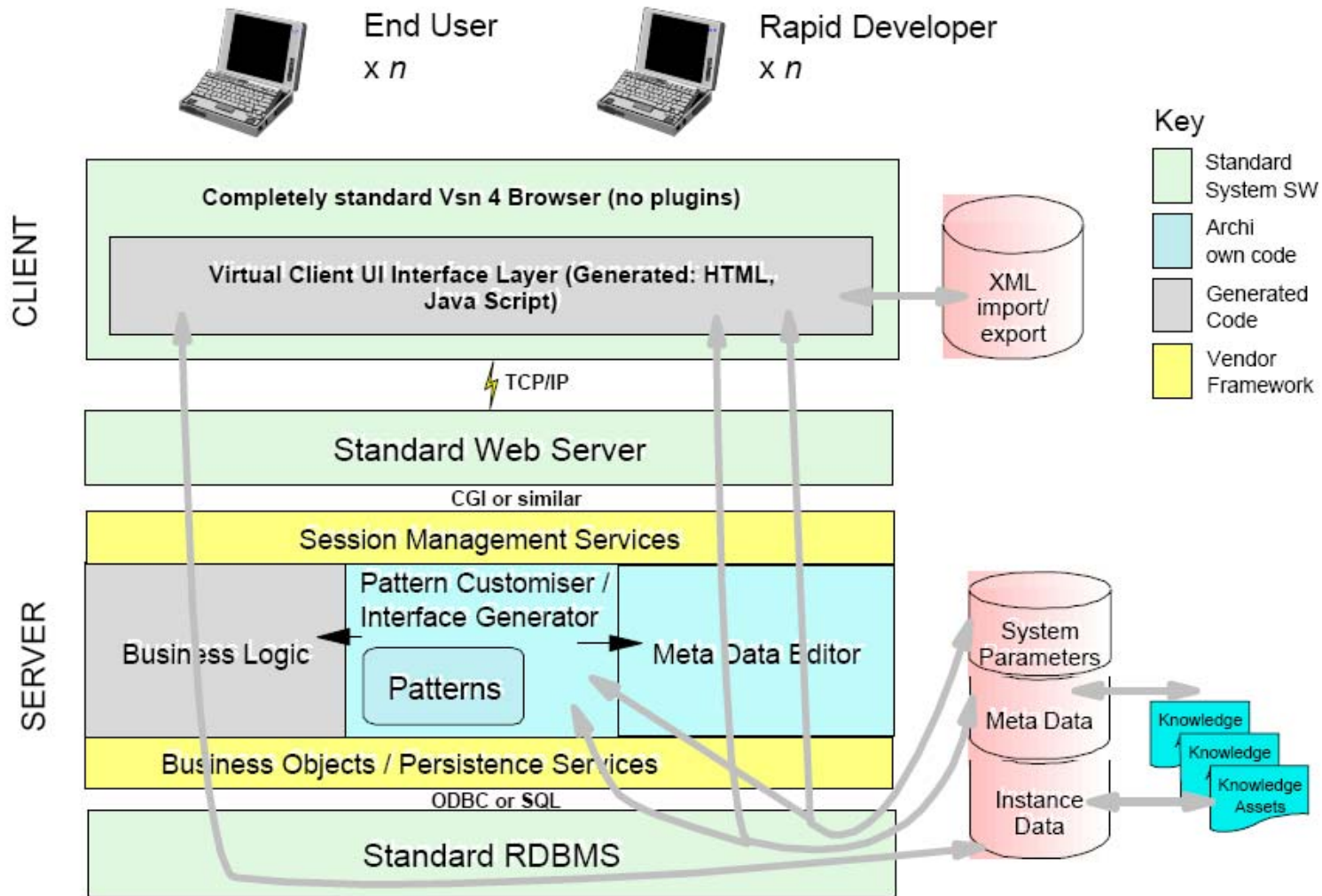
- **Inspired**
- **PROMIS Solutions AG**
- **Archi/EA Webmodeler => Now EVA Netmodeler**
  - ► Collaborative, web based enterprise modeling tool and repository
  - ► Flexible meta model
  - ► Multi- User, Notation, Framework, Perspective
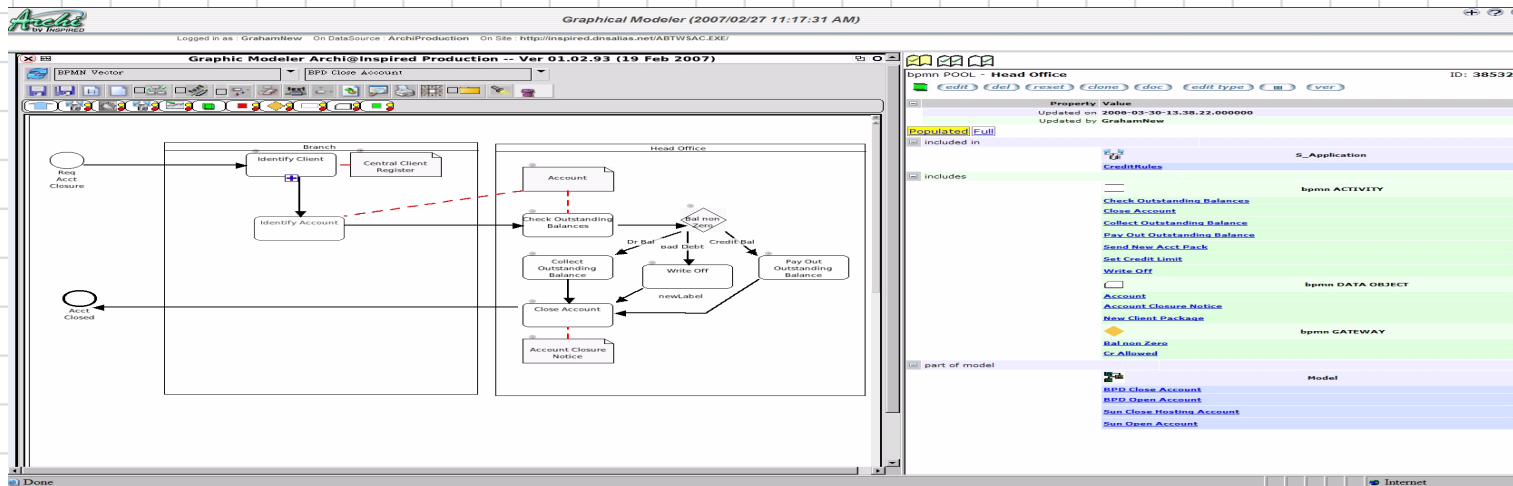
- **Large Clients/Projects**
- **Wide range of models and concepts**
- **High volumes**

inspired!
IT ● Consulting ● Training ● Research ● Tools

PROMIS
We deliver Solutions!

Archi **Architecture**

End User x *n*

Rapid Developer x *n*

CLIENT

Completely standard Vsn 4 Browser (no plugins)

Virtual Client UI Interface Layer (Generated: HTML, Java Script)

XML import/ export

Key

- Standard System SW
- Archi own code
- Generated Code
- Vendor Framework

⚡ TCP/IP

Standard Web Server

CGI or similar

SERVER

Session Management Services

Pattern Customiser / Interface Generator

Business Logic

Patterns

Meta Data Editor

Business Objects / Persistence Services

ODBC or SQL

Standard RDBMS

System Parameters

Meta Data

Instance Data

Knowledge

Knowledge

Knowledge Assets

Copyright 2000 *Inspired*

inspired!
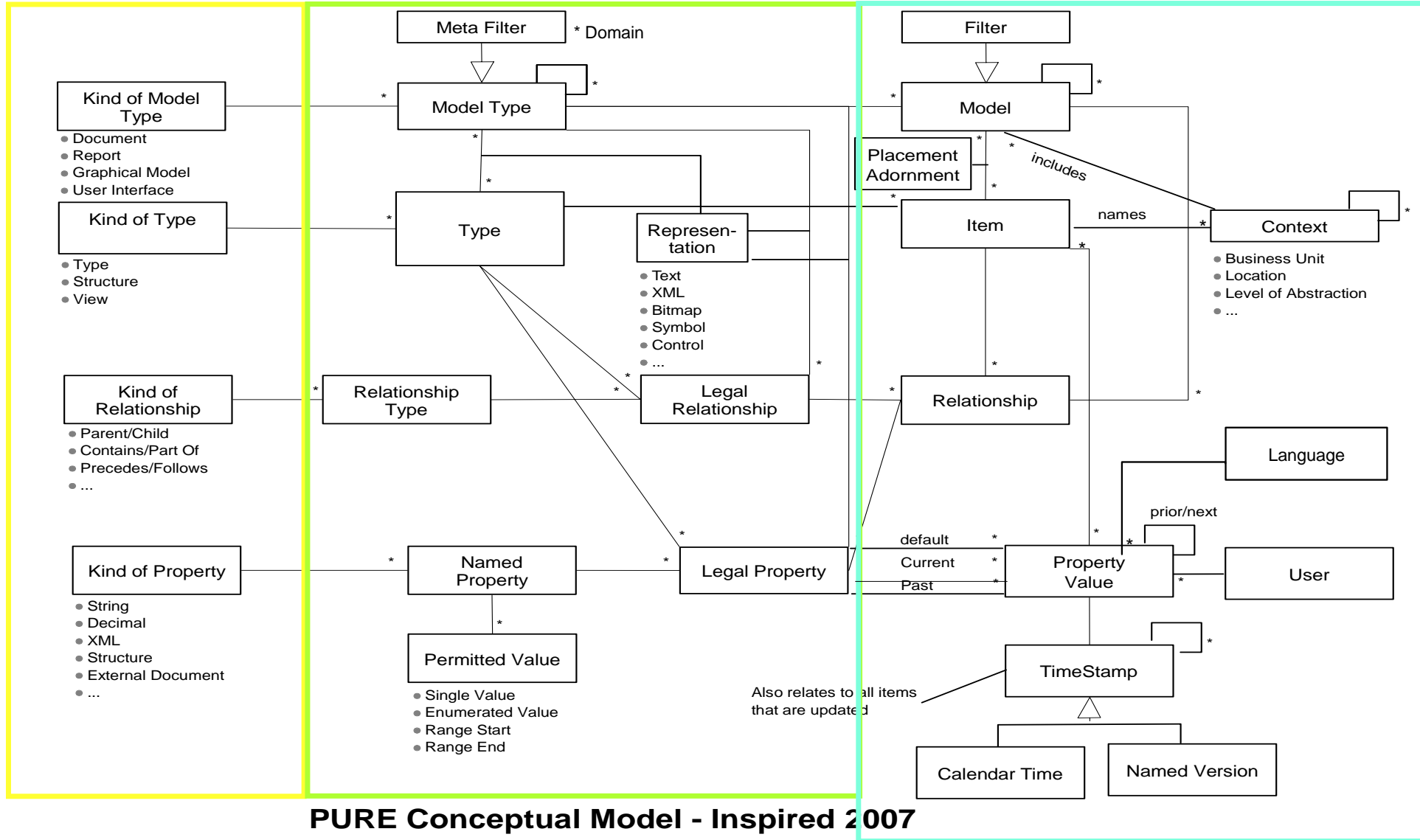IT ● Consulting ● Training ● Research ● Tools

- Volumes and Performance
- Managing Ownership of Objects and Rights
- Avoiding Information Overload in the User Interface
- Managing Local and Global Views
- Presenting Information in Best way for Different Groups
- Supporting Different "Versions of the Truth"
- Dealing with Same Semantics in Different Languages
- Variable Information Quality or Status

- Domains
- Caching and Inherent Hierarchies
- Automated "Chunking"
- Context
- Filters
    - ► Domains
    - ► Items, based upon
        - – Attribute values
        - – Relatedness
        - – Hierarchy (level, parent, related to subtree)
    - ► User Input during execution
    - ► Context
    - ► Status
    - ► User (creating, relating, modifying, deleting)
    - ► Timeframe
    - ► Sampling

Potential Solutions Cont'd

- Opinions
- Representation as an Abstraction
- Model Types
- Model
- Project
- Documents
- Versions
- Scenarios
- Packages
- Journaling

inspired!
IT ● Consulting ● Training ● Research ● Tools

- **Reviewed work of**
  - ▶ Martin Fowler, esp Domain and Design Patterns
  - ▶ Pieter Wisse on context and time in Information Systems
  - ▶ Lukas Renggli on the meta-described web development framework, Magritte
- **Extensive Smalltalk Experience**
  - ▶ VAST - IBM/Instantiatoins
  - ▶ Squeak - Public Domain
- **Prototyped core ideas and populated with sample data from production repositories for today's tool**
- **Incrementally built meta model to support requirements**
- **Accessed effectiveness in addressing issues, meeting requirements**

Experience and Findings

- Three layer object architecture vital (at conceptual level)
  - ► Even though higher two levels translate to Smalltalk classes
- Context is powerful and reusable for multiple purposes and moves to the meta meta level
- Relationship Typing Introduced
- Model Type absorbs concepts of
  - ► Graphical Model
  - ► Document
  - ► Report
  - ► User Interface
- Filters as a specialisationof Model Type (with exclusion rather than assembly behaviour)
- Time, Version, Baseline tracking unified via relationships
- State implemented as a property filter or context

PURE Conceptual Model - Inspired 2007

Diagram entities and labels:

- **Meta Filter** — * Domain
- **Kind of Model Type**
  - Document
  - Report
  - Graphical Model
  - User Interface
- **Model Type**
- **Filter**
- **Model**
- **Kind of Type**
  - Type
  - Structure
  - View
- **Type**
- **Representation**
  - Text
  - XML
  - Bitmap
  - Symbol
  - Control
  - ...
- **Placement Adornment**
- **Item** — names
- **Context**
  - Business Unit
  - Location
  - Level of Abstraction
  - ...
- **includes**
- **Kind of Relationship**
  - Parent/Child
  - Contains/Part Of
  - Precedes/Follows
  - ...
- **Relationship Type**
- **Legal Relationship**
- **Relationship**
- **Language**
- **Kind of Property**
  - String
  - Decimal
  - XML
  - Structure
  - External Document
  - ...
- **Named Property**
- **Legal Property** — default, Current, Past
- **Property Value** — prior/next
- **User**
- **Permitted Value**
  - Single Value
  - Enumerated Value
  - Range Start
  - Range End
- **TimeStamp**
- Also relates to all items that are updated
- **Calendar Time**
- **Named Version**

Remarks and Future Work

- **Remarks:**
  - ▶ Model has proven very flexible and expressive
  - ▶ Challenging to implement with good performance
    - Exploiting Smalltalk class creation at runtime for ease of business logic and performance vs earlier persistence layer with much assembly
    - Collections used heavily and proving effective with large execution memory
    - Experimenting with set operations to improve filter performance - encouraging

- **Future work**
  - ▶ Defining business logic patterns in more generic way
  - ▶ Using representation model to drive User Interface Generation for complex types
  - ▶ For language designers: Maybe provide direct support for the concept or context (ala Wisse) more directly in languages (OK, some support namespaces, but this is a one to one concept). Maybe an extension to the Smalltalk dependency mechanism?

PROMIS
We deliver Solutions!

- ## Graham McLeod
  graham@inspired.org
  graham.mcleod@pro-mis.com
  +27 82 578 1834

- ## Inspired
  www.inspired.org
  +27 21 531 5404

- ## PROMIS Solutions AG
  www.pro-mis.com
  +41 (41) 7 60 73 88

Graham McLeod has been involved in IS practice and teaching for over thirty years. He has produced advanced methods for systems delivery, project management and integrated business and IT strategy. He is the architect of the Inspired Enterprise Architecture Frameworks and meta model and the Archi/EA Webmodeler tool, now known as EVA Netmodeler.

His research interests include systems development methods, modeling, meta modeling, architecture and tool design. Recent focus areas include collaborative modeling, 3D virtual Spaces and the use of game design principles in professional tools.

Graham is currently principal of Inspired and CTO of Promis Solutions AG. He has authored several books and consults at board level via Inspired, Promis and Gartner.

inspired!
IT ● Consulting ● Training ● Research ● Tools