



# **A UML-based Architectural Framework for Web-based Information Systems**

Egil P.Andersen  
Norwegian Computing Center  
P.O.Box 114, Blindern, 0314 Oslo, Norway  
Tel: +47 22 85 25 94, Fax: +47 22 69 76 60  
Egil.Paulin.Andersen@nr.no



# Distributed Information Systems - SynEx and SINAI

**Distributed Information Systems** play a key role in the business of many companies and organisations. They may need an ability to manage and maintain business information from several heterogeneous information sources, where authorised users can access and share this information with few constraints as to where they are located geographically, and sometimes also the equipment they are required to possess.

Web-based information systems - provide access to information from "anywhere" to "anyone".

## SynEx - Synergy on the Extranet

To support continuity of care between organisations, and make relevant patient information readily available, there is a growing trend towards *shared care*.

Synapses and SynEx are EU projects which aims to provide healthcare professionals with seamless access to patient records and related information, where different record parts may reside in different EPR information systems, nationally or internationally.

The ultimate goal for electronic patient records is to provide *complete* access to *any* part of *any* record to *any authorised* healthcare professional in a *secure* way.

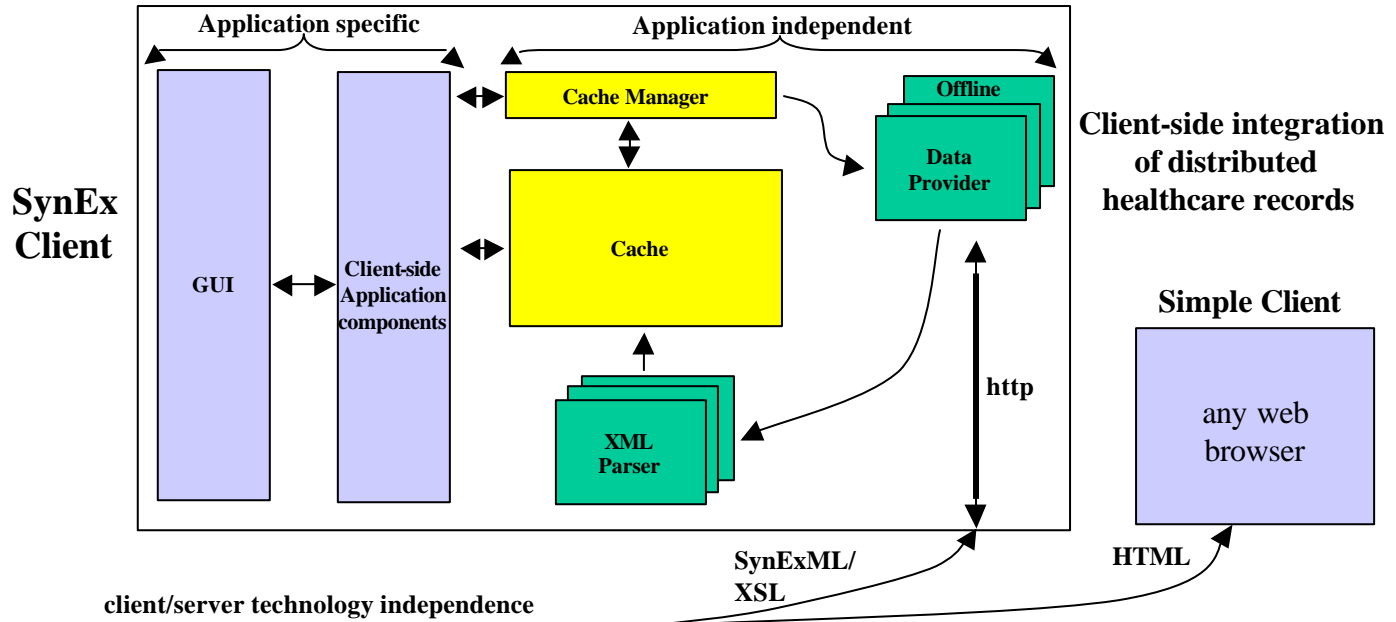
## SINAI

Healthcare information systems are *open and generic* - the kind of information they must be able to manage is not fully known at design-time. They evolve over time and must be designed for this.

SINAI is a UML-based architectural framework for web-based information systems. It supports the development of web-based *generic* and *extendable* distributed information systems, but without being generic itself; i.e., it allows for *domain-specific* specialisation, customisation and optimisation. It also supports seamless integration of information from heterogeneous data sources and legacy systems.



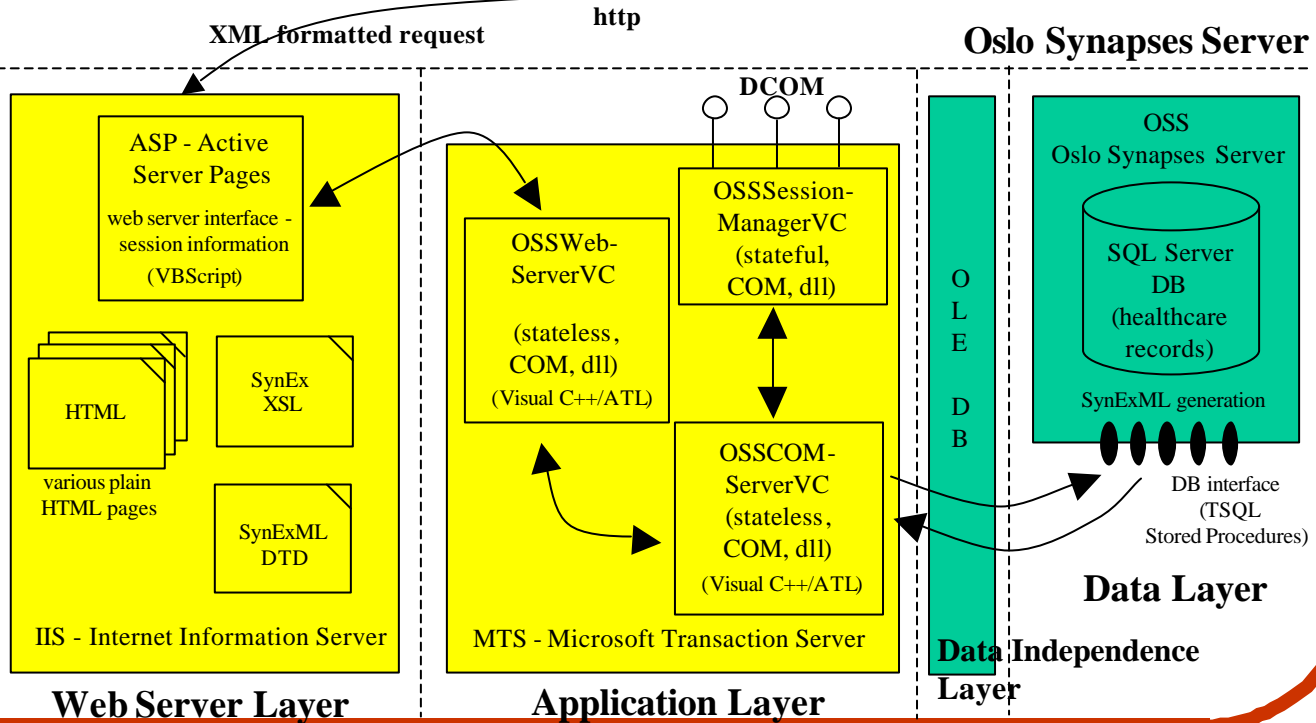
# SynEx - Technical Architecture



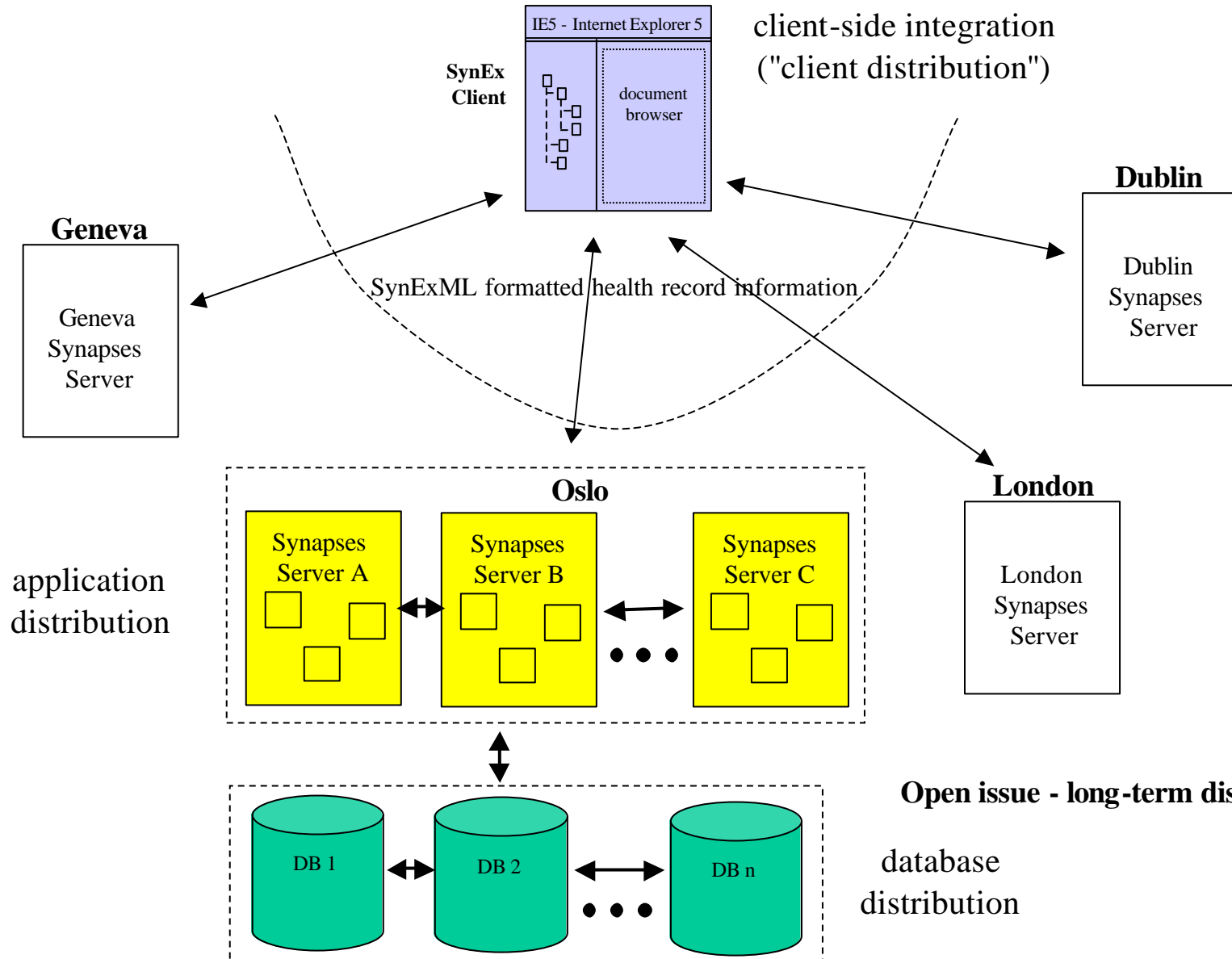
SynEx Client:  
ActiveX components  
in Internet Explorer

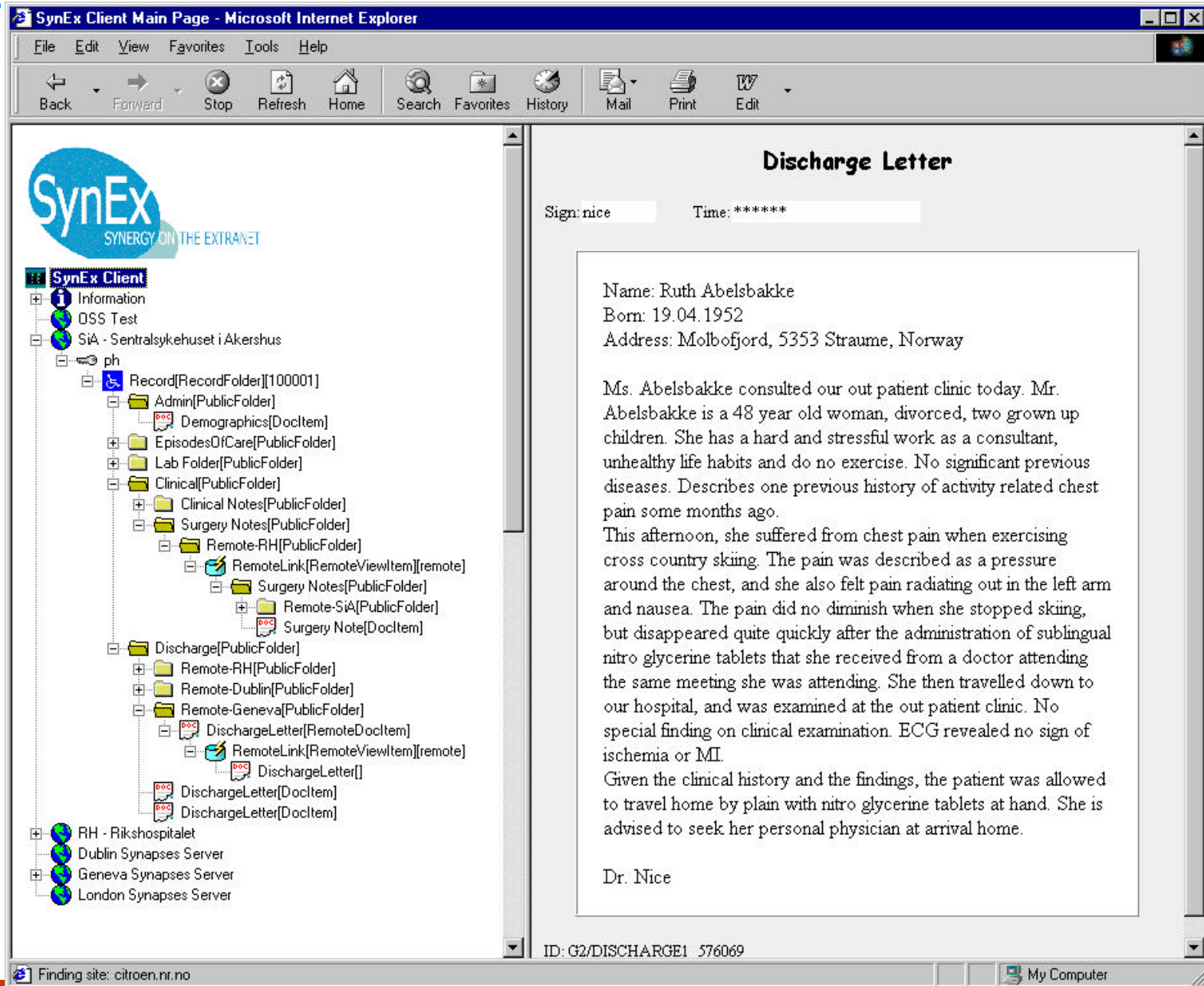
Request/Response:  
XML over http

Oslo Synapses Server:  
IIS/ASP on Win/NT  
and COM components  
under MTS



# Seamless Integration of Distributed Electronic Patient Records





**SynEx**  
SYNERGY ON THE EXTRANET

**SynEx Client**

- Information
- OSS Test
- SiA - Sentralsykehuset i Akershus
  - ph
    - Record[RecordFolder][100001]
      - Admin[PublicFolder]
        - Demographics[DocItem]
      - EpisodesOfCare[PublicFolder]
      - Lab Folder[PublicFolder]
      - Clinical[PublicFolder]
        - Clinical Notes[PublicFolder]
        - Surgery Notes[PublicFolder]
          - Remote-RH[PublicFolder]
            - RemoteLink[RemoteViewItem][remote]
              - Surgery Notes[PublicFolder]
                - Remote-SiA[PublicFolder]
                  - Surgery Note[DocItem]
          - Discharge[PublicFolder]
            - Remote-RH[PublicFolder]
            - Remote-Dublin[PublicFolder]
            - Remote-Geneva[PublicFolder]
              - DischargeLetter[RemoteDocItem]
                - RemoteLink[RemoteViewItem][remote]
                  - DischargeLetter[DocItem]
            - DischargeLetter[DocItem]
            - DischargeLetter[DocItem]
  - RH - Rikshospitalet
  - Dublin Synapses Server
  - Geneva Synapses Server
  - London Synapses Server

**Discharge Letter**

Sign: nice      Time: \*\*\*\*\*

Name: Ruth Abelsbakke  
 Born: 19.04.1952  
 Address: Molbofjord, 5353 Straume, Norway

Ms. Abelsbakke consulted our out patient clinic today. Mr. Abelsbakke is a 48 year old woman, divorced, two grown up children. She has a hard and stressful work as a consultant, unhealthy life habits and do no exercise. No significant previous diseases. Describes one previous history of activity related chest pain some months ago.

This afternoon, she suffered from chest pain when exercising cross country skiing. The pain was described as a pressure around the chest, and she also felt pain radiating out in the left arm and nausea. The pain did no diminish when she stopped skiing, but disappeared quite quickly after the administration of sublingual nitro glycerine tablets that she received from a doctor attending the same meeting she was attending. She then travelled down to our hospital, and was examined at the out patient clinic. No special finding on clinical examination. ECG revealed no sign of ischemia or MI.

Given the clinical history and the findings, the patient was allowed to travel home by plain with nitro glycerine tablets at hand. She is advised to seek her personal physician at arrival home.

Dr. Nice

ID: G2/DISCHARGE1 576069

Finding site: citroen.nr.no      My Computer



## SynExML and XML over http (e.g. SOAP)

### SynExML

XML has the power to become the independent data exchange format of the future.

An XML DTD, *SynExML (SynEx Markup Language)*, was defined for inter-site exchange of EPR (Electronic Patient Record) information; i.e., the basis for semantic interoperability between SynEx components w.r.t. EPR information.

### XML over http (e.g. SOAP)

Communication between a client and a server can be formatted as XML over http both ways (e.g. according to the *SOAP (Simple Object Access Protocol)* specification).

There are several advantages by this:

- http is a simple protocol with good coverage and few demands on the client XML, as strings, are well-suited for transmission via http
- Most firewalls are readily configured for common security options dealing with well known internet protocols and ports. This as opposed to e.g. DCOM or CORBA protocols (IIOP). In practice, the ability for remote machines to interact via DCOM and IIOP is more limited. DCOM and IIOP can be well-suited for computers within e.g. a limited area, but not between "any" remote client and server on the internet.
- XML over http makes the underlying client- and server-side technology transparent to each other. E.g. the Oslo EPR server is Microsoft based while the Dublin server uses Apache with CGI scripts.
- XML is likely to be well-suited for mobile clients; e.g. WML is XML.
- There are numerous software tools and systems with good support for XML.

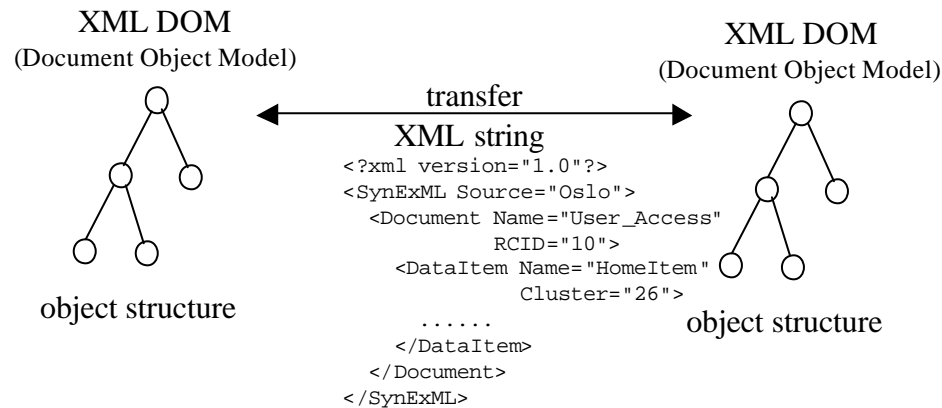
http GET command (QueryString)

```
http://citroen.nr.no/synexdemo/oss.asp?<OSSrequest>  
    <Function Name="RecordInfo">  
        <Arg Name="User">onordmann</Arg>  
        <Arg Name="RecordID">{C7910C91....0000}</Arg>  
        <Arg Name="Retrieval">shape</Arg>  
        <Arg Name="ResponseType">html</Arg>  
    </Function>  
</OSSrequest>
```

## What is XML?

The essential characteristic of XML -

XML can be "morphed" from a string of text into a structure of objects, and vice versa.



- Well-suited format for technology independent information transfer - a string  
Particularly between "any" web client and a server
- Most platforms can support XML as strings
- When creating or receiving an XML string it can be accessed and operated upon as any other kind of object structure, with an interface of functions, and also supporting events.



### What is XML not?

XML is *not* a modelling language and *should not be used as such*.

For example - it was a very time-consuming process in SynEx for every partner to agree on a common XML format. Many changes were made along the way, and more are to be expected, and for each change XML parsing code had to be changed.

SINAI - proper modelling methods should be used for information modelling

### XML schema "explosion" - a general problem

If every small group or business sets out to define its own XML schema, this will lead to an "explosion" in the number of non-interoperable schemas and standardisation will be jeopardised.

Making changes in XML parsing code can be expensive; i.e., it is too much to expect that XML schema specific code will be fully encapsulated so that changes due to future standardisation can be confined and implemented with little cost.

BizTalk, OASIS and XML.ORG are all initiatives to support XML *schema repositories* where organisations can publish their XML schemas for the purpose of sharing XML formats and develop industry standards.

SINAI - XML purely a technical means for communication - a single XML format



## Standardised use of XML based on UML

To assure meaningful client-server communication they need a *common understanding* regarding which requests can be made by the client, and what kind of information will be returned from the server.

This can be based on meta-information about UML models; i.e., XML transferred between clients and servers can use a single XML schema, with two major sub-schemes:

- information on **object instances of a UML model** (actual object relationships, attribute values, etc) or **UML model information** (meta-information - classes, structural and behavioural properties, etc)
- how to invoke **object methods/functions** according to a UML model

A client receives not only information on a particular structure of objects, but also their business rules, their methods, etc.

Application developers will be presented with object models and object interfaces that adhere to SINAI defined UML conventions - both when manifested as XML or IDL component interfaces.

A unified way of working with objects made according to models that may well be very domain specific.

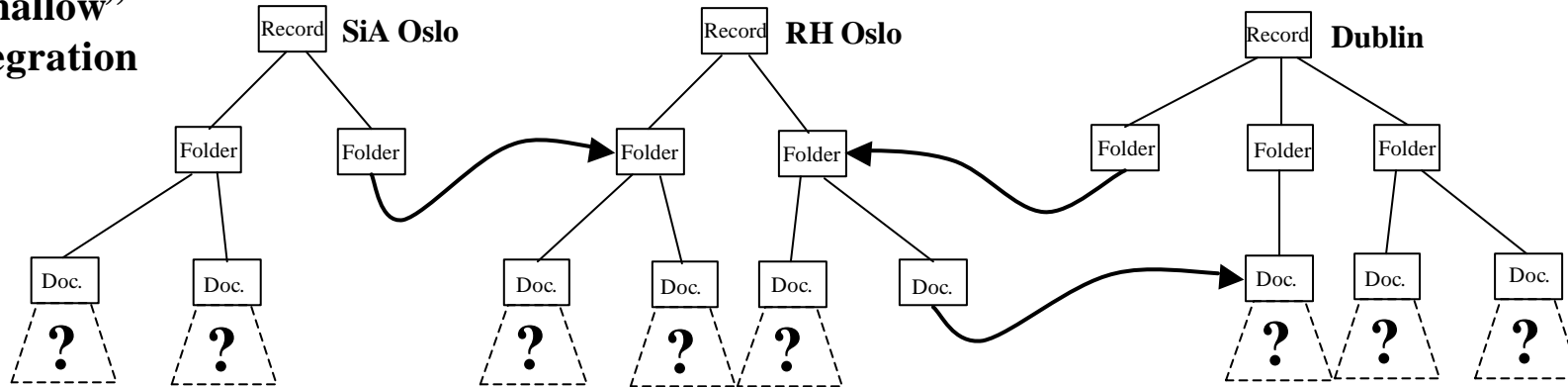
Benefits:

- Foundation for client-server common understanding remains the same regardless of model changes
- Client-server communication infrastructure remains unchanged
- Information and application integration - different systems adhere to the same conventions regarding their structural and behavioural properties and interfaces (XML and IDL component interfaces)

Projects like SynEx can focus more on conceptual issues and less on technology and platform

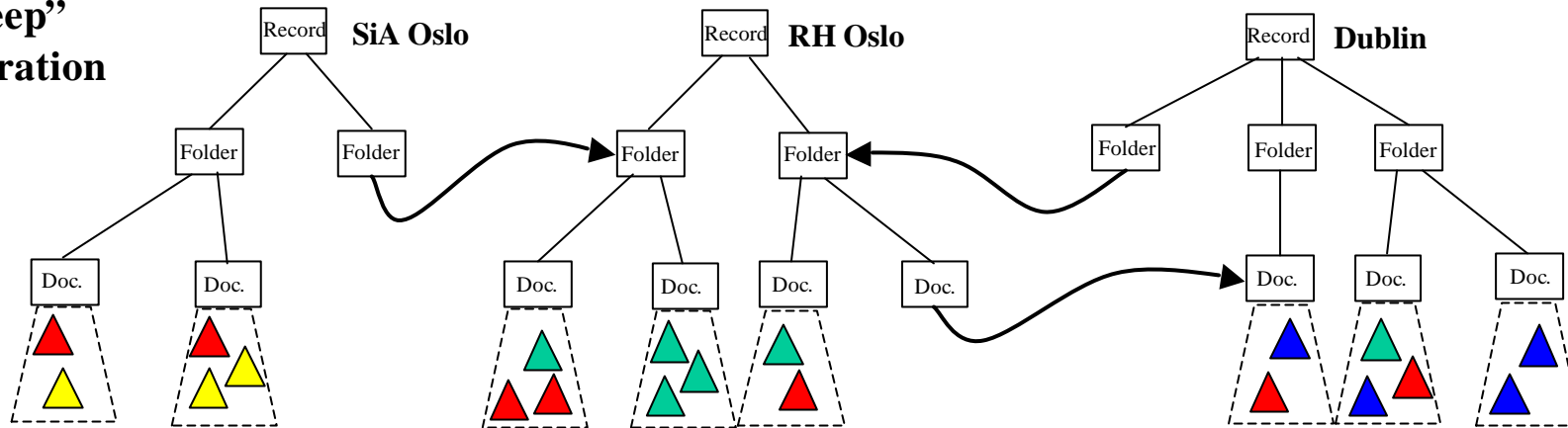
Healthcare information systems are *open and generic* - the kind of information they must be able to manage is not fully known at design-time. They evolve over time and must be designed for this.

## ”Shallow” integration



Sharing and seamless integration of distributed electronic patient record - but for their **shapes** only!

## ”Deep” integration



EPR documents can be considered dynamically extendable sets of site-specific and standardised models in combination. We need to extend and integrate domain specific models in a manner that cannot be foreseen at design time.

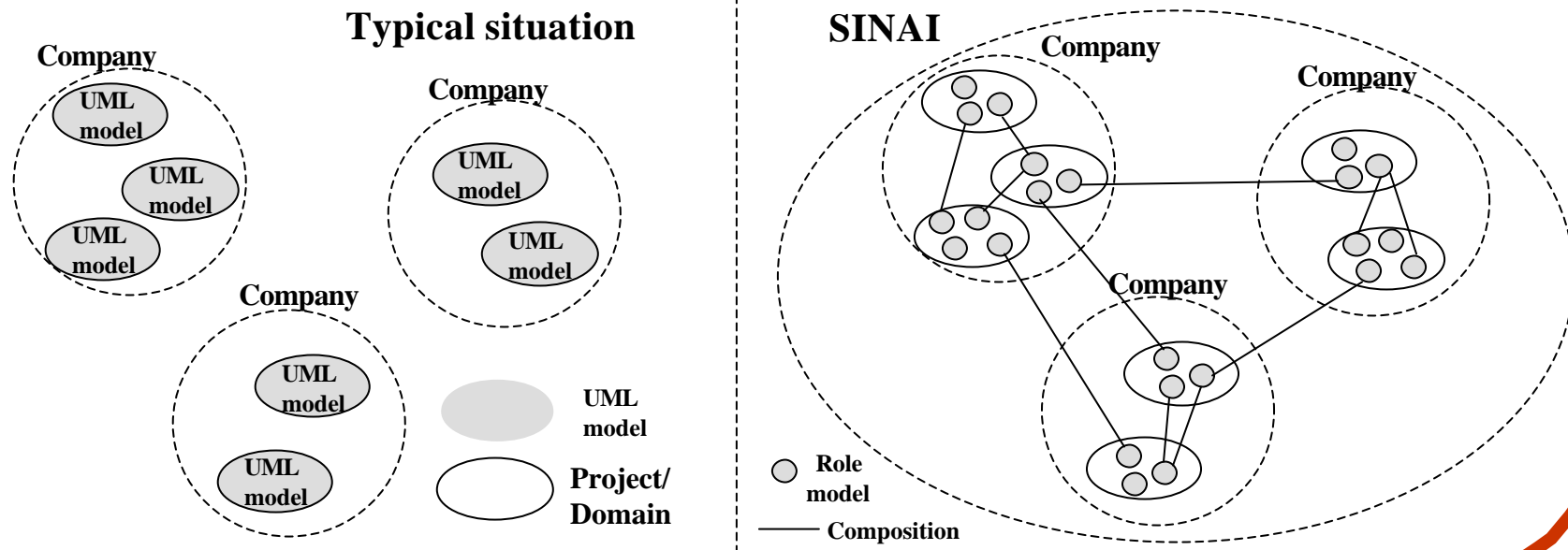
## SINAI:

Not large, monolithic (unmanageable...) UML models, e.g. for different projects or domains.

Numerous “micro” models (role models - see below) for different areas of concern.

Models can be distributed, integrated and reused in a seamless manner regardless of origin.

Hypothesis: for electronic patient records, a separate model can be made of its "record shape and hyperlink" structure (e.g. nested tree-structures with local or remote hyperlinks); separate models are made for various kinds of record content (parts of medical documents); other models for authorisation and access control, presentations (various devices), and so on.



# The SINAI Unified Role Model for Distributed Information Models

A SINAI based system will have many of the characteristics of a generic system, but without being generic itself - it will not predominantly consist of generic software components.

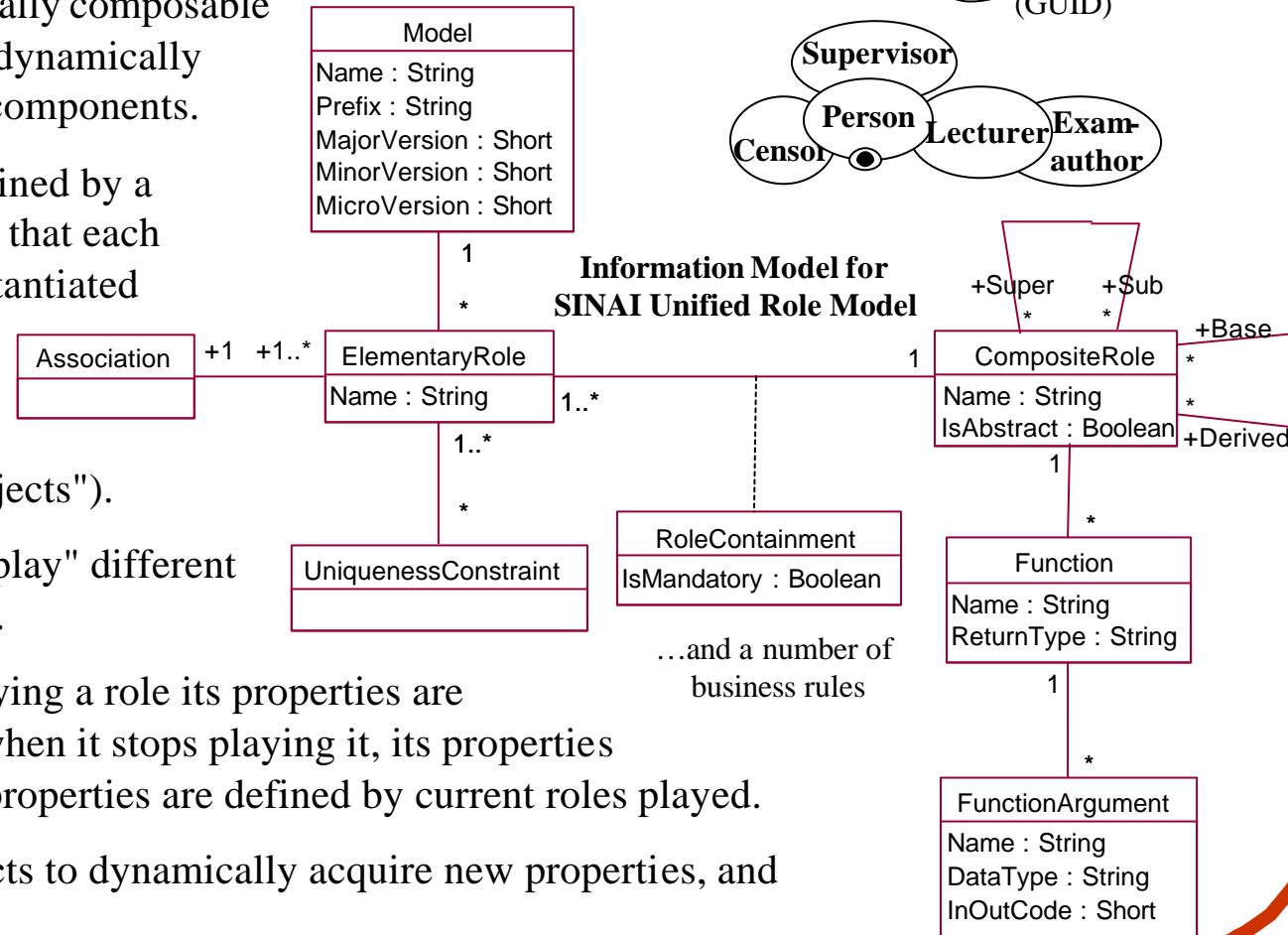
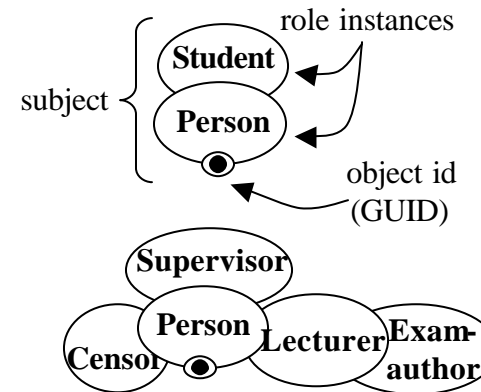
It will consist of dynamically composable units of information, and dynamically configurable application components.

A SINAI data layer is defined by a set of information models that each offer *roles* that can be instantiated into role instances and dynamically composed into persistent, stateful information objects ("subjects").

Information objects can "play" different roles during their lifetime.

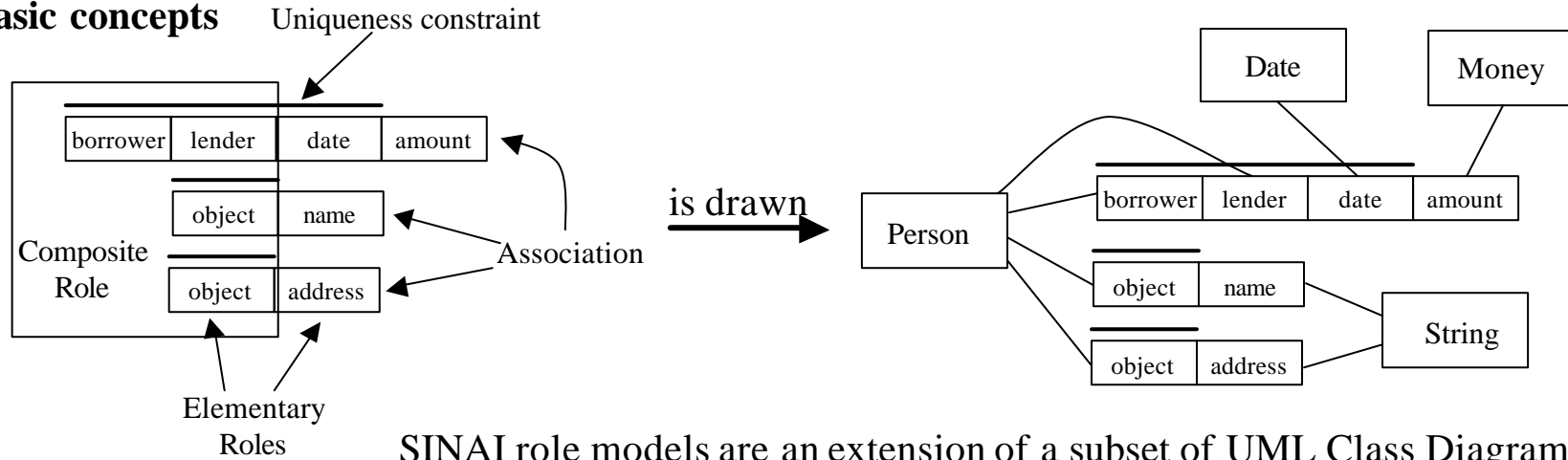
When an object starts playing a role its properties are added to the object, and when it stops playing it, its properties are removed; i.e., object properties are defined by current roles played.

Thus roles allow for objects to dynamically acquire new properties, and later release them.

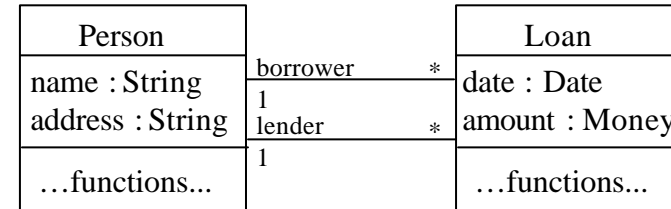


# The SINAI Unified Role Model for Distributed Information Models (cont.)

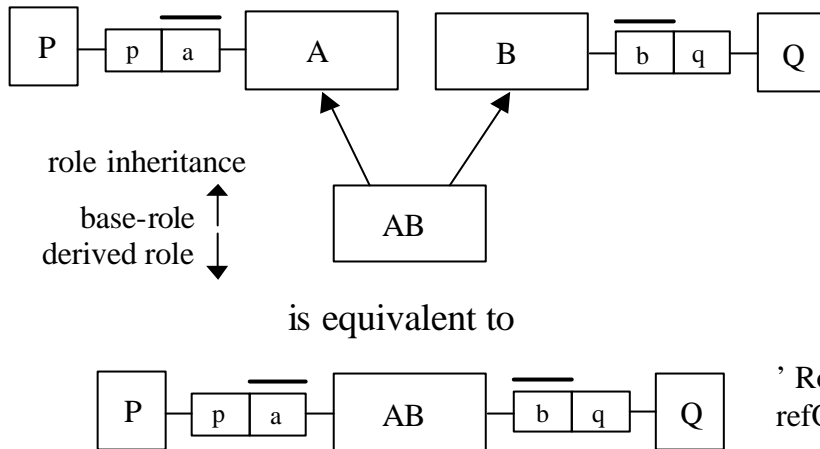
## Basic concepts



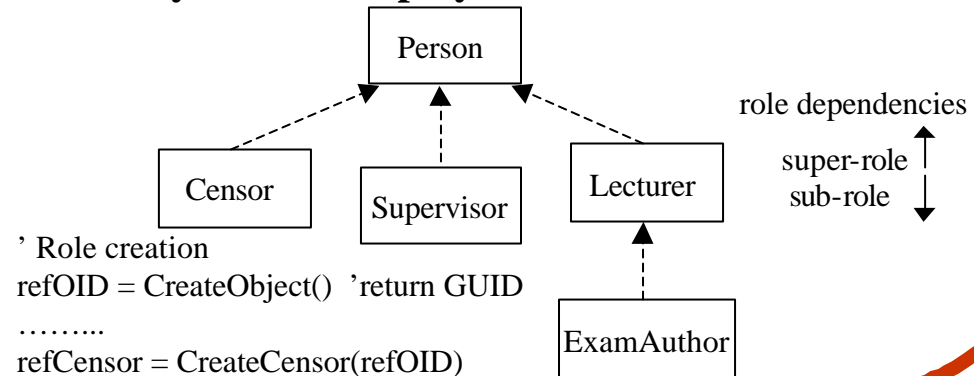
SINAI role models are an extension of a subset of UML Class Diagrams - can use UML notation; e.g.



## Role Inheritance

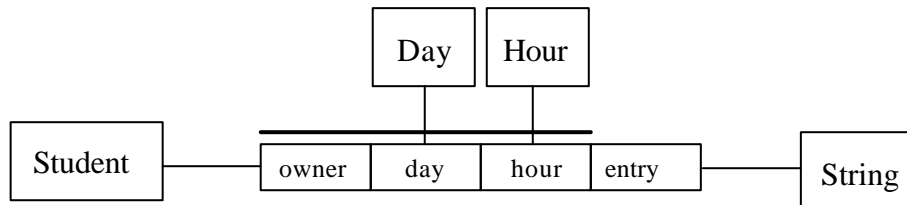


## Dynamic role play and role constraints



# The SINAI Unified Role Model for Distributed Information Models (cont.)

n'ary associations for e.g. multi-dimensional arrays



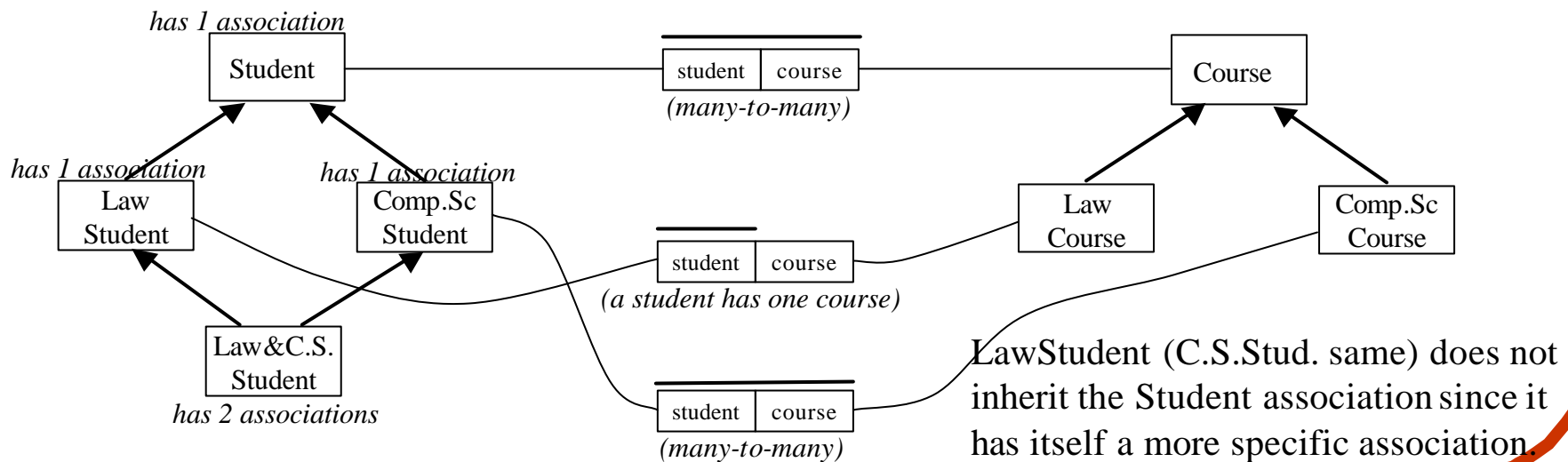
A problem in COM IDL - function returntypes are "occupied"...

```
Visual Basic: refStudent.FindDay("Tuesday").FindHour("17.30").SetEntry = "math"
strEntry = refStudent.FindDay("Tuesday").FindHour("17.30").GetEntry
```

## Inheritance and association subtype relationships (the main difference from NIAM/ORM)

Role inheritance is different from role dependencies.

When inheriting - for subtype related associations - only the most specific association is inherited.



**SINAI Information Models** define **stateful objects** residing on persistent storage.

Their their interfaces contain functions for creating, removing, updating and retrieving role instances.

The data layer implementation of these functions handle information specific, but application independent, constraints.

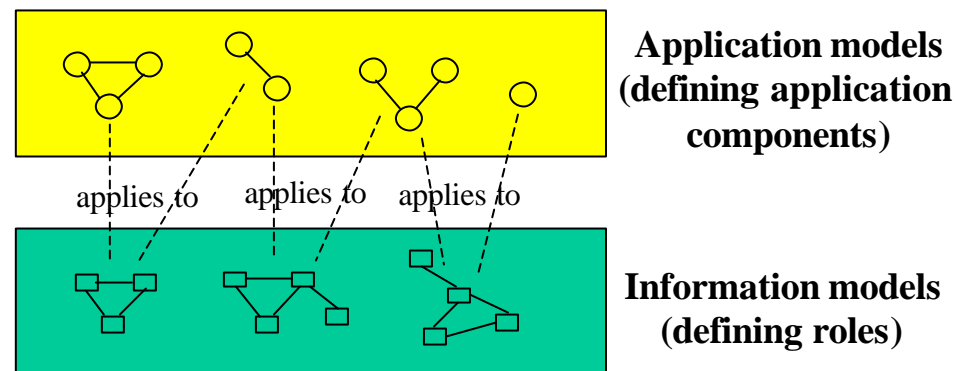
**SINAI Application Models** are service-oriented (e.g. originating from Use Cases) - concerns a particular set of related services that can be applied to particular information models.

Application layer objects are mostly **"stateless" objects** as required for scalability and good performance in large-scale information systems.

Component interfaces defined by both information and application models will be model specific - no generic interfaces.

Hence manual customisation and modification is possible at the level of individual, model specific functions - e.g. complex business rules usually require non-trivial manual intervention at a detailed, model specific level

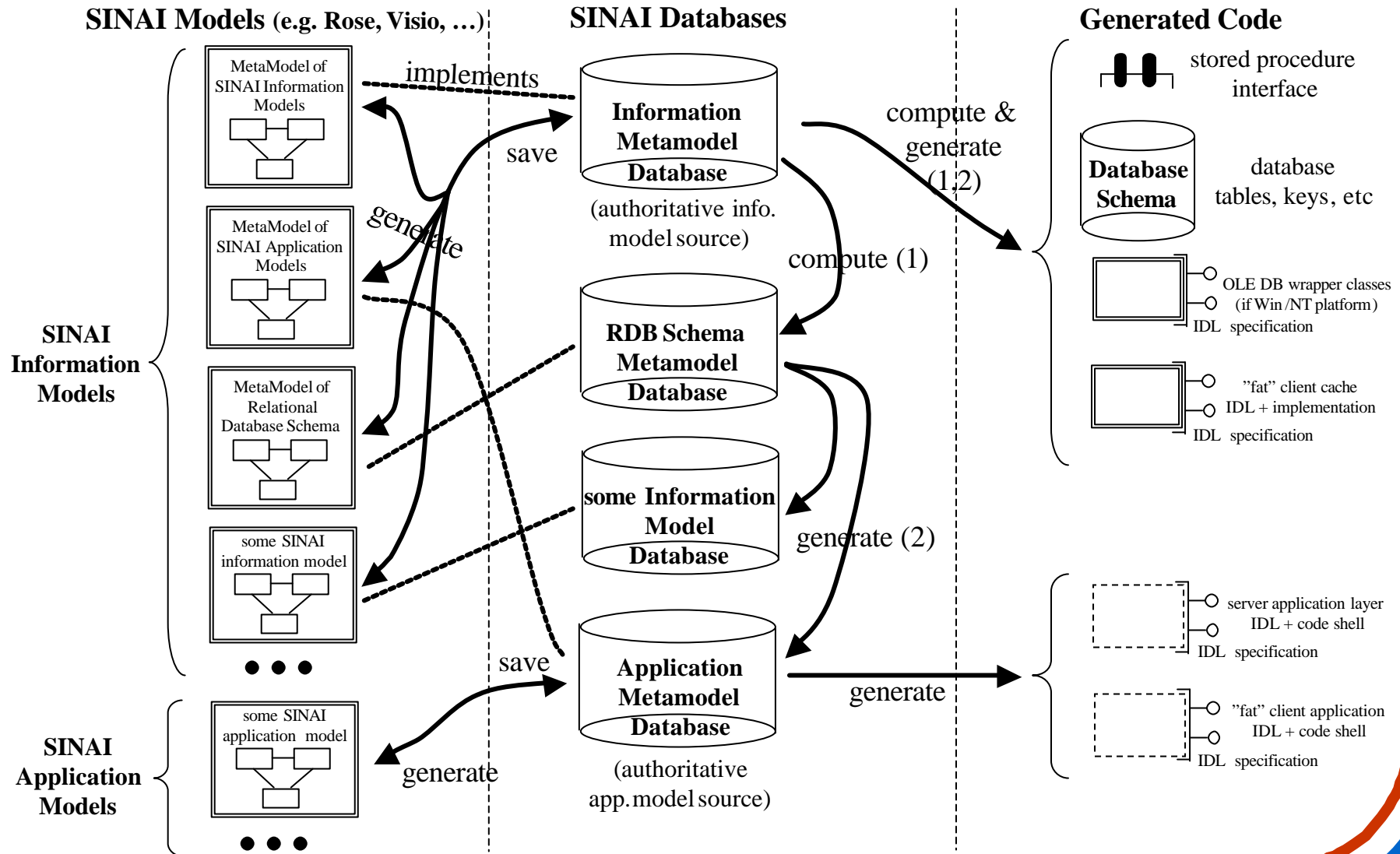
*PS: Conceptual work remains on SINAI application models*



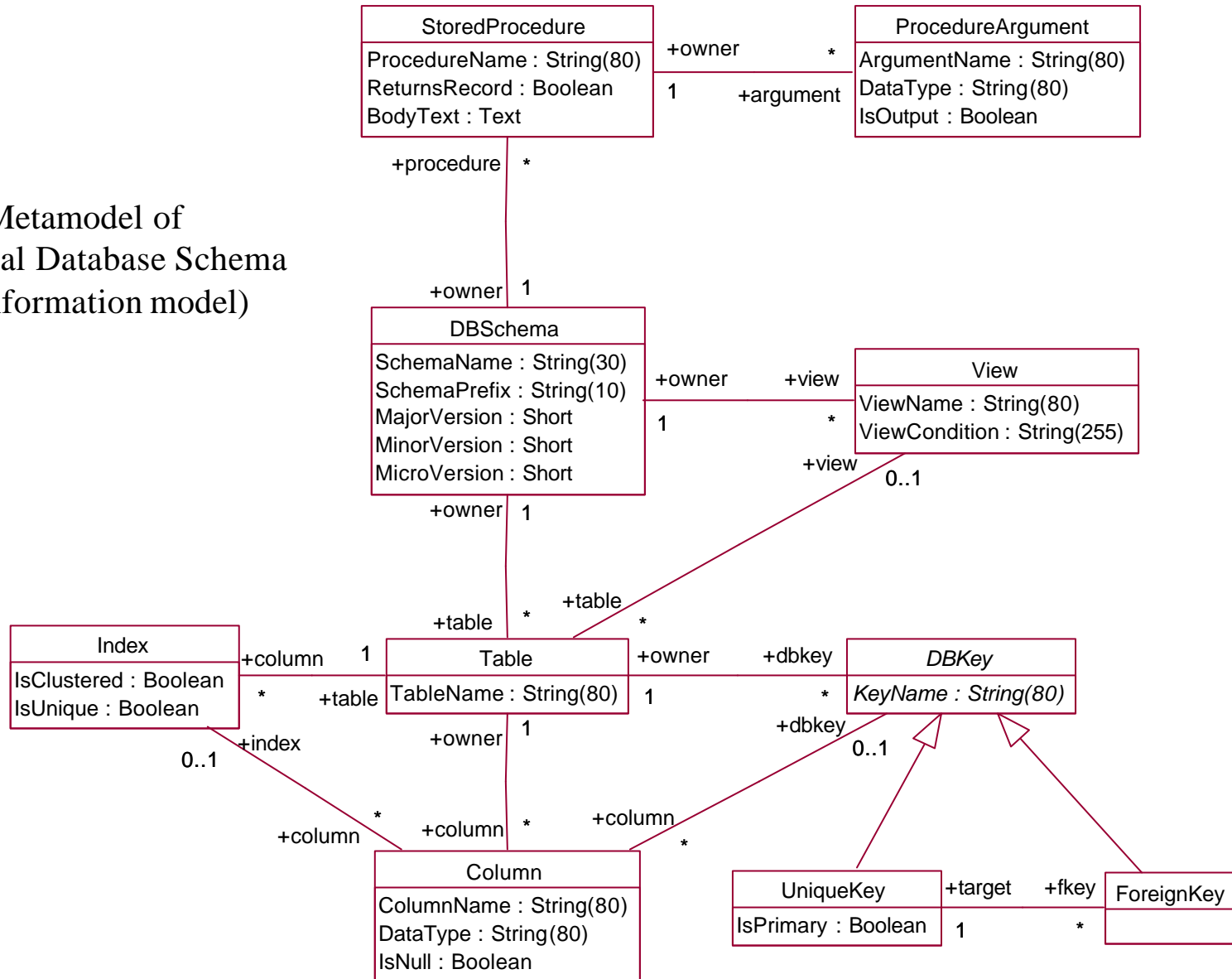


# Standardised SINAI Architectures

SINAI models can be used to *generate* parts of the model specific server code - and ("fat") client side components => saves some coding, and more standardised (product line) architectures

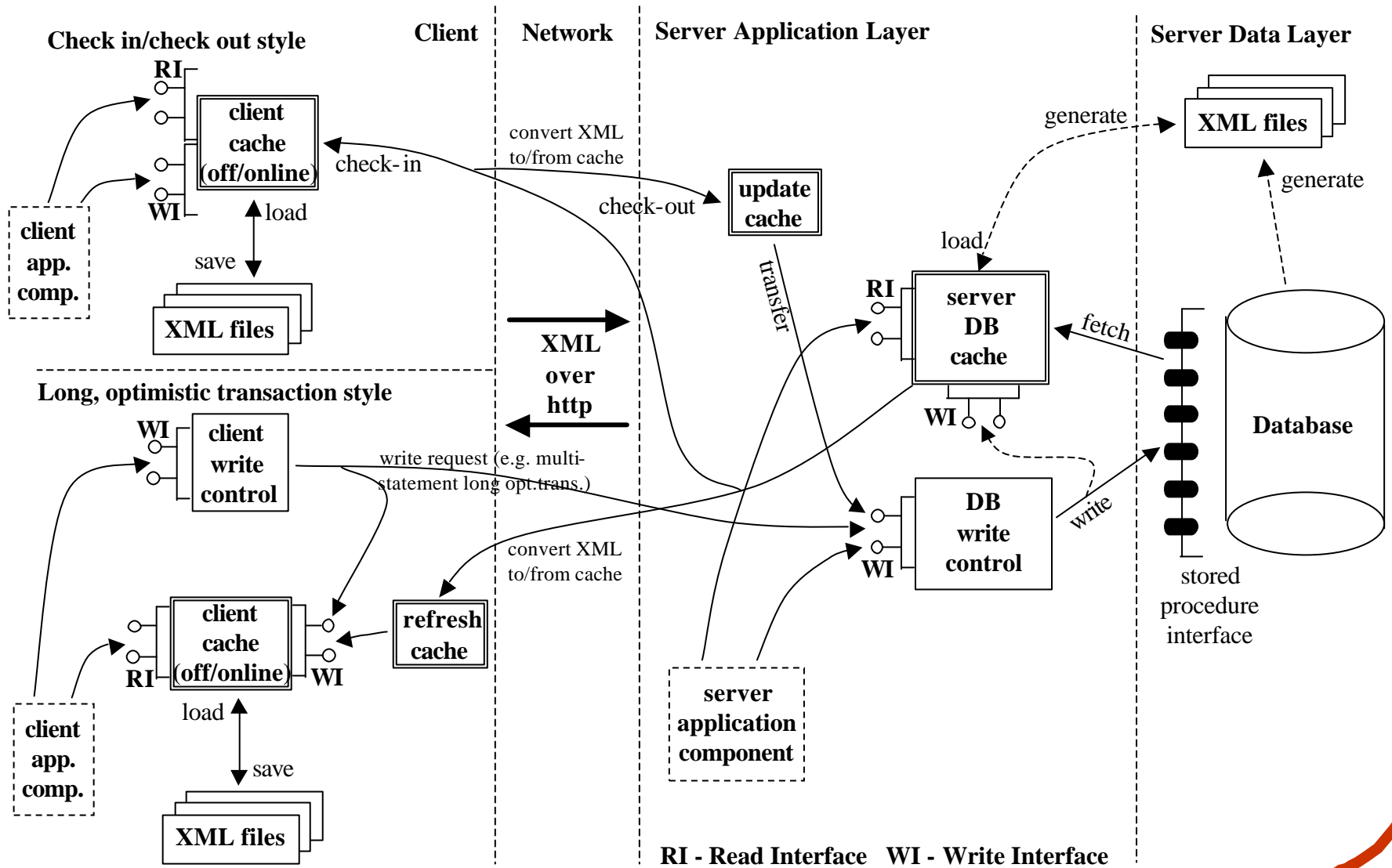


Metamodel of  
Relational Database Schema  
(an information model)



# Standardised SINAI Architectures (cont.)

Some typical "fat" client-server scenarios. "Thinner" clients imply more processing on the server. Notice the distinction between "read" vs "write" interfaces (operations) for transaction handling.



## SynEx

With state-of-the-art internet technology it is relatively simple, technically, to achieve sharing and seamless integration of distributed electronic patient record **shapes** (record structure).

Patient records are quite complex information entities. Their documents can be considered dynamically extendable sets of site-specific and standardised models in combination.

We need to extend and integrate domain specific models in a manner that cannot be foreseen at design time. A pure, generic solution has proven insufficient/too complex.

## SINAI

The SINAI project aims to demonstrate that

- UML can be used to achieve a basic common understanding between parties involved (clients, servers, components, applications) that makes it easier to achieve integration and interoperability of information and applications.
- UML can be used to standardise the use of XML and avoid XML schema "explosion".
- Code generation from UML models can be used to achieve more standardised architectures (at least for a particular product line).
- Model and component integration via the combined use of role modelling techniques and component technology can be used to support the development of generic and extendable information systems.