Designing and Implementing Clinical Information Systems

Centre for Health Informatics

Prof John Chelsom

Runs: 3 hours
Tutor: Prof John Chelsom
Mode of attendance: Classroom

Learning Objectives

- Cover the principles of good design and implementation of Clinical Information Systems (CIS)
- Specific learning objectives are to:

1 Understand the phases of implementation

2 Define the role of information architecture

3 Gain an overview of methods for logical system design

4 Learn about good practice in clinical user interface design

5 Understand the options for integration of clinical systems

Designing and Implementing CIS

- Design and implementation of CIS
- Requirements for health records systems
- Logical design of a health record
- Designing a clinical user interface
- Integration of clinical systems
- References and Further Reading

Information Sources

- Sources are listed in the references at the end of these slides

Some definitions and descriptions have been taken from quoted resources.

Retrieved October 2010.

Where consensus on definitions or descriptions Is required, these have been taken from Wikipedia.

Retrieved October 2010.

Understanding the Phases of Implementation

Characteristics of a Health Informatics Project

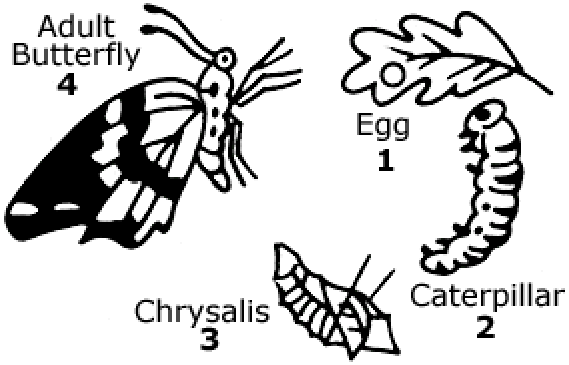
- Health Informatics projects are in many ways just the same as any other systems implementation project

- But have some specific characteristics which require careful management
  - Clinical safety
  - Patient confidentiality
  - Data Migration
  - Continuous live environment of healthcare
  - A unique set of stakeholders

Products and Solutions

- Different approach to software development for products and solutions
  - Software products are created to serve a market of many different customers (users)
  - The development approach features
    - An overall product roadmap
    - Market and user requirements
    - Regular release cycle
    - Feature set per release, with carry over
  - Software solutions are built for a specific customer, and may include customisation or configuration of a number of products
    - Scope determined by business requirements
    - Business and user requirements
    - Release cycle determined by scope/quality/resource
    - Minimum (fixed?) feature set, no carry over

Application (Solution) Lifecycle

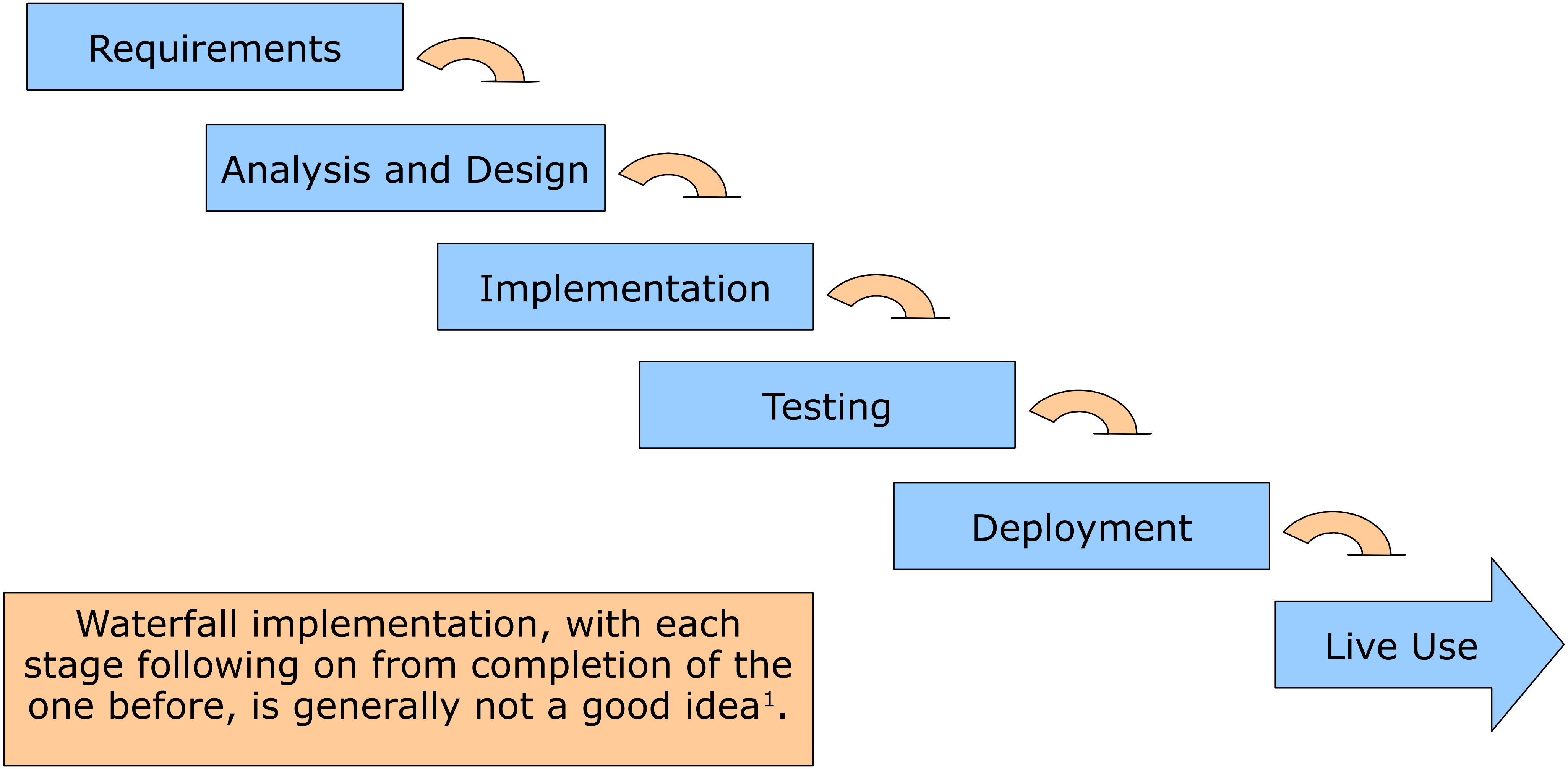


1. Prototype
2. Pilot
3. System
4. Legacy System

"A legacy application is an application that works."

Implementation Lifecycle

- The 'traditional' Waterfall of phases



Waterfall implementation, with each stage following on from completion of the one before, is generally not a good idea[1].

Unified Software Development Process

> The Unified Software Development Process or Unified Process is a popular iterative and incremental software development process framework.
>
> http://en.wikipedia.org/wiki/Unified_process

- The Unified Process is the generic name for the Software Development Process pioneered by Ivar Jacobson, Grady Booch and James Rumbaugh.
- It combines various approaches developed by the three, who came together at Rational Software Inc, a firm later acquired by IBM
- The Rational Unified Process (RUP) is now sold by IBM as a methodology and associated toolset
- It is not necessary to use the RUP licensing or toolset to use the Unified Process – there are plenty of other (generic) information sources and toolsets, some of which are available as free open source software
- OpenUP is one open source UP toolset, built on the Eclipse project

Principles of the Unified Process

The Unified Process is based on several key principles:

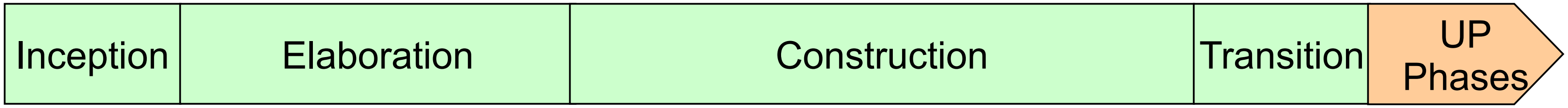
- develop software iteratively
- manage requirements
- use component-based architectures
- visually model software (e.g. with UML)
- continuously verify software quality
- control changes to software

Unified Process – Phases

The Unified Process is divided into four phases

- Inception
  - Specifies the vision of the end-product and its business case. Defines the scope of the project.
- Elaboration
  - Planning of activities and resources, eliciting requirements, specifying the system features and designing the architecture, eliminating the highest identified risks
- Construction
  - Entire system is developed and tested, with user documentation
- Transition
  - Move the completed product to the user community, training them in its use and signing off acceptance

- Within each phase there may be several iterations

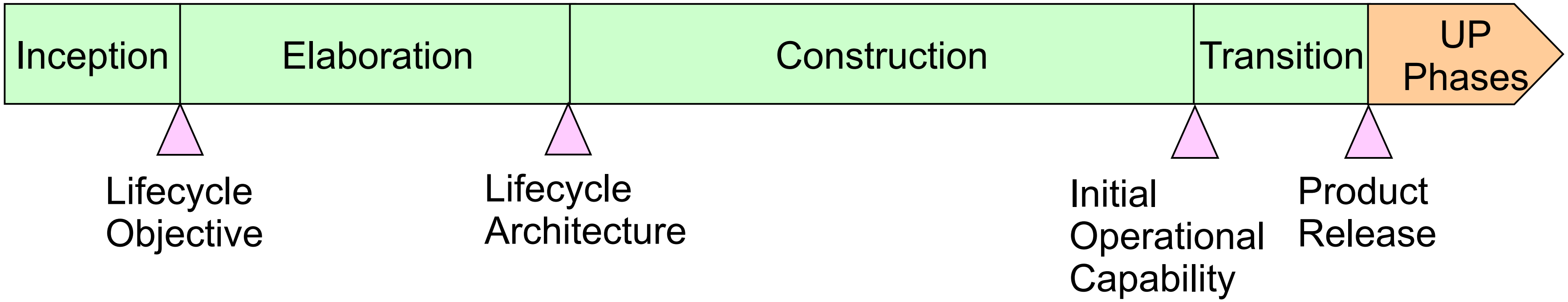


Inception | Elaboration | Construction | Transition | UP Phases

Unified Process – Milestones

Major milestones mark the boundaries between each phase

- Lifecycle Objective
  - Defined scope, estimated costs, understood requirements and risks
- Lifecycle Architecture
  - Detailed requirements, stable architecture, major risks resolved
- Initial Operational Capability
  - Functional system, full scope, release candidate
- Product Release
  - System accepted, objectives satisfied, ready for live operation



Disciplines

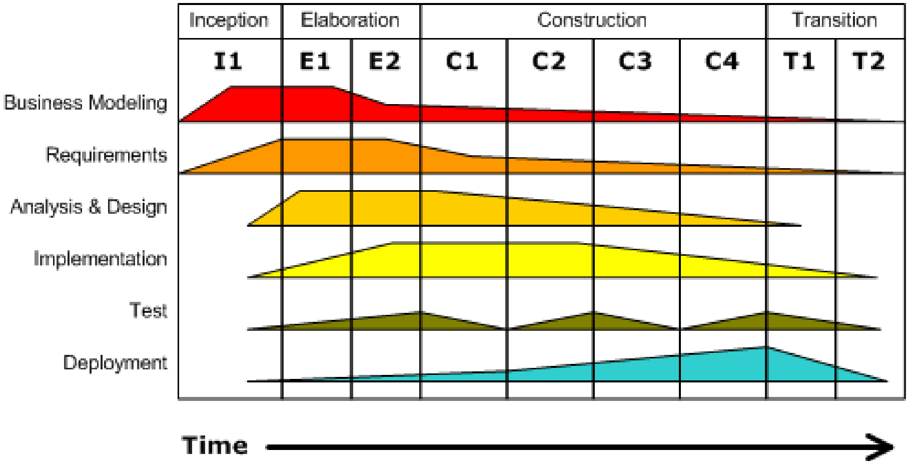
Disciplines are used as a way to group activities, according to the primary skill sets required to complete them.

The core technical disciplines are

- Business Modelling
- Requirements
- Analysis and Design
- Implementation
- Test
- Deployment

There are three additional supporting disciplines

- Project Management
- Configuration / Change Management
- Environment



Each discipline has some involvement throughout the whole development lifecycle.

Knowledge Transfer which should be responsible for documentation and training deliverables falls under the Deployment discipline.

Unified Process – Inception Phase

Objectives

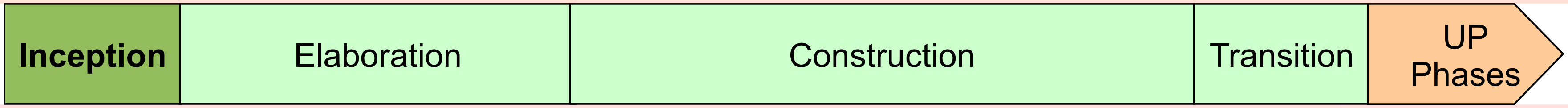
- Defined scope/acceptance criteria
- Outline major Use Cases
- Assess candidate architectures
- Estimate of costs

Activities

- Formulate scope
- Specify Requirements (high level)
- Use Case definition (summary)
- Risk assessment
- Appraise candidate architectures

Deliverables

- Vision document, with scope and key requirements, features, constraints
- Use Case model, with major actors and use cases defined
- Business Case
- Risk register
- High level project plan
- First Work Breakdown Structure and estimates



Unified Process – Elaboration Phase

Objectives

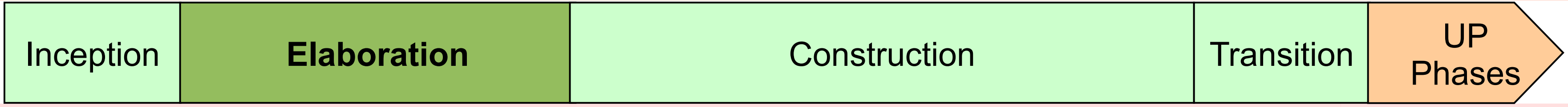
- Validate the system architecture
- Detailed plan for Construction
- Demonstrate the Vision to stakeholders
- Verify estimates
- Mitigate major risks

Activities

- Build the Vision into a fully understood set of requirements
- Set up the software engineering environment
- Create executable product build
- Planning and estimating

Deliverables

- Complete Use Case model that identifies all cases and actors
- Software Requirements Specification, including non-functional requirements
- Software Architecture Description
- Functional build, release and automated test environment
- Executable build of the product that verifies the architecture
- User interface storyboard, wireframe or click through prototype
- Development plan for Construction phase



Inception

Elaboration

Construction

Transition

UP Phases

Unified Process – Construction Phase

Objectives

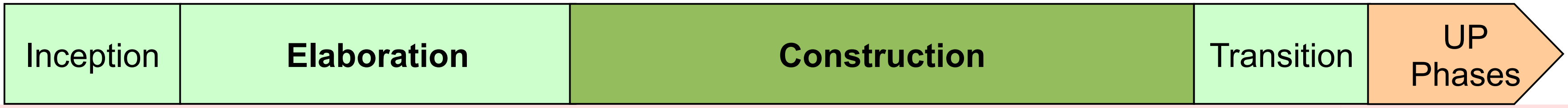
- Create software release candidates within set timeframe
- Ensure quality goals are met
- Manage the resources (cost) to deliver required scope, within the limits set

Activities

- Develop and test software components
- System and integration testing
- Create materials for knowledge transfer
- Planning and estimating

Deliverables

- Software product verified on the target platform(s)
- Documentation and training material
- Plan for deployment and rollout (Transition)



Inception | **Elaboration** | **Construction** | Transition | UP Phases

Unified Process – Transition Phase

Objectives

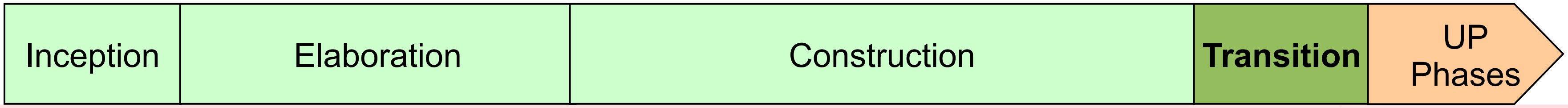
- Go live with final product
- Gain stakeholder acceptance and sign off
- Readiness for support of the live system

Activities

- Deployment to live environment
- Training and other knowledge transfer
- User Acceptance Testing (UAT)
- Planning for support and maintenance

Deliverables

- Functional software product release
- Complete knowledge transfer materials
- Support and maintenance environment
- User Acceptance report
- Handover report of outstanding issues and risks



Inception | Elaboration | Construction | **Transition** | UP Phases

Characteristics of a Health Informatics Project

- Health Informatics projects have some specific characteristics which require careful management
  - Clinical safety
  - Patient confidentiality
  - Data Migration
  - Continuous live environment of healthcare
  - A unique set of stakeholders

Clinical Safety

> The overall aim of clinical safety [..] is to promote and help embed safer working practice methods and patient safety solutions, enabled by IT, applied consistently [..] and to ensure that systems delivered [..] are clinically safe.
>
> NHS Office of the Chief Clinical Officer

- Clinical safety of IT systems in the UK is the responsibility of the Office of the Chief Clinical Officer (OCCO) and the NHS Commissioning Board Special Health Authority (transfered from the National Patient Safety Agency (NPSA))
- As well as normal quality checks, Health Informatics systems should also undergo assurance for clinical safety, with a focus on
  - Adherence to published Clinical Guidelines
  - Risks associated with
    - patient (mis)identification.
    - professional communication
    - availability of information
    - information not acted upon
  - Professional codes of conduct of the target clinical users

NHS Clinical Safety Guidance

- Two major guidance documents were published in association with the NpfIT

Safer Implementation – Key Clinical Safety Activities

A Guide to Implementation

http://www.connectingforhealth.nhs.uk/engagement/clinical/occo/safety/guide

Safer Design - Key Clinical Safety Activities

Clinical Risk Guidance Document

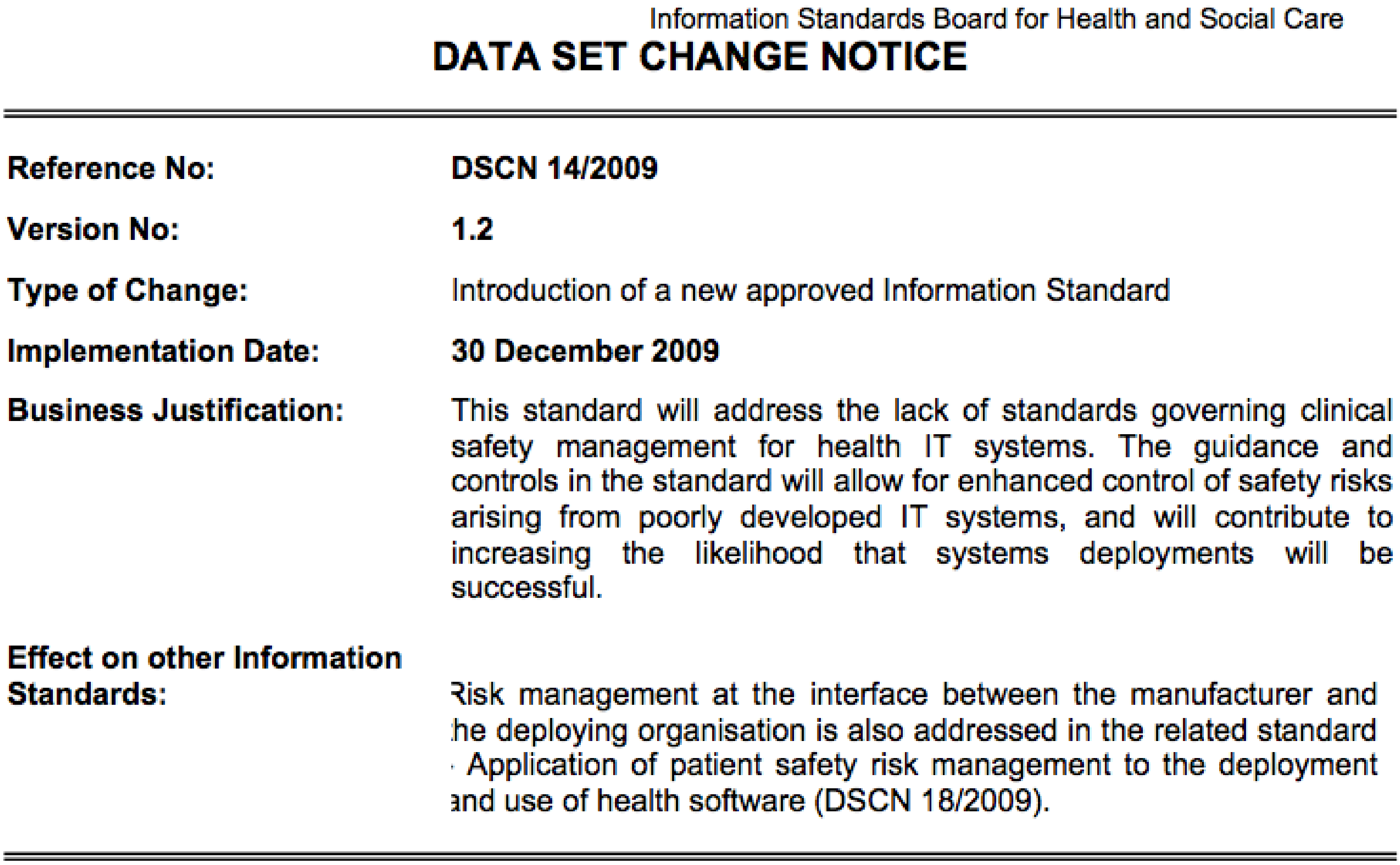
http://www.connectingforhealth.nhs.uk/engagement/clinical/occo/safety

Patient Safety Assessment

- Requirement to create a patient safety assessment document in accordance with:

Health informatics — Guidance on the
management of clinical risk relating to
the deployment and use of health software
Formerly ISO/TR 29322:2008(E)
DSCN18/2009

Patient Safety Assessment



Information Standards Board for Health and Social Care

DATA SET CHANGE NOTICE

Reference No:	**DSCN 14/2009**
Version No:	**1.2**
Type of Change:	Introduction of a new approved Information Standard
Implementation Date:	**30 December 2009**
Business Justification:	This standard will address the lack of standards governing clinical safety management for health IT systems. The guidance and controls in the standard will allow for enhanced control of safety risks arising from poorly developed IT systems, and will contribute to increasing the likelihood that systems deployments will be successful.
Effect on other Information Standards:	Risk management at the interface between the manufacturer and the deploying organisation is also addressed in the related standard - Application of patient safety risk management to the deployment and use of health software (DSCN 18/2009).

Extract from DSCN14/2009

To be compliant with the standard, the manufacturer must establish, document and maintain throughout the lifecycle an ongoing process for identifying clinical hazards associated with the health software product, estimating and evaluating the associated clinical risks, controlling these risks, and monitoring the effectiveness of the controls throughout the lifecycle. This process must include the following elements:

context, requirements and scope identification;
creation of clinical risk management plan;
setting the requirements for and defining the competencies of personnel;
clinical hazard identification;
clinical risk analysis;
clinical risk evaluation;
clinical risk control;
residual clinical risk acceptance;
creation of clinical safety case report(s);
post deployment monitoring;
post-production maintenance of clinical risk management process.

Patient Confidentiality

- Most healthcare environments have specific rules on patient confidentiality

  In the UK guidance on Patient Confidentiality and Access to Health Records is given by the DoH

  Information systems must adhere to the Caldicott Principles

  - Justify the purposes for which information is required
  - Don’t use patient identifiable information unless it is absolutely necessary
  - Use the minimum necessary patient identifiable information
  - Access to information should be on a strict need to know basis
  - Everyone with access should be aware of their responsibilities
  - Understand and comply with the law

  Each NHS organisation must have a Caldicott Guardian

a senior person responsible for protecting the confidentiality of patient and service-user information and enabling appropriate information-sharing

Data Migration

- Many new health informatics systems are designed to replace one or more existing systems
- These systems may be electronic or paper based and there may be a requirement to migrate data from them, to the new system
- Clinical data migration is a notoriously difficult activity to manage
  - Data sets and functionality may vary between old and new systems
    - Requires careful mapping and transformation
  - Data quality in the NHS can be very poor – blame will attached to new systems, not the environment that created the problem
  - Undocumented practices that have be tacitly supported for many years, can cause huge problems in data (and system) migration
- All good reasons to examine carefully the need for data migration
- Most clinical data has a very short useful life

Continuous Live Environment of Healthcare

- Health Informatics systems are generally introduced into an environment that is 'continuously live'
- The process and practice of healthcare cannot be stopped for the introduction of new systems
- Often the healthcare service runs 24x7 and cannot be disrupted
- Resources and opportunities for parallel running of new systems with old can be limited
- The consequences of disruption can be severe
  - For patients, service finances, managers, politicians

This is a good reason to integrate and evolve information systems, not go for the strategy of “rip and replace”.

Technology should be designed to ease migration to new systems.

A Unique Set of Stakeholders

- Health Informatics systems generally have a set of stakeholders that is quite unlike any other environment
- These stakeholders may include
  - Healthcare professionals
    - Doctors, nurses (who provide most care), other care professionals
  - Patients
  - Healthcare administrators
  - Health service managers, civil servants, politicians

"Ask one hundred clinicians what they want, and you'll get one hundred and fifty different answers"

  - This may be true, but is not a reason to ignore stakeholders

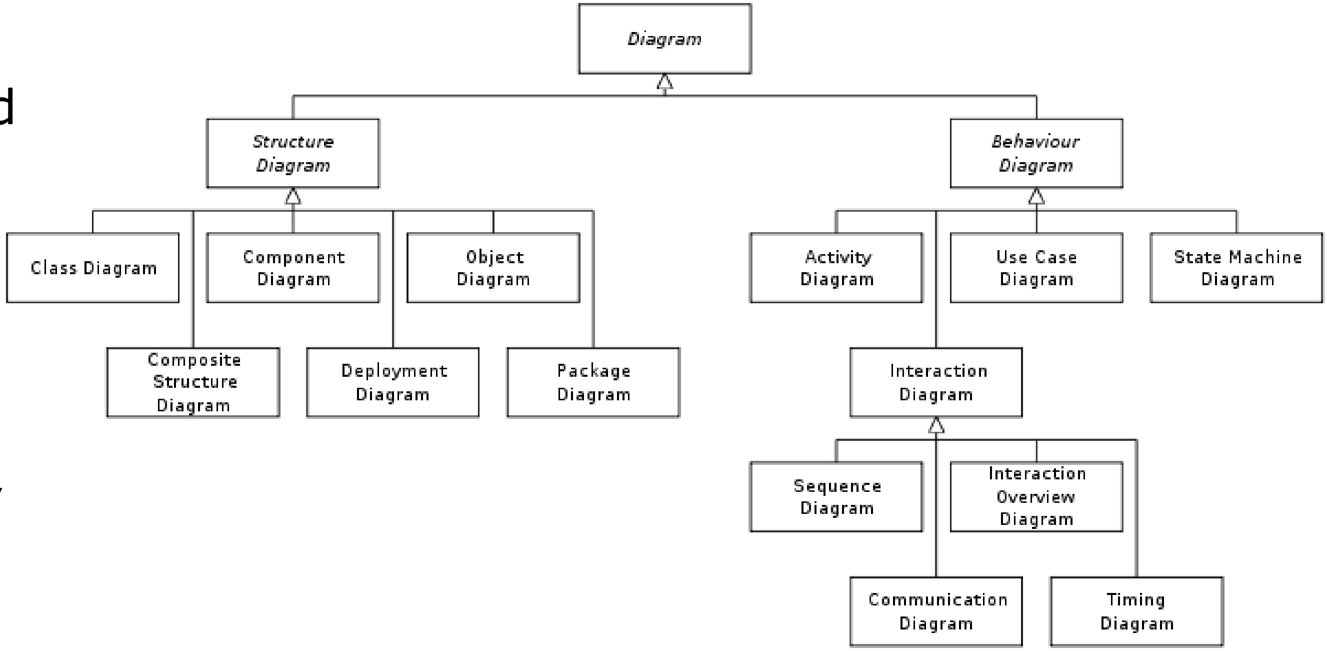
Unified Modelling Language

Visually Model Software

The Unified Modeling Language (UML) is an open method used to specify, visualise, construct and document the artifacts of an object-oriented software intensive system under development.

http://en.wikipedia.org/wiki/Unified_Modeling_Language

- The Unified Process encourages the use of the Unified Modelling Language (UML) to visually model software
- UML 2.0 has 13 types of diagram as shown in this chart, which is itself a UML Class Diagram
- Policies need to be in place to manage how UML is used
- Its best use is as an aid to design and communication during Inception and Elaboration
- The links between models and software code are easy to break in Construction and costly to maintain

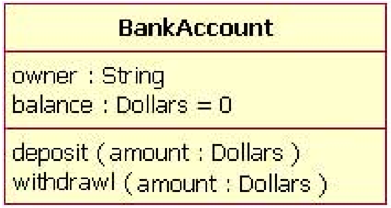


UML Tools

- There are many (often expensive) modelling tools
  - Rational Rose (IBM)
  - Enterprise Architect
  - Visual Paradigm
  - Altova Umodel (they do good XML tools too)

- And some free, open source
  - My favourite is Bouml
  - (except that it isn't open source any more)

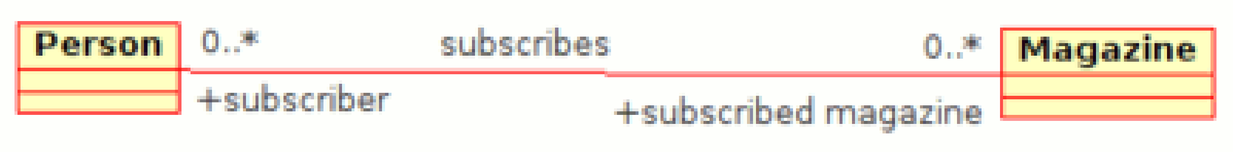
UML Class Diagram



Class Name

Attributes

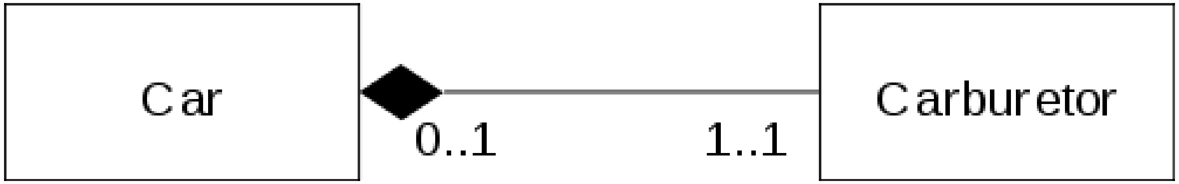
Methods



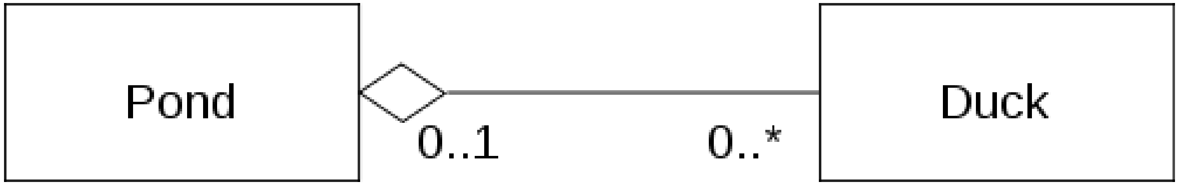
Association Link

Links Class/Attributes

Uni or bi-directional



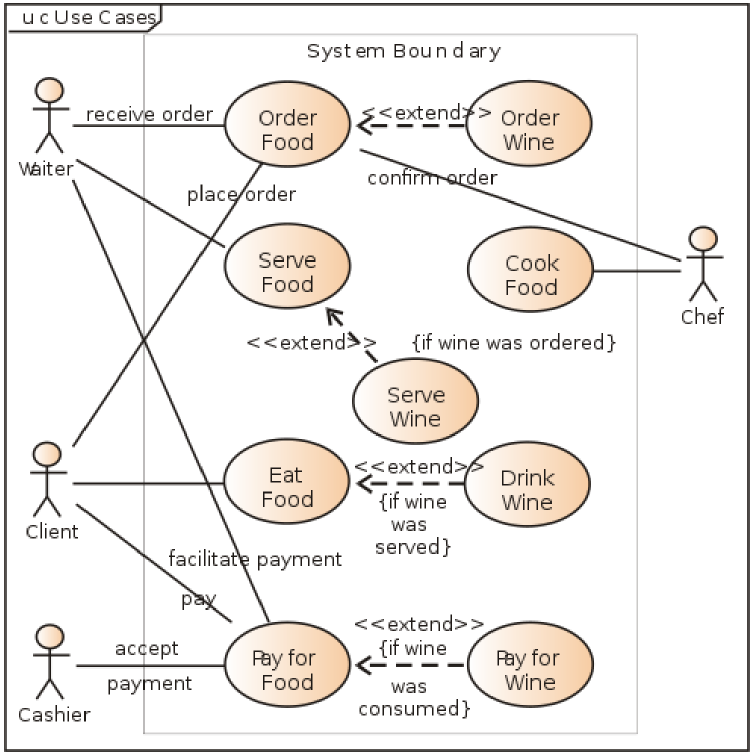
Composition
Car has one Carburetor
Carburator may be part of a Car



Aggregation
Pond may have one or more ducks
Duck may have a pond

http://en.wikipedia.org/wiki/Class_diagram

UML Use Case Diagram



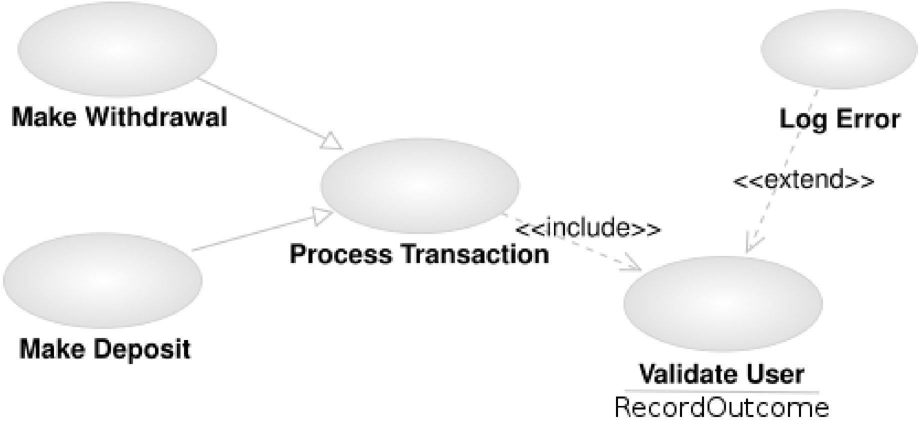
Actors shown as a stick person

Use Case shown as an oval

System boundary is a rectangular box

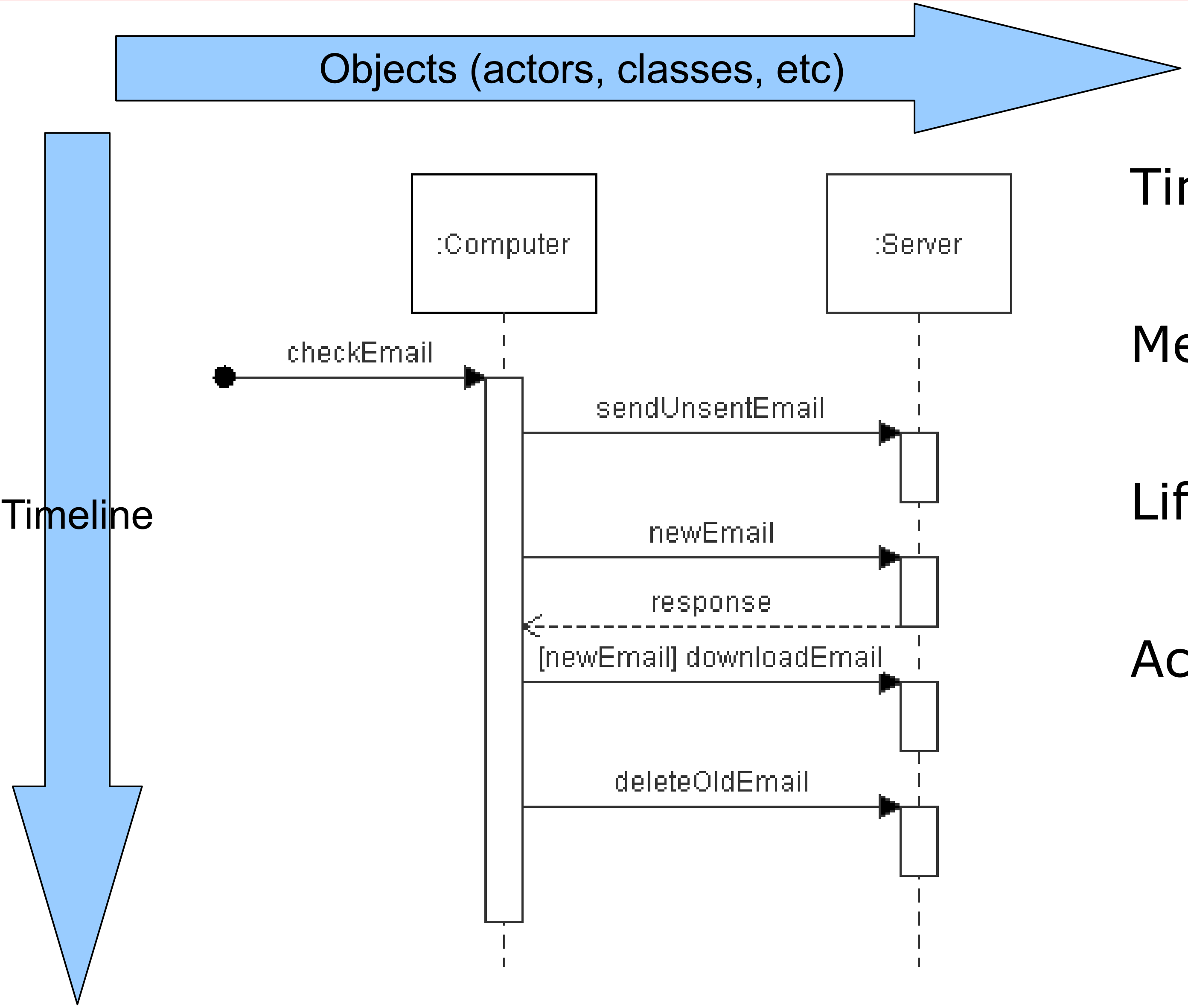
Reuse Use Cases shown with the <<include>> stereotype

Variant Use Cases shown with the <<extends>> stereotype



http://en.wikipedia.org/wiki/Use_case_diagram

UML Sequence Diagram



Objects (actors, classes, etc)

Timeline

Time runs from top to bottom

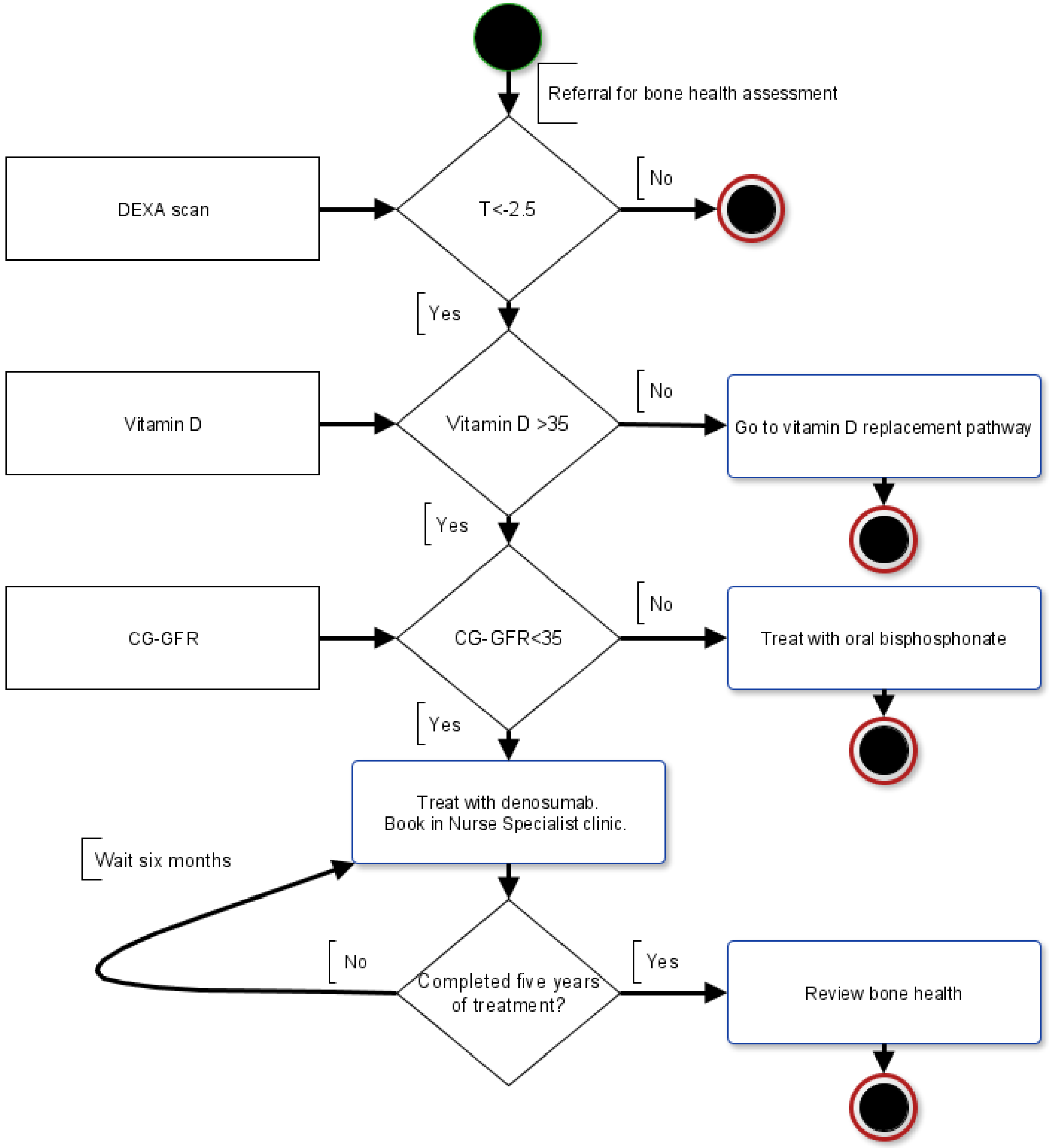
Message originates from external source

Lifeline shown with dotted line from object

Activation box on top of swimlane shows response to message

http://en.wikipedia.org/wiki/Sequence_diagram

UML Activity Diagram


Referral for bone health assessment
DEXA scan
T<-2.5
No
Yes
Vitamin D
Vitamin D >35
No
Go to vitamin D replacement pathway
Yes
CG-GFR
CG-GFR<35
No
Treat with oral bisphosphonate
Yes
Treat with denosumab.
Book in Nurse Specialist clinic.
Wait six months
No
Completed five years of treatment?
Yes
Review bone health

Can be used to model workflow

E.g. for care pathways

(more formal than flow charts)

http://en.wikipedia.org/wiki/Activity_diagram

Requirements for health records systems

Requirements Specification

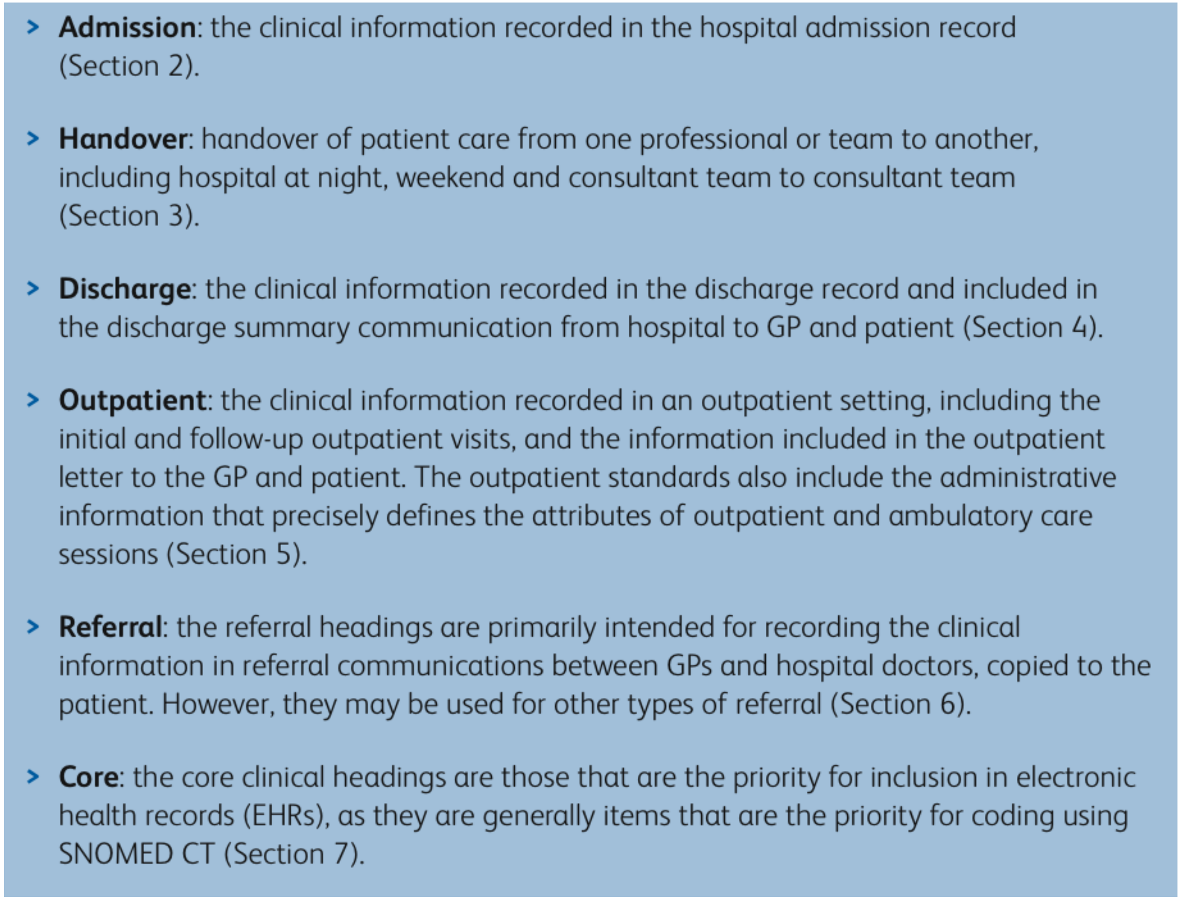
- Key artefacts from the Unified Process
  - Vision document
  - Use Cases
  - Software Requirements Specification
- Get input from existing sources
  - CCHIT (US)

    http://www.cchit.org
  - HL7 EHR Functional Model

    http://www.hl7.org
  - NpfIT OBS – hard to find now online!
  - RCP Health Record requirements

    http://www.rcplondon.ac.uk/clinical-standards/hiu/Pages/Record-keeping.aspx

RCP Clinical Headings

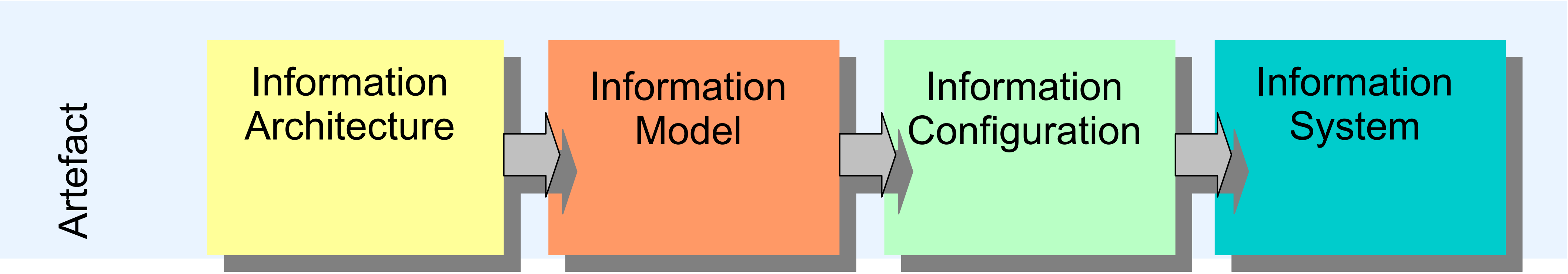


- **Admission**: the clinical information recorded in the hospital admission record (Section 2).
- **Handover**: handover of patient care from one professional or team to another, including hospital at night, weekend and consultant team to consultant team (Section 3).
- **Discharge**: the clinical information recorded in the discharge record and included in the discharge summary communication from hospital to GP and patient (Section 4).
- **Outpatient**: the clinical information recorded in an outpatient setting, including the initial and follow-up outpatient visits, and the information included in the outpatient letter to the GP and patient. The outpatient standards also include the administrative information that precisely defines the attributes of outpatient and ambulatory care sessions (Section 5).
- **Referral**: the referral headings are primarily intended for recording the clinical information in referral communications between GPs and hospital doctors, copied to the patient. However, they may be used for other types of referral (Section 6).
- **Core**: the core clinical headings are those that are the priority for inclusion in electronic health records (EHRs), as they are generally items that are the priority for coding using SNOMED CT (Section 7).

- Developed by clinicians
- Published by RCP / HSCIC
- Endorsed by all Royal Colleges
- A good checklist
- May not be practical to implement

Logical design of a health record system

From Architecture to System



- The modelling process starts with *architecture*
  - form, techniques and materials
- From which an information model is built
- Which underlies the specific system configuration
- That drives the operational system

Logical and Physical Models

Logical data models

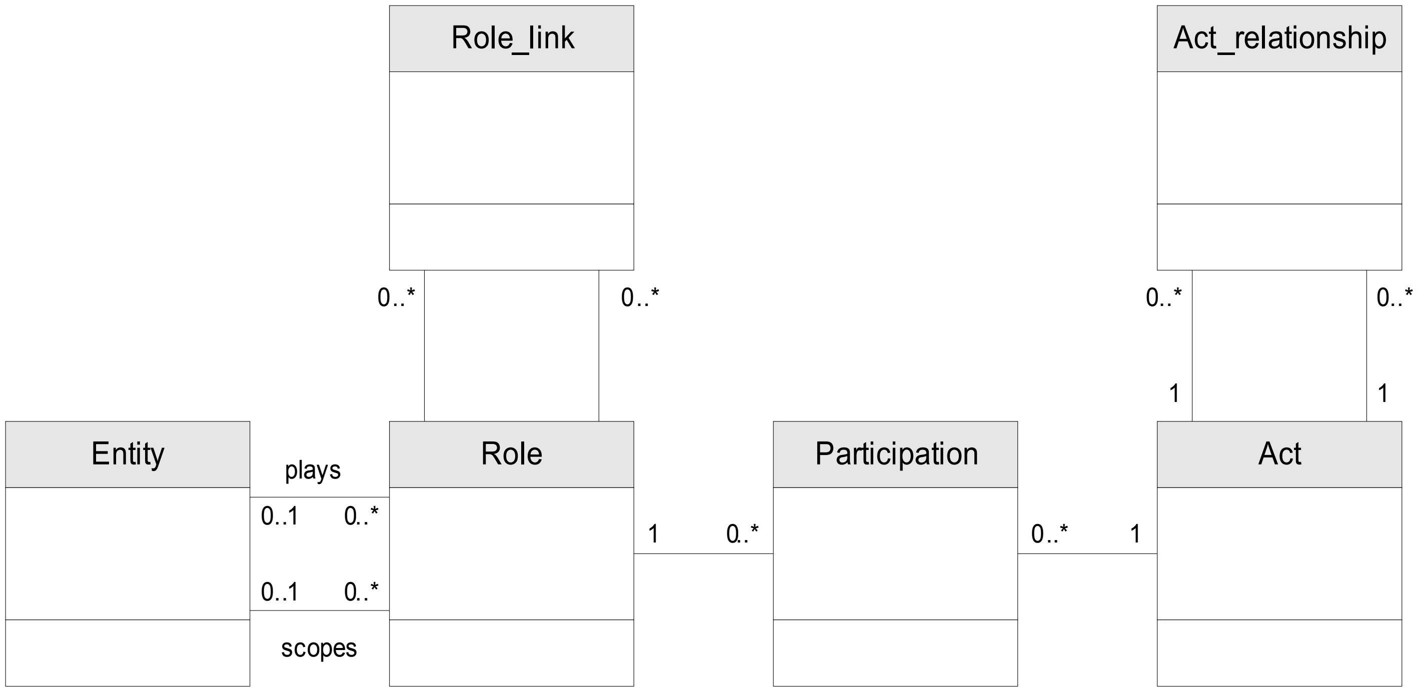
Logical data models represent the abstract structure of some domain of information. They are often diagrammatic in nature and are most typically used in business processes that seek to capture things of importance to an organization and how they relate to one another. http://en.wikipedia.org/wiki/Logical_data_model

Physical data models

A physical data model (a.k.a. database design) is a representation of a data design which takes into account the facilities and constraints of a given database management system. In the lifecycle of a project it is typically derived from a logical data model. http://en.wikipedia.org/wiki/Physical_data_model

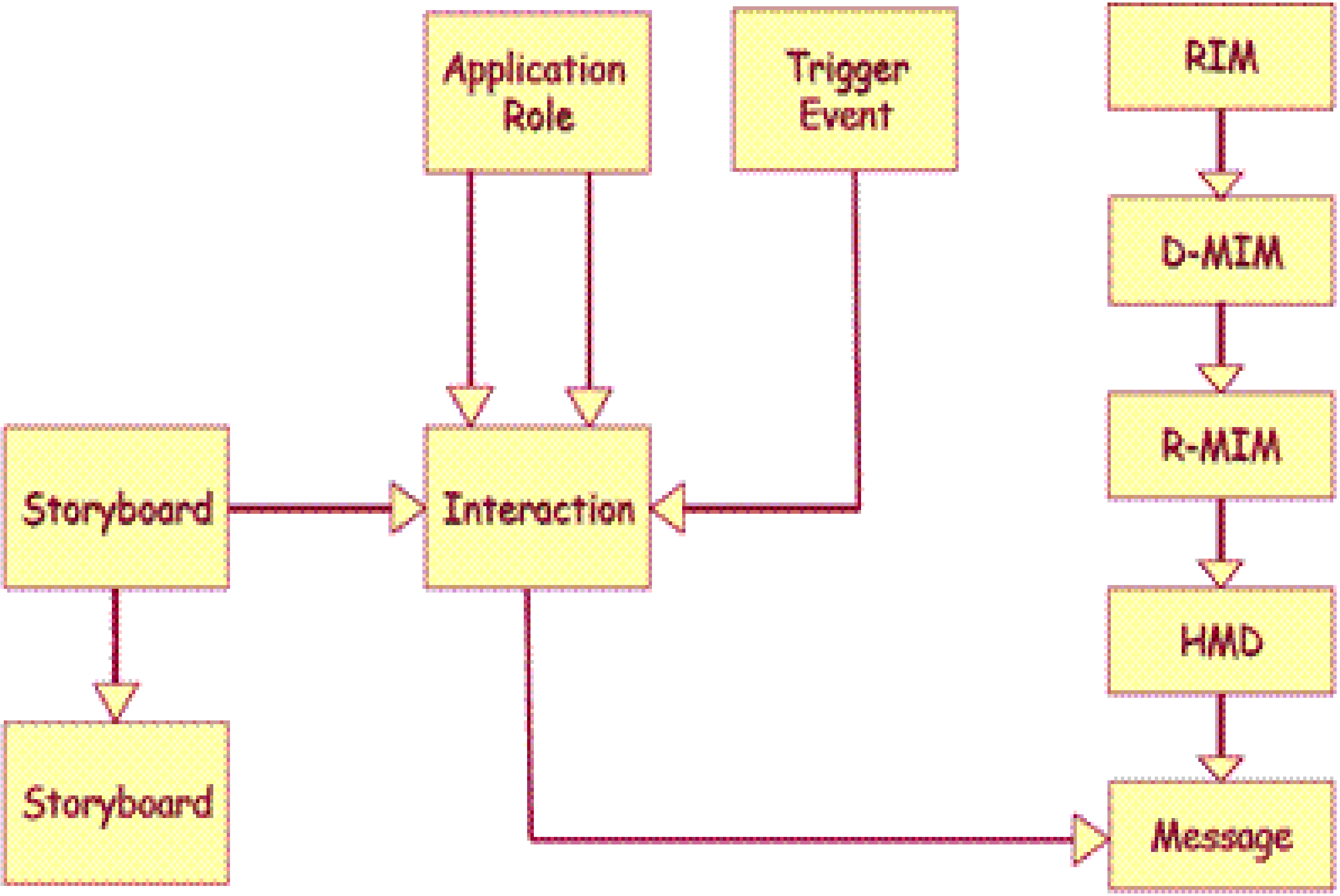
When implementing XML-based systems we are concerned with moving from the logical model to an XML representation of the main data elements.

HL7 Logical Model



- HL7 – Reference Information Model
  - Has six core classes

HL7 – Modelling Methodology



HL7 CDA – A Physical Model

CDA Header

```xml
<typeId extension="#CityEHR:Message:A31Registration" root="cityEHR"/>
    <templateId extension="#ISO-13606:Folder:Rheumatology" root="#ISO-13606:EHR_Extract:ORCHID"/>
    <id extension="timeStamp" root="cityEHR"/>
    <code code="134436002" codeSystem="2.16.840.1.113883.2.1.3.2.4.15" codeSystemName="SNOMED" displayName="Patient
Registration"/>
    <effectiveTime value="2012-01-06T09:48:44.14Z"/>
    <recordTarget>
        <patientRole>
            <id extension="K1287932" root="2.16.840.1.113883.2.1.4.1"/>
            <patient>
                <name>
                    <prefix/>
                    <given>Timothy</given>
                    <family>Abernathy</family>
                </name>
                <administrativeGenderCode code="M" codeSystem="2.16.840.1.113883.2.1.3.2.4.15" displayName="Male"/>
                <birthTime value="1979-03-28"/>
            </patient>
            <providerOrganization>
                <id extension="" root="2.16.840.1.113883.2.1.4.3"/>
            </providerOrganization>
        </patientRole>
    </recordTarget>
    <author>
        <time value=""/>
        <assignedAuthor>
            <id extension="forename.surname" root="#ISO-13606:EHR_Extract:cityEHR"/>
            <assignedPerson>
                <name>cityEHR User</name>
            </assignedPerson>
            <authoringDevice>
                <softwareName>cityEHR Base Application</softwareName>
            </authoringDevice>
            <representedOrganization>
                <id root="#ISO-13606:EHR_Extract:cityEHR"/>
                <name/>
            </representedOrganization>
        </assignedAuthor>
    </author>
```

HL7 CDA – A Physical Model

CDA Structured Body

```
<structuredBody>
            <component>
                <section cityEHR:layout="Ranked">
                    <id extension="#ISO-13606:Section:Demographics" root="cityEHR"/>
                    <title>Patient Demographics</title>
                    <entry cityEHR:initialValue="#CityEHR:EntryProperty:Default"
                       cityEHR:layout="Unranked"
                      cityEHR:rendition="#CityEHR:EntryProperty:Form">
                        <observation>
                            <typeid extension="Type:Observation" root="cityEHR"/>
                            <id extension="#ISO-13606:Entry:Knumber" root="cityEHR"/>
                            <code code="xxxx" codeSystem="" displayName="K Number"/>
                            <value code="" codeSystem="" displayName=""
                              cityEHR:elementType="#CityEHR:ElementProperty:simpleType"
                              extension="#ISO-13606:Element:String"
                              root="cityEHR" xsi:type="xs:string" units="" value="K1287932"
                              cityEHR:valueRequired="#CityEHR:ElementProperty:Optional"/>
                        </observation>
                    </entry>
                </section>
            </component>
        </structuredBody>
```

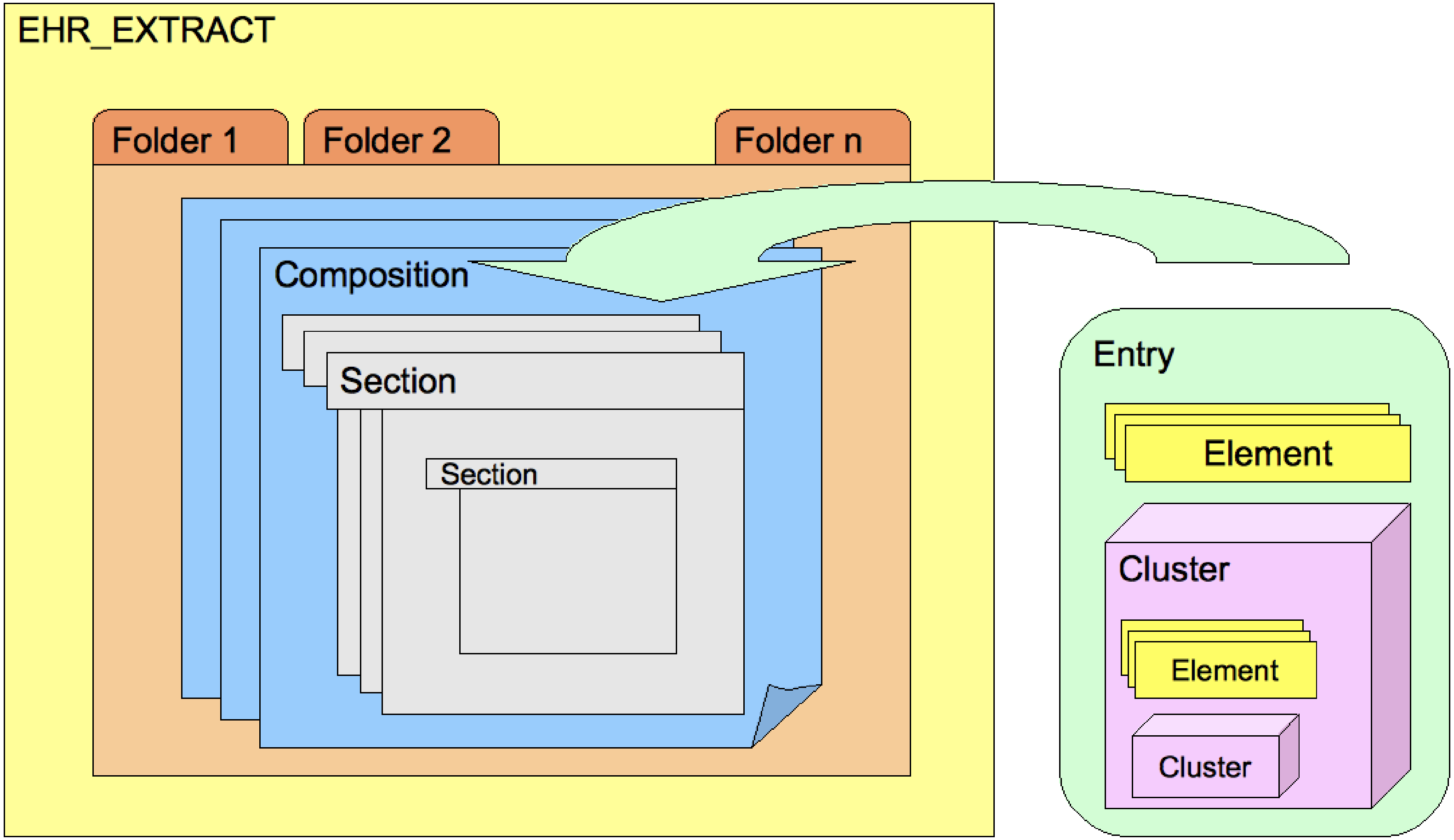
OpenEHR www.openehr.org

- An open consortium whose mission is
  - "To improve the clinical care process by fostering the development and implementation of open source, interoperable EHR components. These components should be based on internationally agreed requirements and address the need for privacy and security, while supporting the development of interoperable and evolving clinical applications."

- The EHR Information Model has been based on EHR implementation experience from the GEHR, Synapses, SynEx and Australian GEHR projects, and the EHCR-SupA and ISO 13606 models.

- Access here to a set of Archetypes – architectural components, from which logical models can be built, based on ISO 13606

ISO 13606 EHR Model



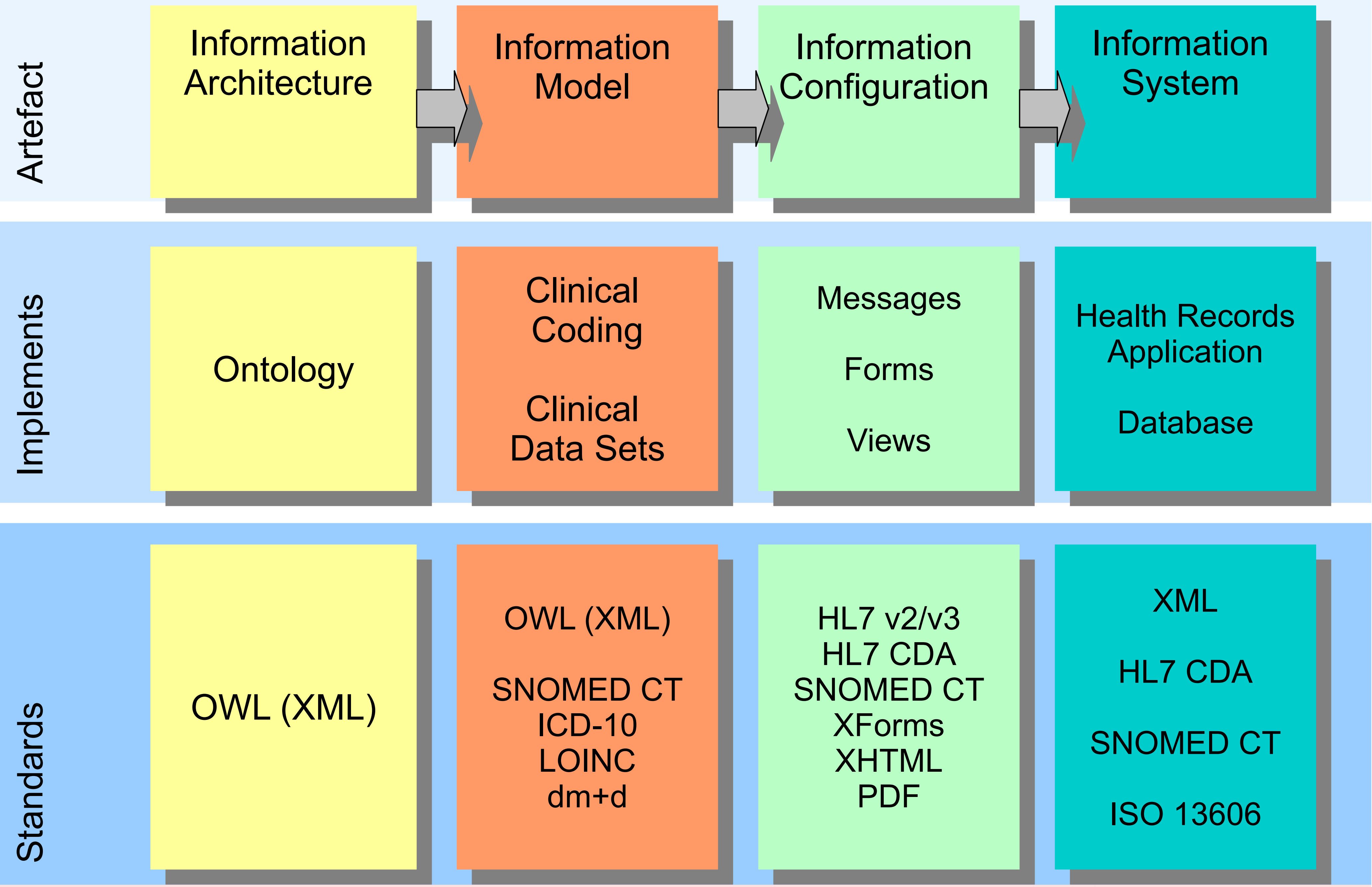
XML Physical Model

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ehr_extract>
    <folder>
        <composition>
            <code code="11488-4" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Consultation note"/>
            <section>
                <entry>
                    <element>Some data here</element>
                    <cluster>
                        <element>Some data here</element>
                    </cluster>
                </entry>
            </section>
        </composition>
    </folder>
</ehr_extract>
```

HL7 CDA / ISO 13606 Compared

ISO-13606	HL7 CDA
extract	Not explicitly included in CDA, although could go for <templateId extension="#ISO-13606:Folder:Rheumatology" root="#ISO-13606:EHR_Extract:ORCHID"/>
folder	Not explicitly included in CDA, although could go for <templateId extension="#ISO-13606:Folder:Rheumatology" root="#ISO-13606:EHR_Extract:ORCHID"/>
composition	<ClinicalDocument>
section	<section>
entry	<entry>
cluster	Not included, but HL7 CDA has an <organizer> element which can be used for collections of <entry> i.e. at a level above the ISO-13606 model
element	<value>

From Architecture to System



Artefact	Information Architecture	Information Model	Information Configuration	Information System
Implements	Ontology	Clinical Coding Clinical Data Sets	Messages Forms Views	Health Records Application Database
Standards	OWL (XML)	OWL (XML) SNOMED CT ICD-10 LOINC dm+d	HL7 v2/v3 HL7 CDA SNOMED CT XForms XHTML PDF	XML HL7 CDA SNOMED CT ISO 13606

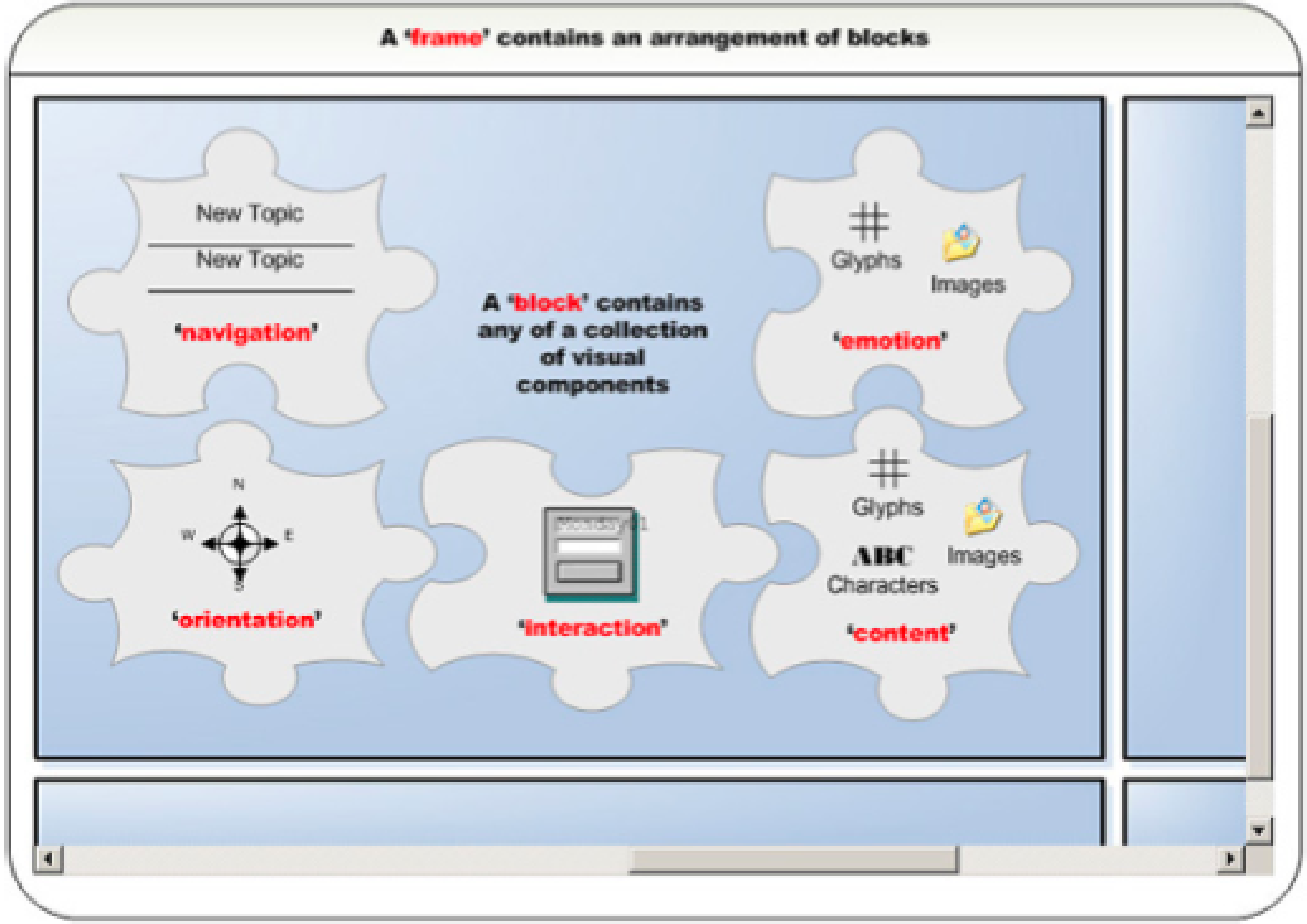
Designing a clinical user interface

User Interface Design Principles

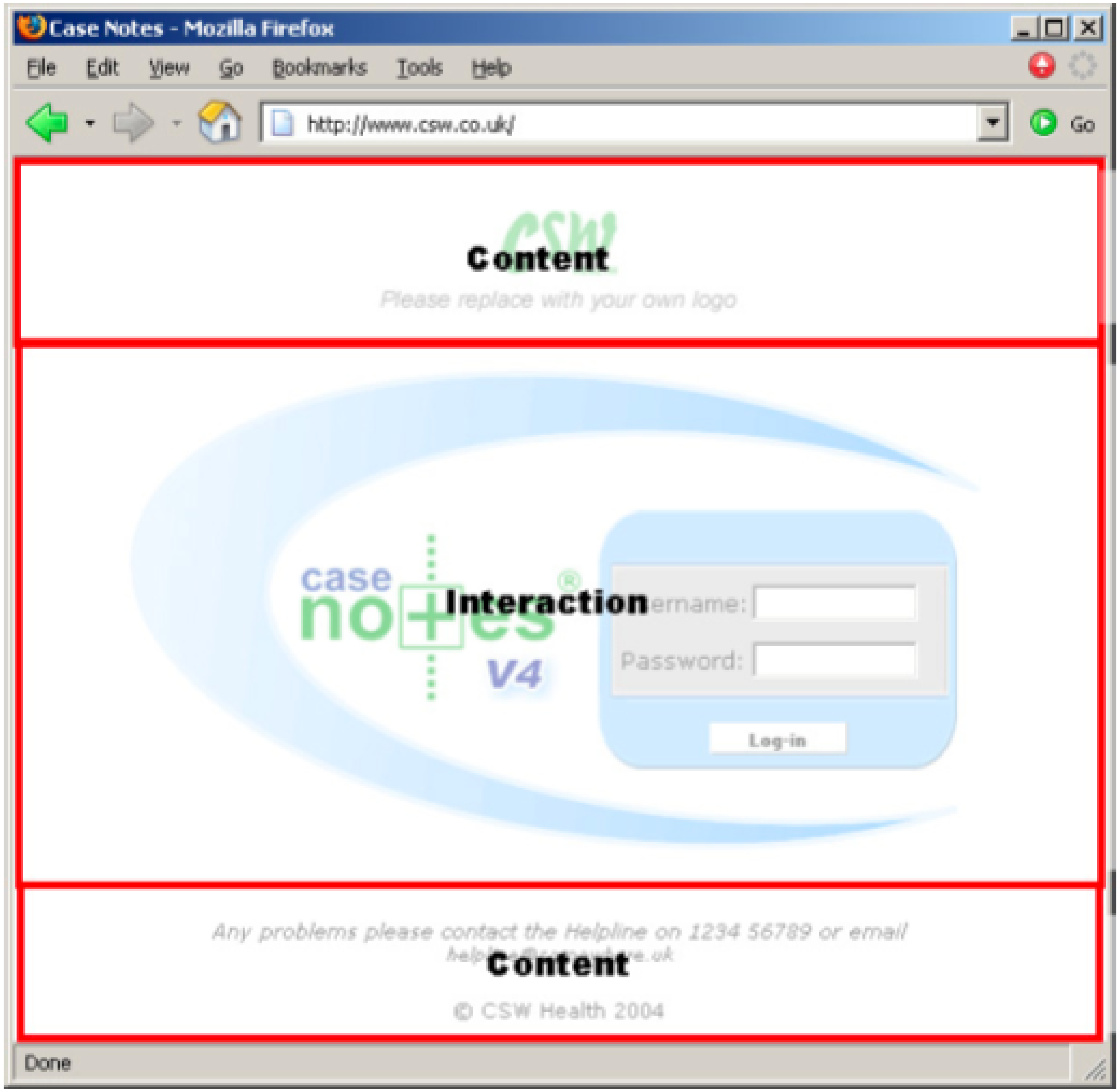
- **The structure principle:** Design should organize the user interface purposefully...
- **The simplicity principle:** The design should make simple, common tasks easy,...
- **The visibility principle:** The design should make all needed options and materials for a given task visible without distracting the user...
- **The feedback principle:** The design should keep users informed of actions or interpretations,...
- **The tolerance principle:** The design should be flexible and tolerant, reducing the cost of mistakes...
- **The reuse principle:** The design should reuse internal and external components...

Constantine and Lockwood

Interface Component Model



Mapping in a Commercial Product



Case Notes - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.csw.co.uk/
Go
csw
Content
Please replace with your own logo
case notes V4
Interaction
Username:
Password:
Log-in
Any problems please contact the Helpline on 1234 56789 or email helpline@csw...e.uk
Content
© CSW Health 2004
Done

UI for Clinical Applications

- Some of the rules of good UI design are different for clinical systems
- This is my list of key points
  - Limited user configuration and preferences
    - No persistence between sessions
  - Very flat navigational structure
    - Ideally no menus or complex navigation
  - Consistent use of interface components
    - Same component and function in fixed position
  - Clear separation of patient context
    - Non-patient, cross-patient, patient-specific

NHS Common User Interface

- User interface guidance and framework for NHS systems
- NHS Common User Interface (CUI)
- Developed in collaboration with Microsoft
  - http://www.cui.nhs.uk
  - http://www.mscui.net/

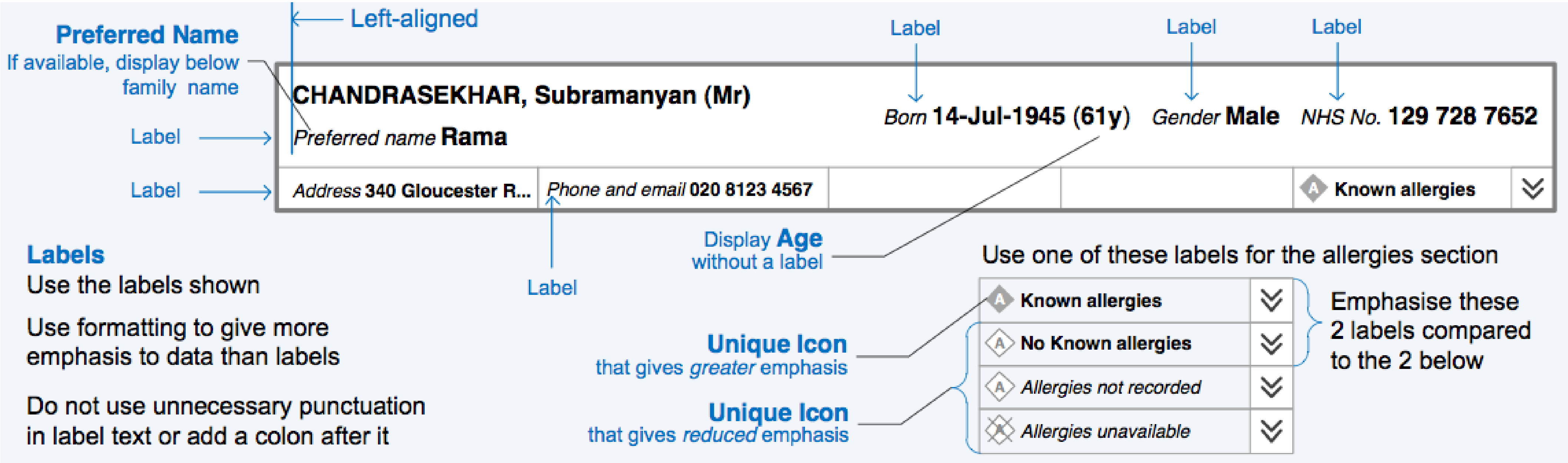
Very Microsoft-centric, and caught up in political / commercial dealings with vendors for the NPfIT

Design Guidance – Accessibility Principles

- 1. Support standard system size, colour, font, input settings, and accessibility options
- 2. Enable programmatic access to user interface elements and text
- 3. Provide keyboard access to all features
- 4. Expose the location of the keyboard focus
- 5. Provide equivalents for non-text elements
- 6. Do not rely exclusively on a single perceptual capability to convey information
- 7. Avoid flashing elements
- 8. Enable user control of timed information presentation and responses
- 9. Ensure consistency between interface elements and display items
- 10. Create accessible documentation about accessibility features

http://www.mscui.net/DesignGuide/accessibilityprinciples.aspx

CUI Example – Patient Banner



Left-aligned

Preferred Name
If available, display below family name

Label

Label

Label

Label

Label

CHANDRASEKHAR, Subramanyan (Mr)
Preferred name **Rama**

Born **14-Jul-1945 (61y)** *Gender* **Male** *NHS No.* **129 728 7652**

Address **340 Gloucester R...** *Phone and email* **020 8123 4567** **Known allergies**

Label

Display **Age**
without a label

Labels
Use the labels shown

Use formatting to give more emphasis to data than labels

Do not use unnecessary punctuation in label text or add a colon after it

Unique Icon
that gives *greater* emphasis

Unique Icon
that gives *reduced* emphasis

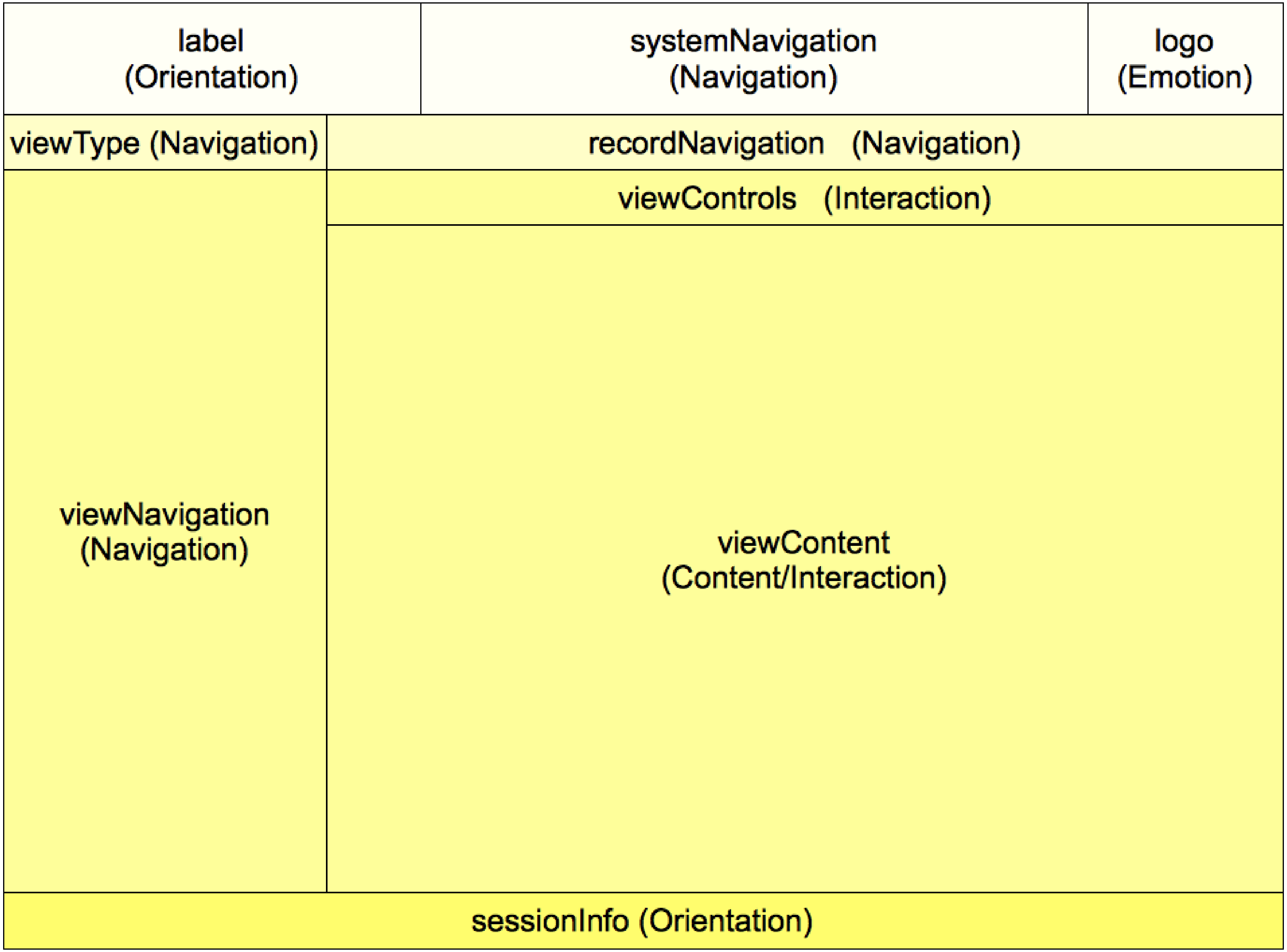
Use one of these labels for the allergies section

Known allergies
No Known allergies
Allergies not recorded
Allergies unavailable

Emphasise these 2 labels compared to the 2 below

- CUI contains guidance on how common UI components should be implemented
- Has close connection with logical models and data dictionary (data types)

City EHR – Open Source EHR



label (Orientation)

systemNavigation (Navigation)

logo (Emotion)

viewType (Navigation)

recordNavigation (Navigation)

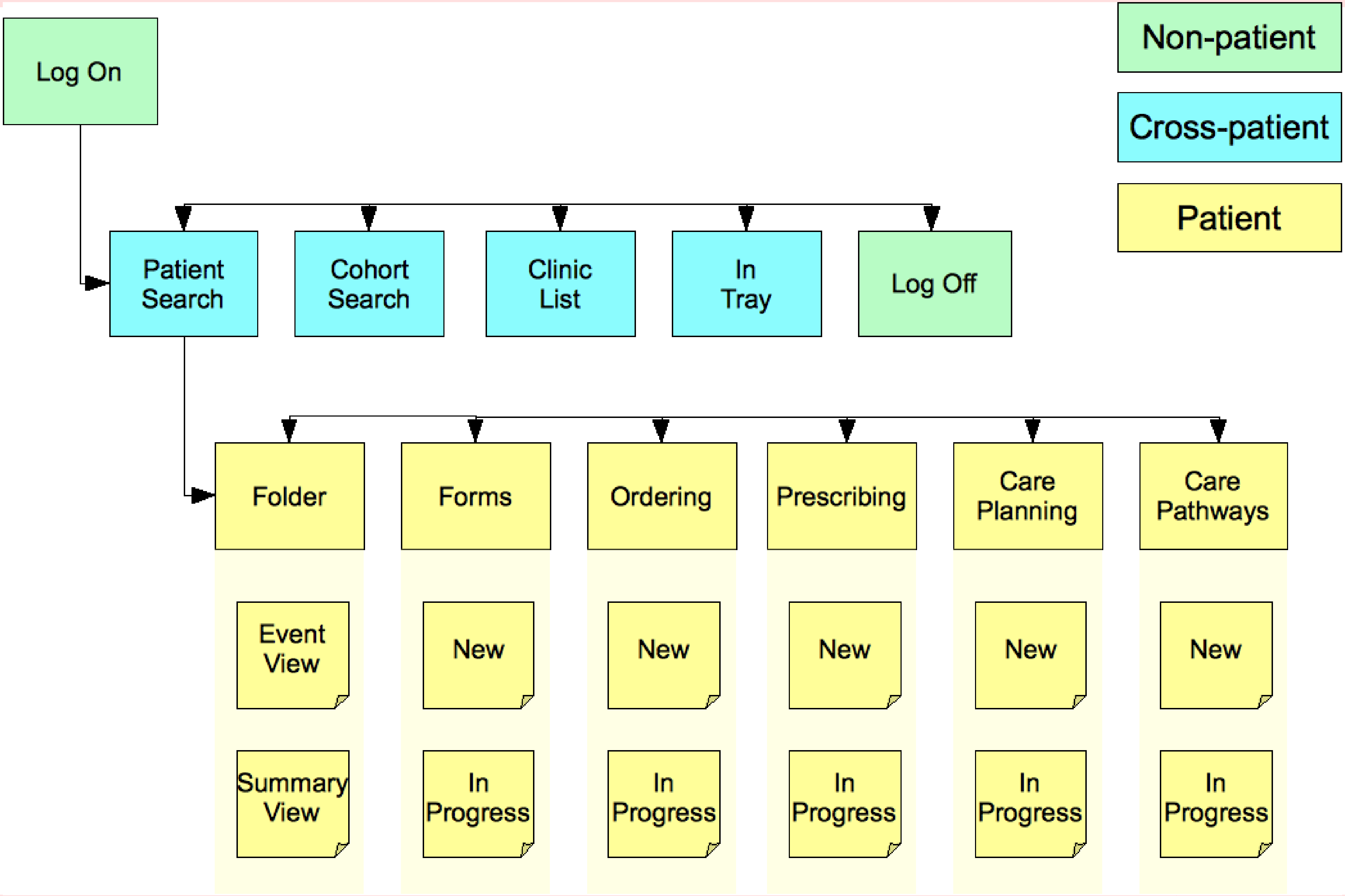
viewControls (Interaction)

viewNavigation (Navigation)

viewContent (Content/Interaction)

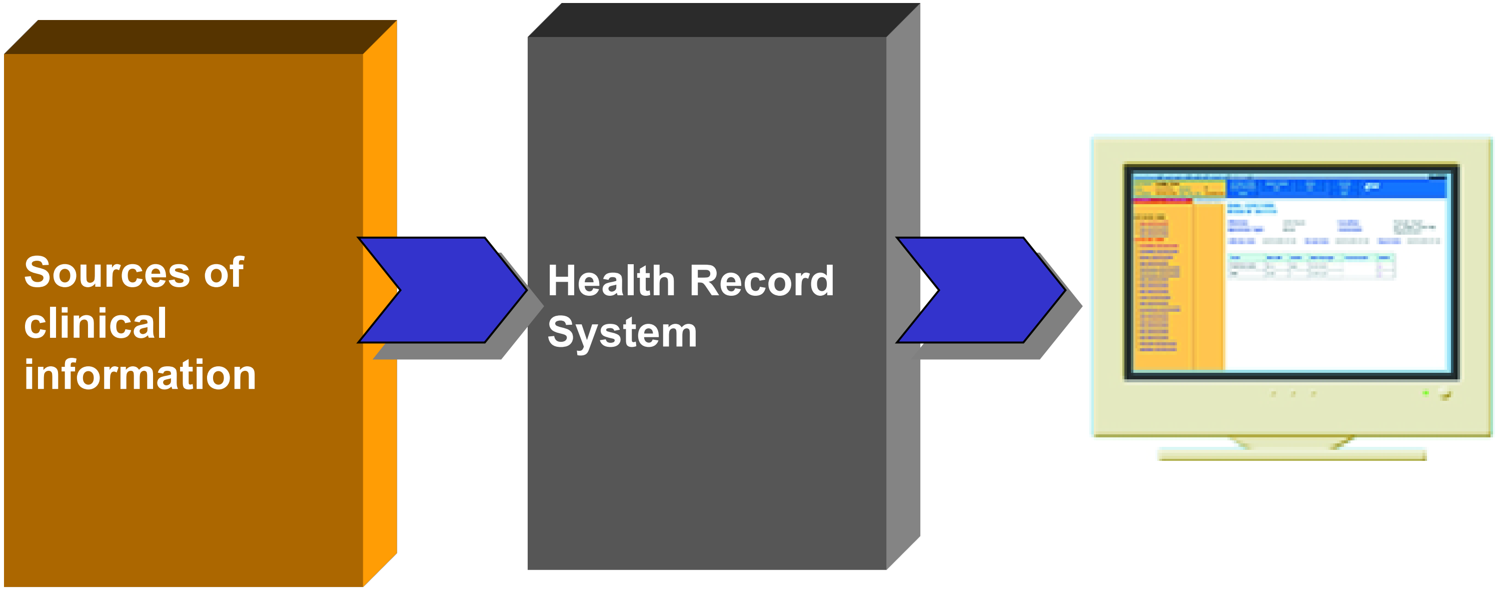
sessionInfo (Orientation)

City EHR – Site Map



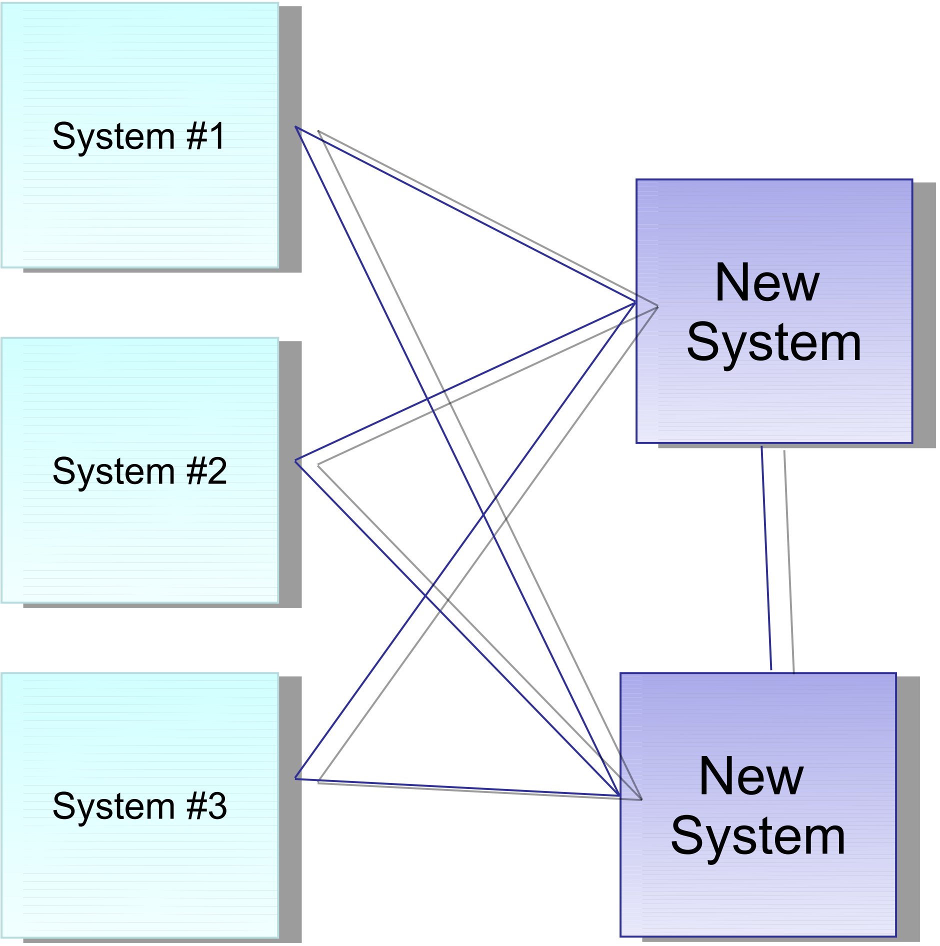
Integration of clinical systems

Logical View of the Health Record



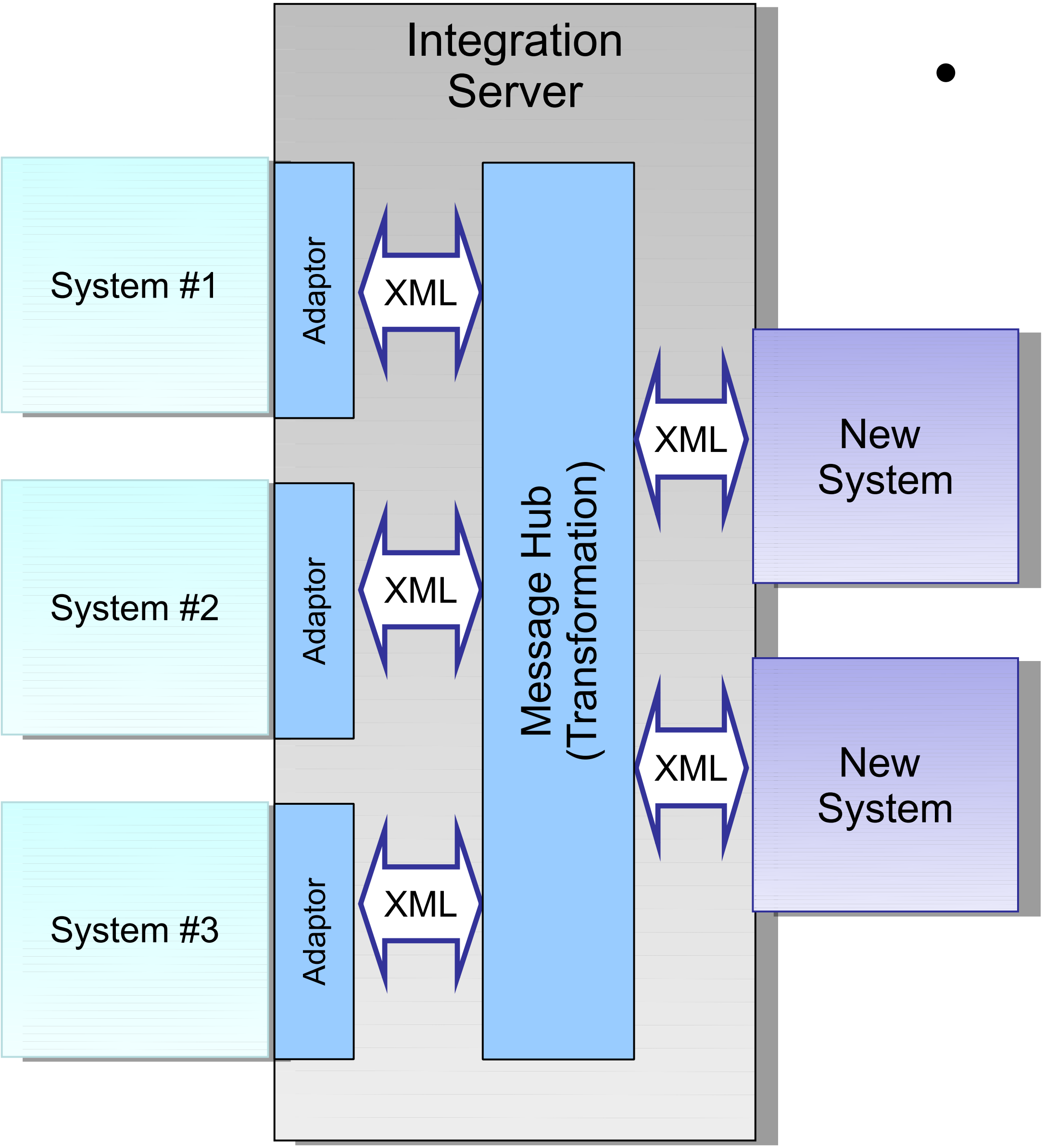
- Health record consists of clinical information from a variety of source systems or modules
- Health record system reorganises into patient centred views tailored for the context of care
- The question is whether the health record system maintains its own persistent store of the clinical information.

Point-to-Point Integration



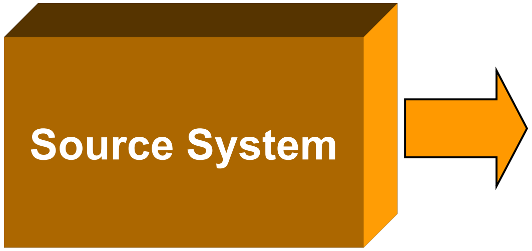
- Individual interfaces between systems
- Highest number of interfaces required
- Interfaces break when systems change
- Each interface can use different technology

XML-Based Integration

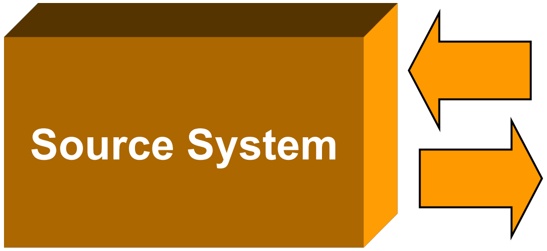


- Adaptors wrap each system
  - ODBC/JDBC
  - APIs
  - Text file import/export
  - CSV
  - Screen scrape
-
  - Integration engine maps XML interfaces
  - System change = mapping, not adapter
  - Simpler, more flexible, cheaper

Interfaces to Source Systems

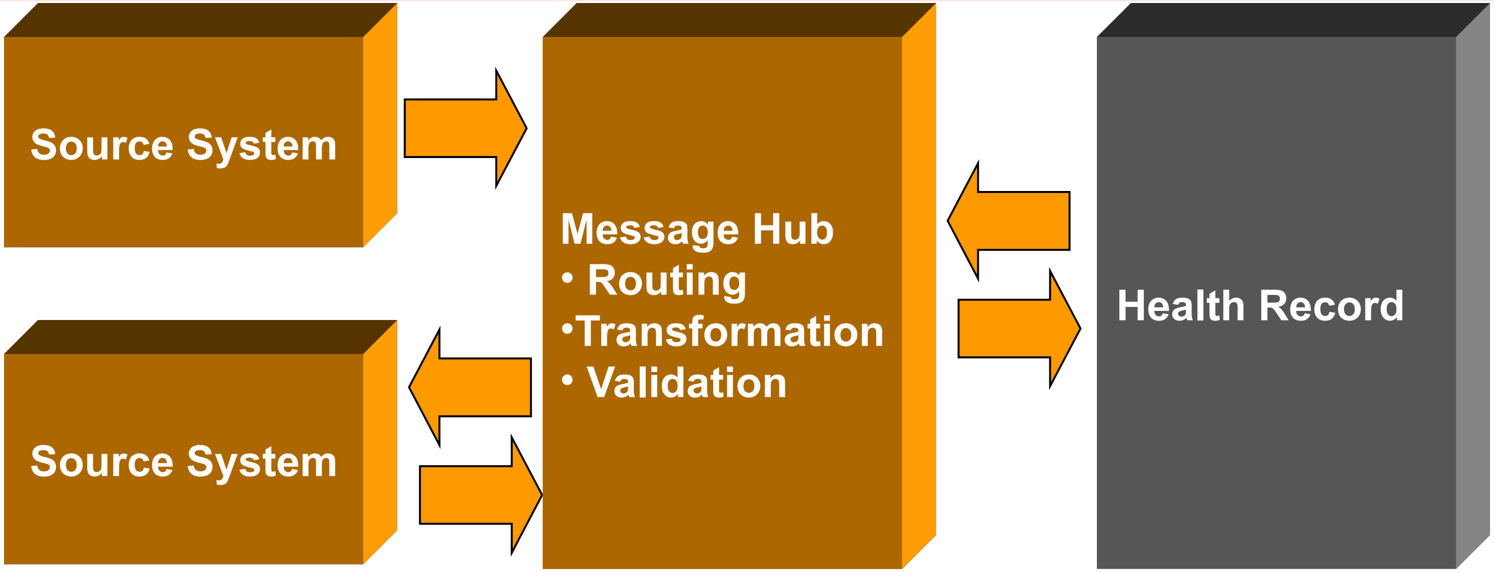


- Push
  - Messages are sent from the source system, without direct request from the records system
  - Synchronous
    - Message is generated whenever relevant information changes in the source system
  - Asynchronous
    - Message(s) generated at predefined intervals, for a batch of information changes



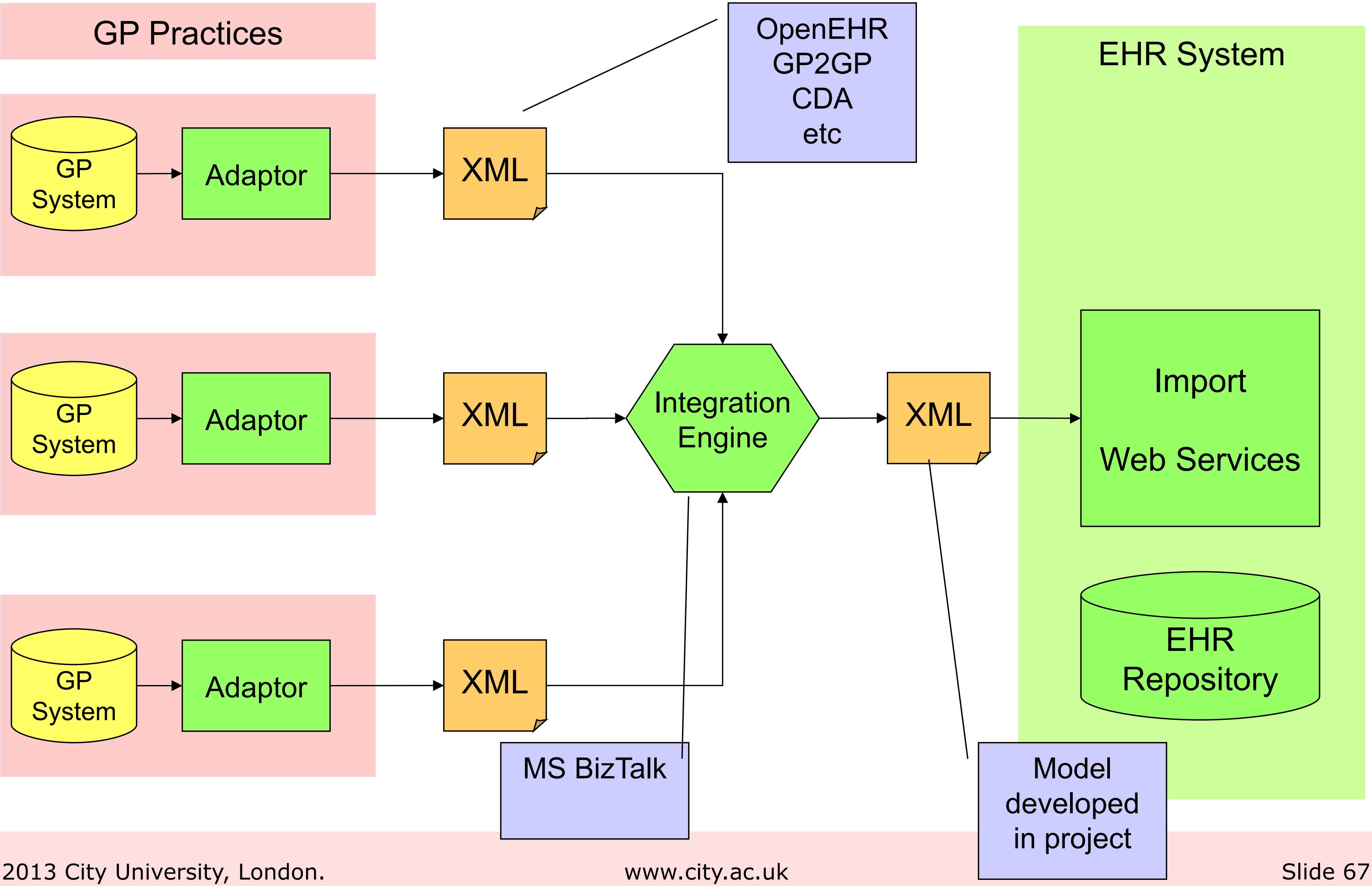
- Pull
  - Message sent from records system requests information from the source
  - Synchronous
    - Request is sent from the records system whenever it needs information
  - Asynchronous
    - Request is sent from records system at predefined intervals, response contains a batch of information

Interfaces with Messaging Hub

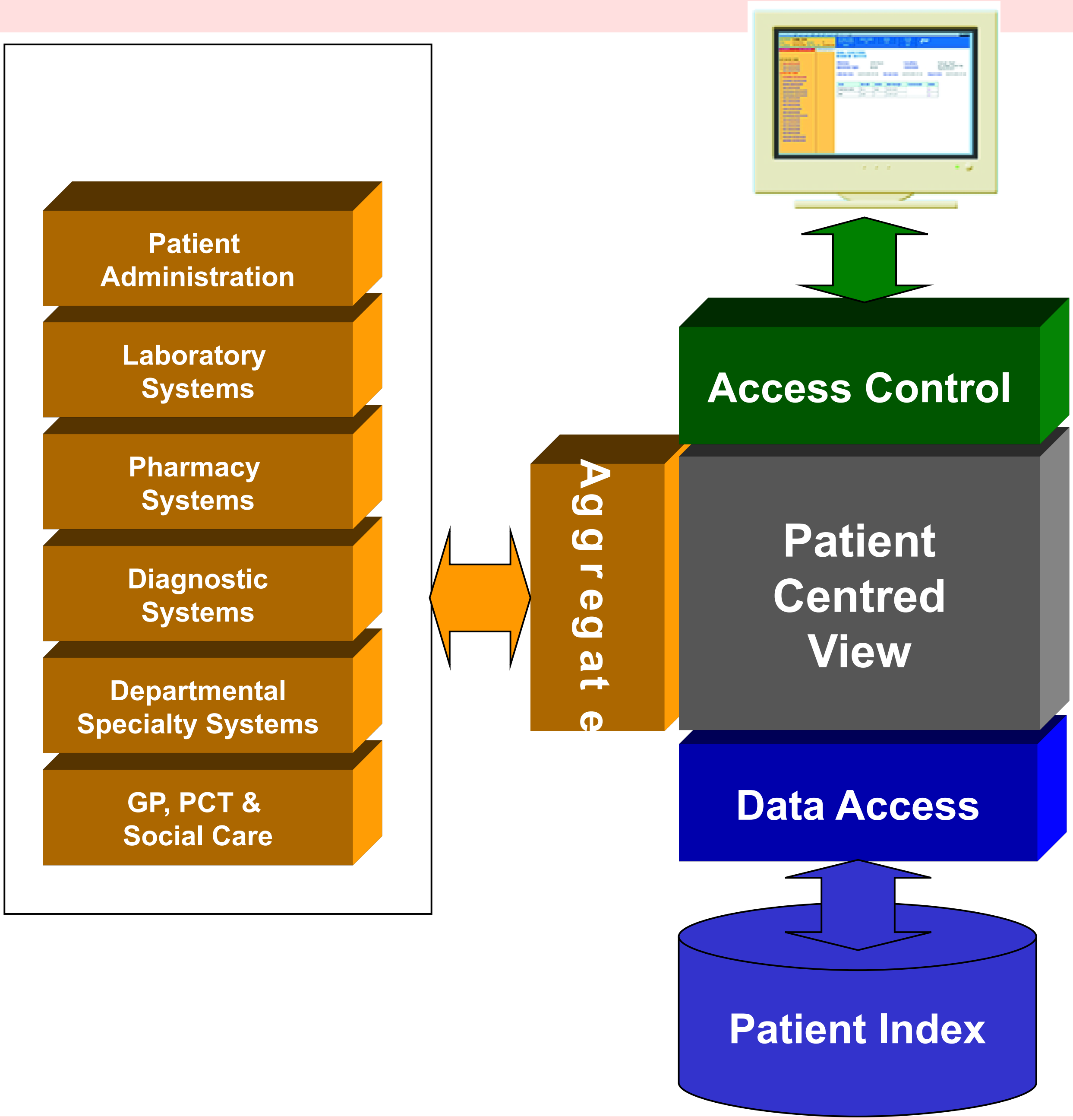


- Message hub buffers between source system interfaces and the records system
  - Routing
    - Can route messages between source systems, the health record and other receiving systems
  - Transformation
    - Transform message format from source interfaces to standard message formats (e.g. HL7 v2 messages to XML)
  - Validation
    - Validate message structure (and completeness) and handle errors

Example - GP System Interfaces



Full Virtual Model



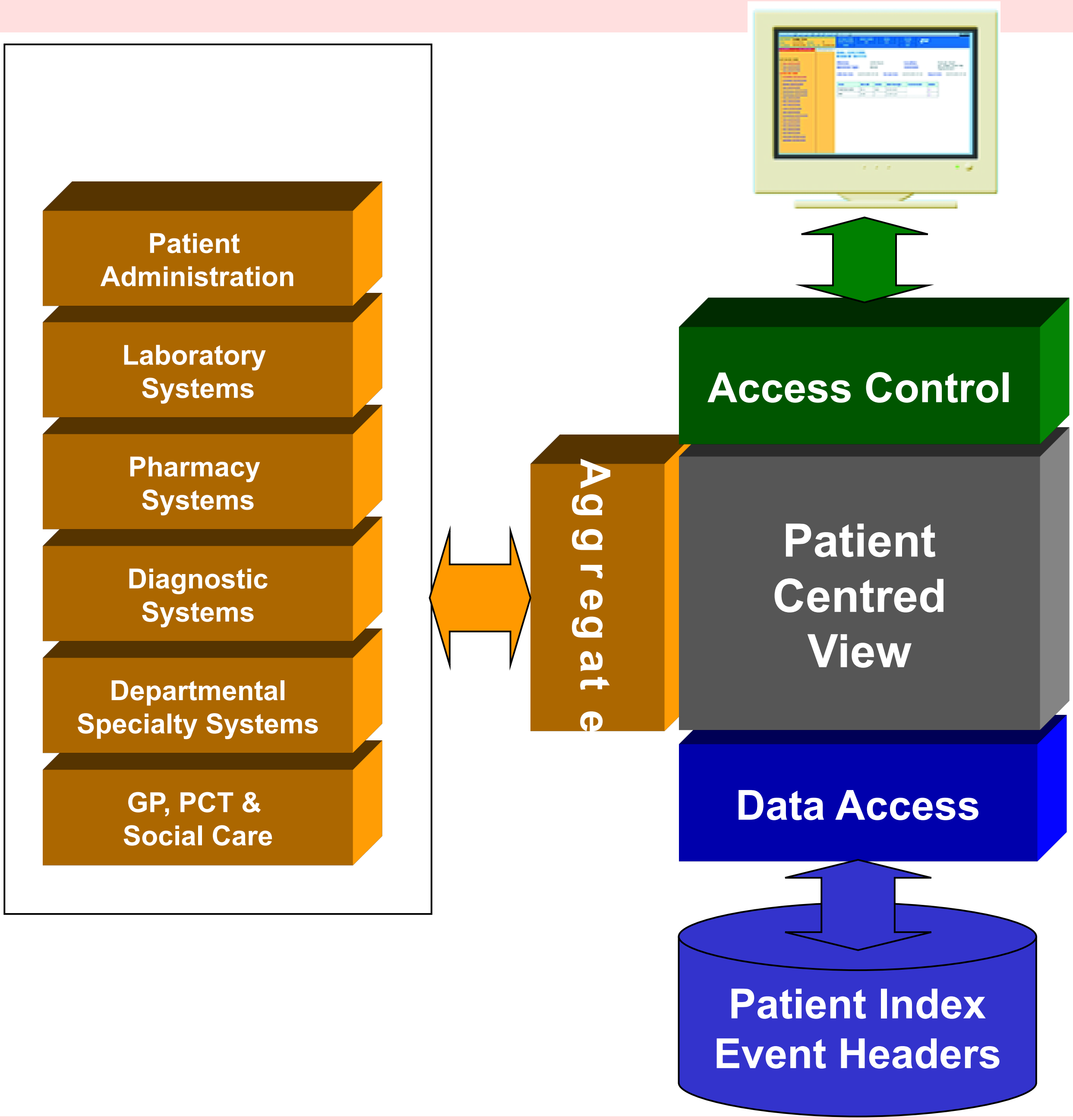
- Repository contains patient index only
- All record information pulled in real time from source systems
- Queries made to source systems using patient id
- Re-organise into patient-centred view in real time

Full Virtual Model Advantages and Disadvantages

- Advantages
  - Clinical data are stored and maintained in the current systems – low impact for users
  - Provided source systems are available, the most current information is always presented in the record

- Disadvantages
  - Cannot specify the ‘data set’ required for the full record
  - Time taken to retrieve the record view is unknown (and potentially large)
  - All processing of the record data for viewing must be ‘in real time’
    - adversely affects response times
  - Source systems may not be able to service the volume of information requests (many will not have been designed to work in this way)
  - If source systems are not available 24x7, the record will be incomplete (and possibly inconsistent)
  - Difficult (impossible?) to make a resilient system, with redundancy and back up
  - Difficult (impossible?) to create a comprehensive audit trail of the record as viewed by clinicians

Semi-virtual Model



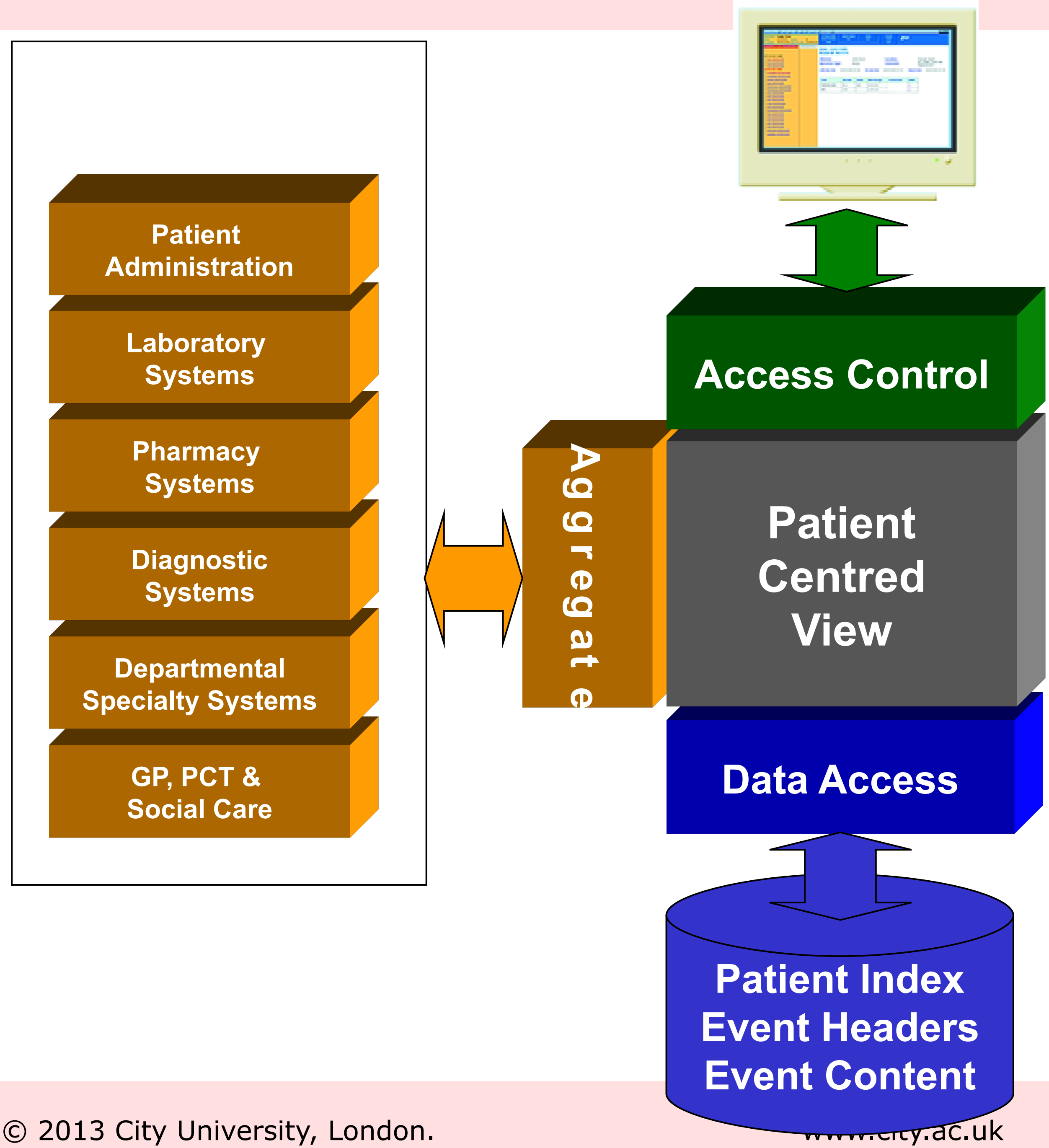
- Repository contains patient index and list of events
- Event header information extracted through push or pull
- Event headers reorganised into patient centred view
- Record information pulled in real time from source systems, using event headers
- Display as patient-centred view in real time

Semi-Virtual Model Advantages and Disadvantages

- Advantages
  - Clinical data are stored and maintained in the current systems – low impact for users
  - The outline of the record for each patient is always available, regardless of performance or availability of source systems
  - Provided source systems are available, the most current information is always presented in the record

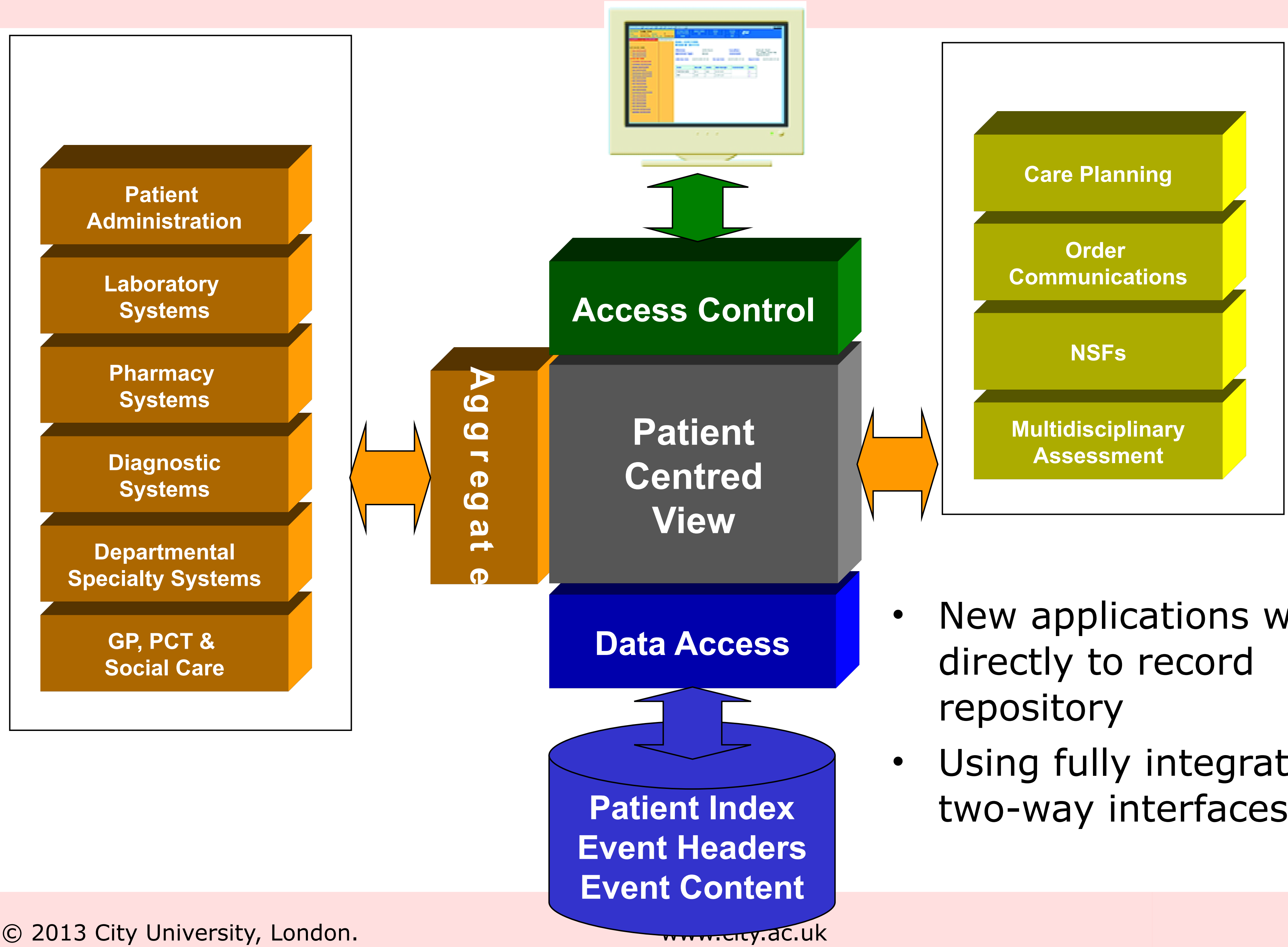
- Disadvantages
  - Time taken to retrieve the record view is unknown (and potentially large)
  - All processing of the record data for viewing must be ‘in real time’
    - adversely affects response times
  - Source systems may not be able to service the volume of information requests (many will not have been designed to work in this way)
  - If source systems are not available 24x7, the record will be incomplete (and possibly inconsistent)
  - Difficult (impossible?) to make a resilient system, with redundancy and back up
  - Difficult (impossible?) to create a comprehensive audit trail of the record as viewed by clinicians

Repository Model



- Repository contains patient index, list of events and full content of events
- Event information extracted through push or pull
- Events reorganised into patient centred view
- Display as patient-centred view in real time

Repository Model – New Applications



- New applications write directly to record repository
- Using fully integrated, two-way interfaces

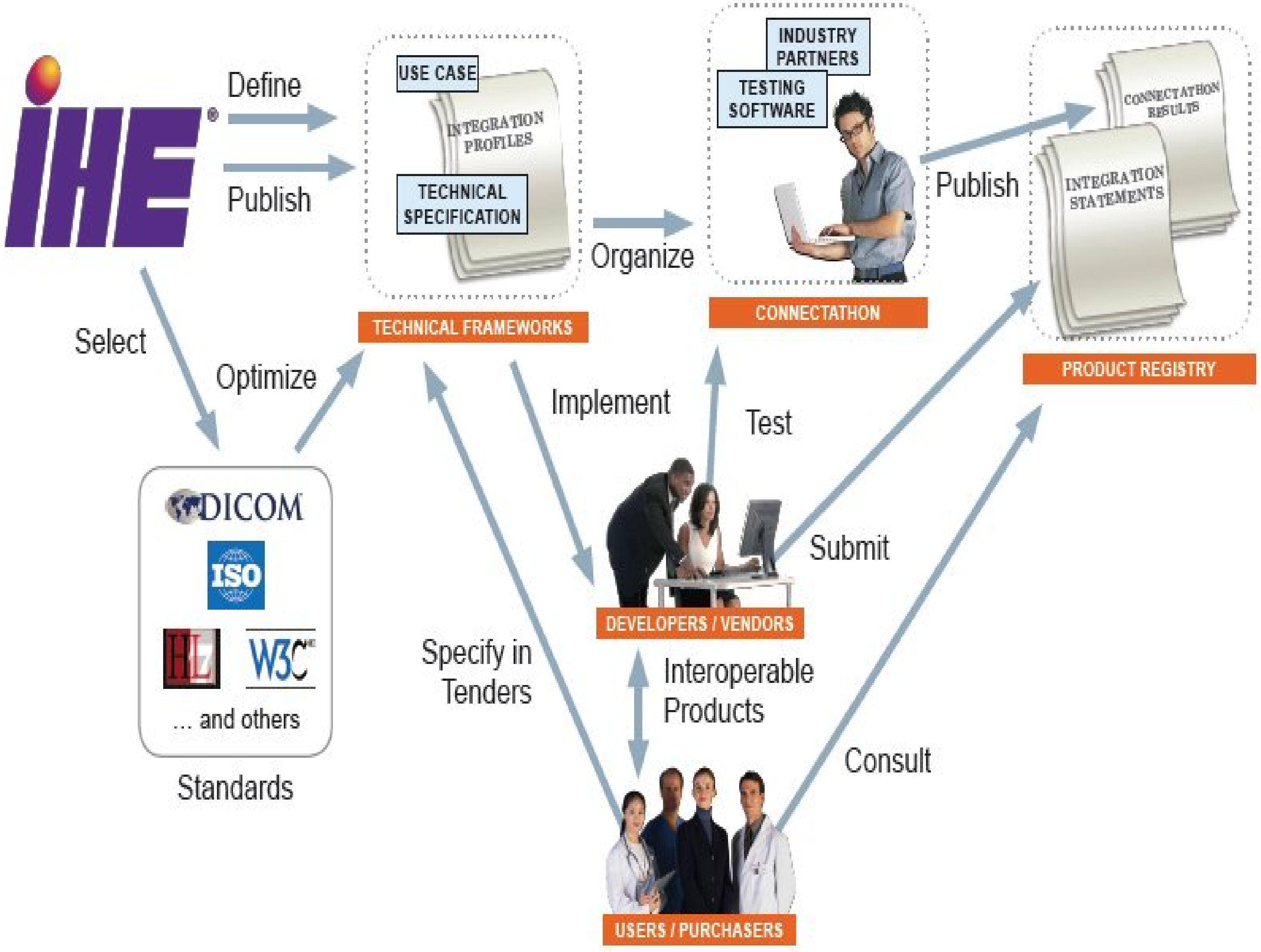
Repository Model

- Advantages
  - Can retrieve record data with rapid response times
  - Has core views already built for fast retrieval
  - Record is always available, regardless of availability of source systems
  - Can replicate and back up for a full resilient architecture
  - Can preserve full audit trail and history
- Disadvantages
  - Data from source systems is cached (ie stored twice)
  - The record may lag behind the actual information stored in source systems, depending on the frequency of update through interfaces
  - Source systems must update the repository when data change (in the source systems) in order to present a ‘current’ view of the record

Integrating the Healthcare Enterprise

- IHE is an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information.
- Promotes the coordinated use of established standards such as DICOM and HL7 to address specific clinical need in support of optimal patient care.
- Managed by IHE committees and sponsored by various national and international bodies.
- See www.ihe.net

The IHE Process



References and Further Reading

References

1. Jacobson, Booch and Rumbaugh (1999) The Unified Software Development Process. ISBN 0201571692

2. Kruchten P (2003) The Rational Unified Process – An Introduction. Addison-Wesley. ISBN 032119770-4

3. OpenUP. http://epf.eclipse.org/wikis/openup/

4. Arlow J and Neustadt I (2005) UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley. ISBN 0321321278

5. Ben Schneiderman (1997). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley; 3rd Edition. ISBN:0201694972

6. Gary Cornelius, John J. Chelsom (2005). Making the right constraints for usable and accessible user interfaces. Proceedings of XML 2005; IDE Alliance.

References

1. Larry L. Constantine, Lucy A. D. Lockwood (2002). Usage-Centered Engineering for Web Applications. IEEE Software (SOFTWARE) 19(2):42-50

2. Connecting for Health, OCCO. Clinical Safety. http://www.connectingforhealth.nhs.uk/engagement/clinical/occo/safety