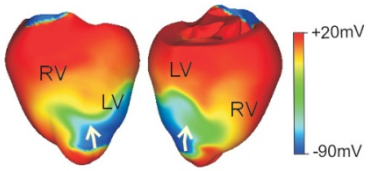
Towards a methodology for software preservation

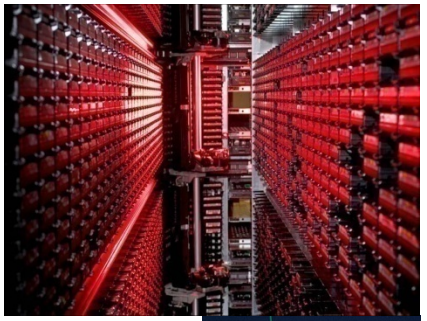
Esther Conway, Brian Matthews, Arif Shaon, Juan Bicarregui, Catherine Jones, Jim Woodcock (Univ of York)

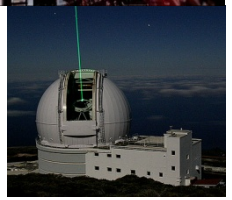
12/05/2009

Science and Technology Facilities Council



- Provide large-scale scientific facilities for UK Science
  - particularly in physics and astronomy
- E-Science Centre – at RAL and DL
  - Provides advanced IT development and services to the STFC Science Programme
  - Strong interest in Digital Curation of our science data
  - Keep the results alive and available
  - R&D Programme: DCC, CASPAR





STFC and Digital Curation

- STFC E-Science Centre interest in the preservation of its science outputs
  - Publications – library systems
  - Data – output from facilities, Petabyte Data Store, Data Centres
  - Keep the results alive and available
- R&D Programme in Digital Curation
  - Partner in the UK Digital Curation Centre
  - Coordinator of the EC Project CASPAR
  - VSR, SCARP, Parse-Insight, ….
  - Case studies in our own data
  - Roll-out to facilities

Work in Preservation of Software

- JISC funded work: Tools & Guidelines for the preservation of software as a research output
  - Used the JISC funded: Significant Properties of Software Report
- Software very large topic
  - Diversity in: *application of software* and *software architecture* and *scale of software* and *provenance* and *user interaction*
- Project needed to limit scope
  - Scientific and mathematical software
  - Limited commercial consideration
  - Limit consideration of user interaction
- Finding information
  - Literature
  - Talking to developers of products and software repositories
- Developing a framework for software preservation properties.

Software Preservation

- What is software preservation?
  - Storing a copy of a software product”
  - Enabling its retrieval in the future
  - Enabling its reconstruction in the future
  - Enabling its execution in the future

Not what most software developers and maintainers do.

Why Preserve Software ?

- Museums and archives:
  - Either supporting Hardware
    - E.g. Bletchley Park, Science Museum,
  - Or in its own right
    - Chilton Computing, Multics History Project
- Preserving the work
  - E.g. research work in Computing Science
  - Reproducible
- Preserving the Data
  - Preserving the software is necessary to preserve other data
  - Keep the data live and reusable
  - Prime motivation for STFC
- Handling Legacy
  - Specialised code from the past which still needs to be used
  - Usually seen as a problem!

Preservation Approaches

- Adequacy: How do we know we have captured enough?
  - Depends crucially on *Preservation Approach*

- **Technical Preservation. (techno-centric)**
  - Maintain the original software (binary), within the *original* operating environment.
  - Sometimes maintain the hardware as well
- **Emulation (data-centric).**
  - Re-creating the original operating environment by programming future platforms and operating systems to emulate the original environment,
  - so that software can be preserved in binary and run "as is".
  - E.g. British Library
- **Migration (process-centric).**
  - Transferring digital information to new platforms before the earlier one becomes obsolete.
  - Updating the software code to apply to a new software environment.
  - Reconfiguration and recompilation – “Porting”
  - An extreme version of migration may involve rewriting the original code from the specification.

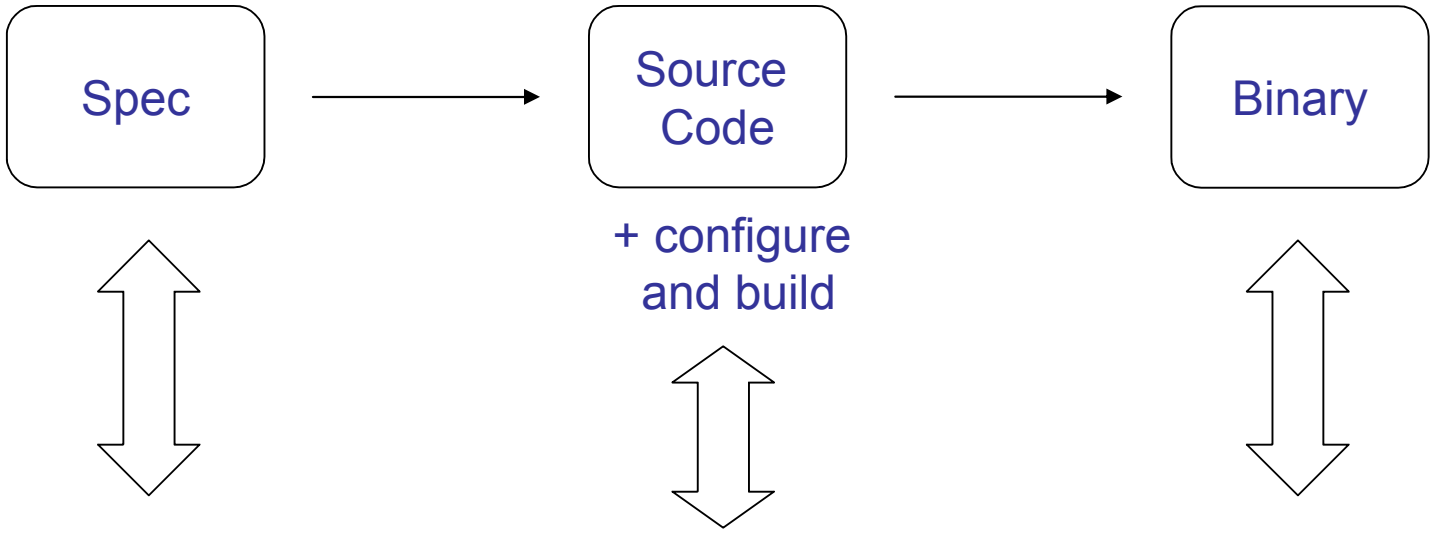
- Different preservation approaches required different significant properties
  - Use a notion of *Performance to assess adequacy*
  - *Test case suites as tests of adequacy*

Conceptual Framework

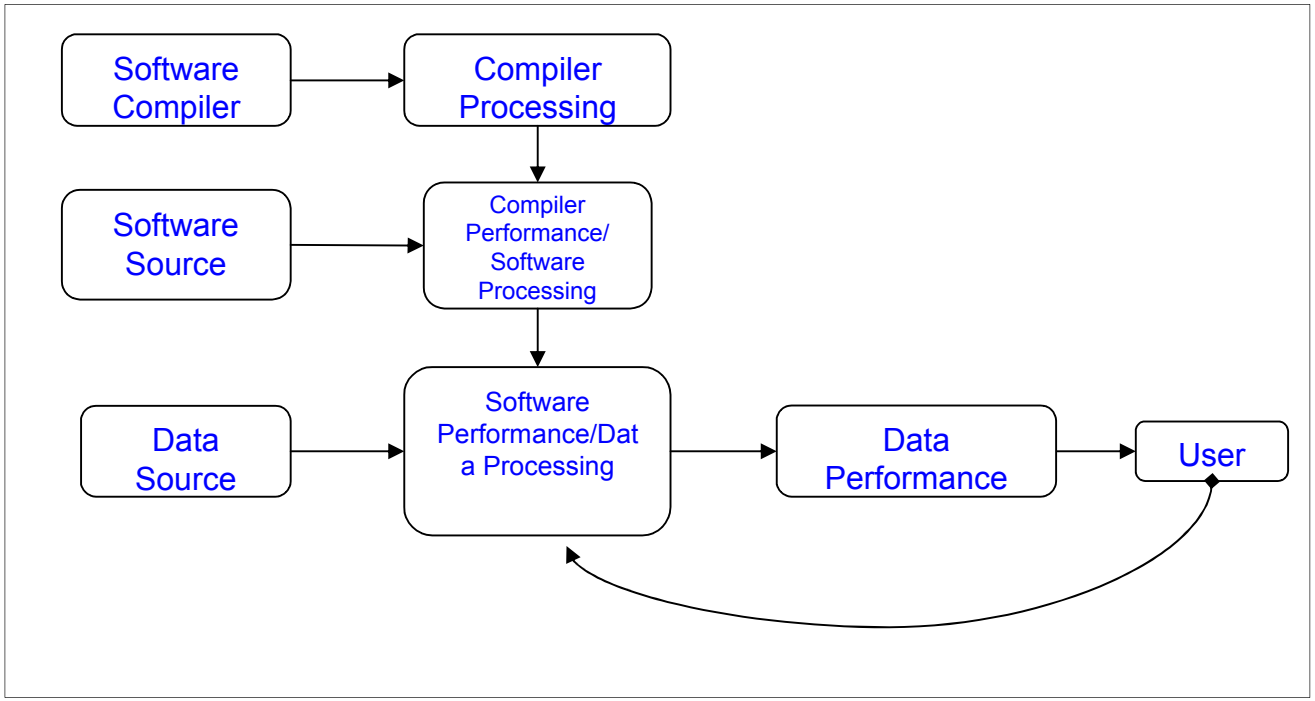
Three aspects to the framework:

- A Performance Model for software
  - Determine what it means to preserve s/w
  - Retrieve – Reconstruct – Replay
  - Adequacy of performance of s/w
- Model for describing s/w artefacts
  - As complex digital objects.
  - Versions and variants
- Properties for preservation
  - For retrieve, reconstruct, replay

Preservation Approach and Software Process



Performance Model for Software



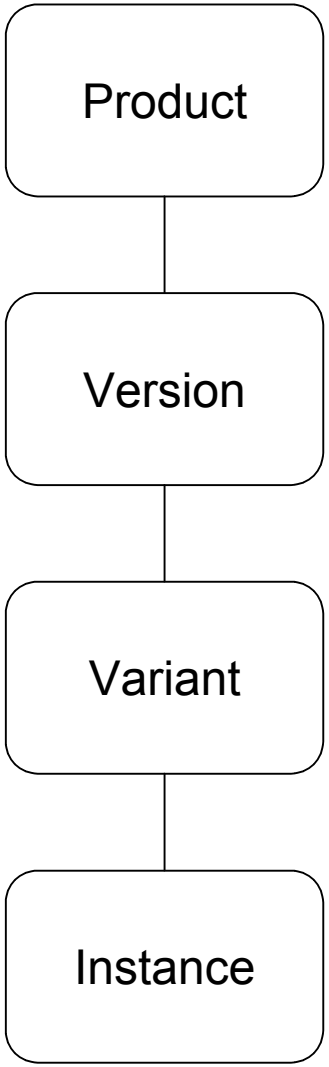
- Testing data performance to judge ***adequacy*** of the software performance.
- Important to maintain software test suite to assess preservation of significant properties of the software.

Adequacy

Adequacy.

A software package (or indeed any digital object) can be said to perform adequately relative to a particular set of features ("significant properties"), if in a particular performance (that is after it has been subjected to a reconstruction and replay process) it preserves those significant properties to an acceptable tolerance.

A Framework for Software



Provide a general model of software digital objects
Relate each concept in the model with a set of significant properties

For different preservation approach, we need different significant properties to achieve a desired level of performance.

- Product
  - The whole software object under consideration
  - Could be single library module, or very large system (e.g. Linux)
  - Comes under one “authority” (legal control)
  - Defines “gross functionality”
- Version
  - Releases of the system
  - Characterises by changes in detailed functionality
- Variant
  - Versions for a particular platform
  - Characterised by operating system and environment
- Instance
  - A particular instance of a particular variant at a particular location
  - Ownership
  - An individual licence
  - Fixed to particular MAC or IP address, URLs etc.

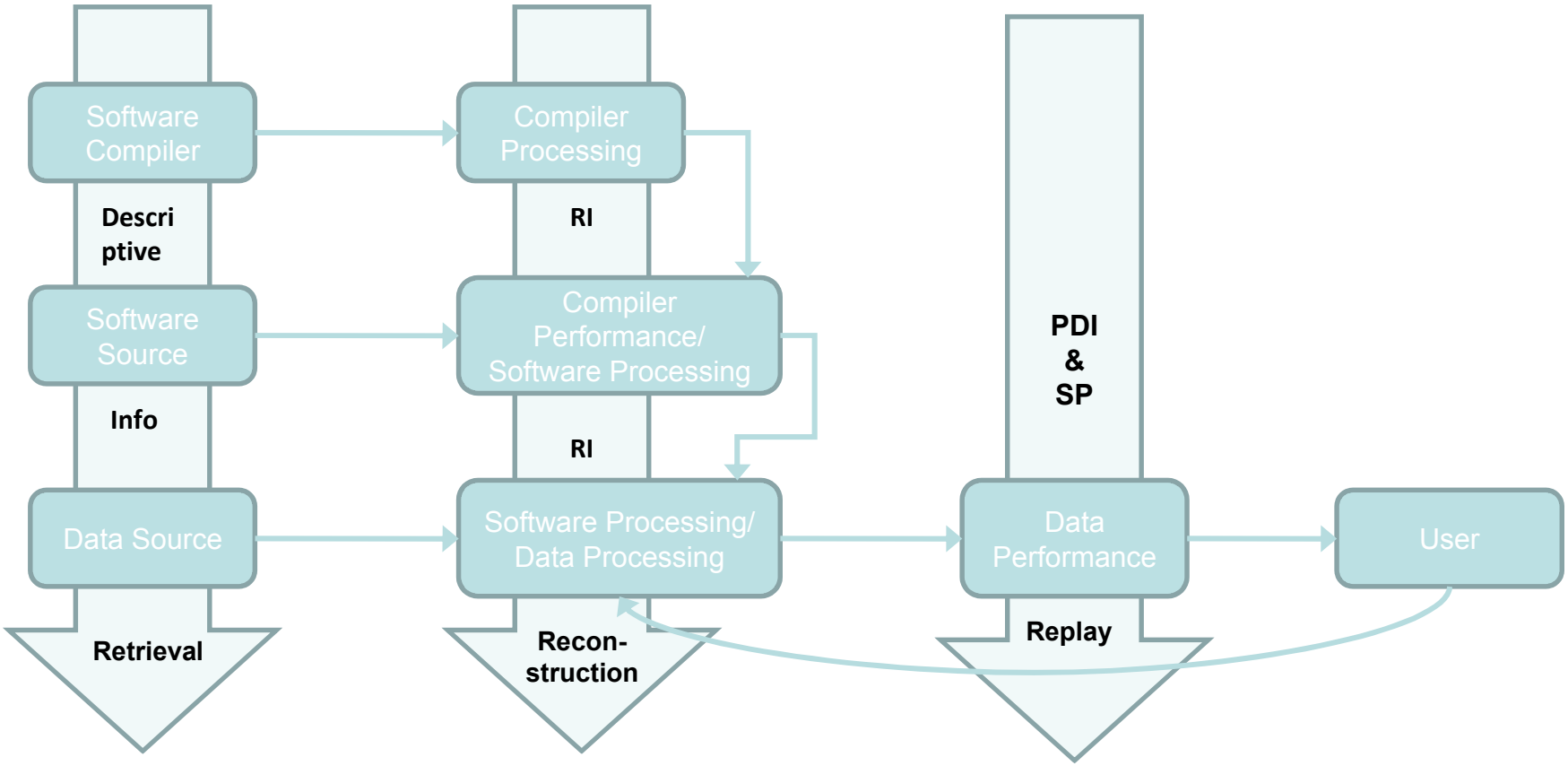
Preservation Properties of Software

- What to attributes do we need to take into account?
  - Functionality
    - what it does and what data it depends on
  - Environment
    - platform, operating system, programming language
    - versions
  - Dependencies
    - Compilation dependency graph
    - Standard libraries
    - Other software products
    - Specialised hardware
  - Software is a Composite digital object
    - Collection of modules
    - Specifications, Configuration scripts, test suites, documentation
  - Architecture
    - Client/server, storage system, input / output
  - User interaction
    - Command line, User Interface
    - User model

Clearly Software is highly complex with a lot of factors which need to be considered

we need a framework to organise and express software.

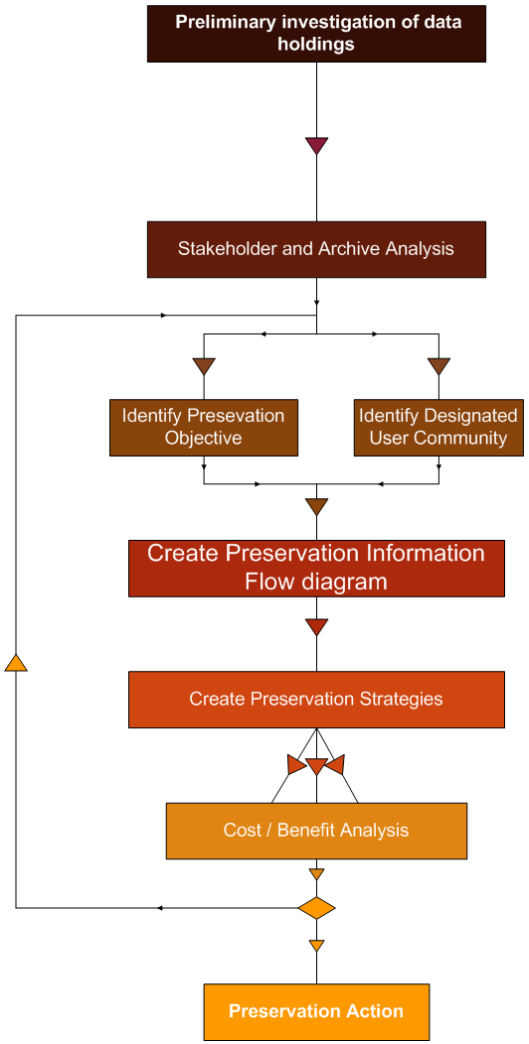
Relationship to the OAIS model

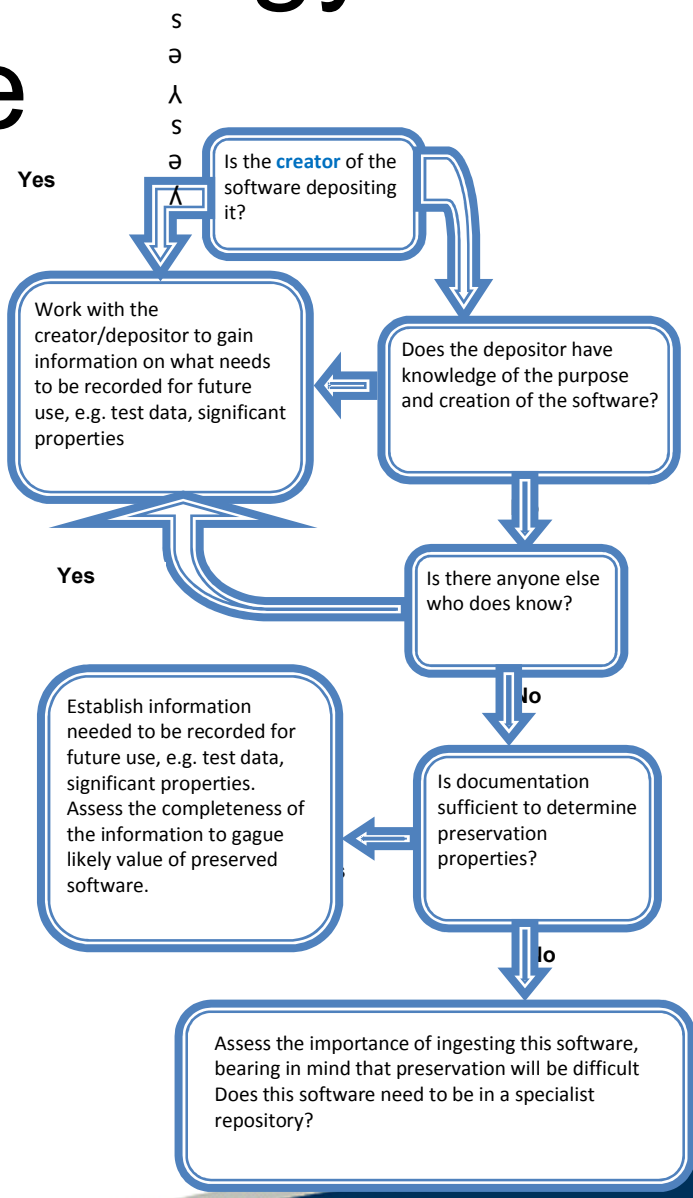


Open Archival Information System (OAIS) – ISO standard for the preservation of digital object.

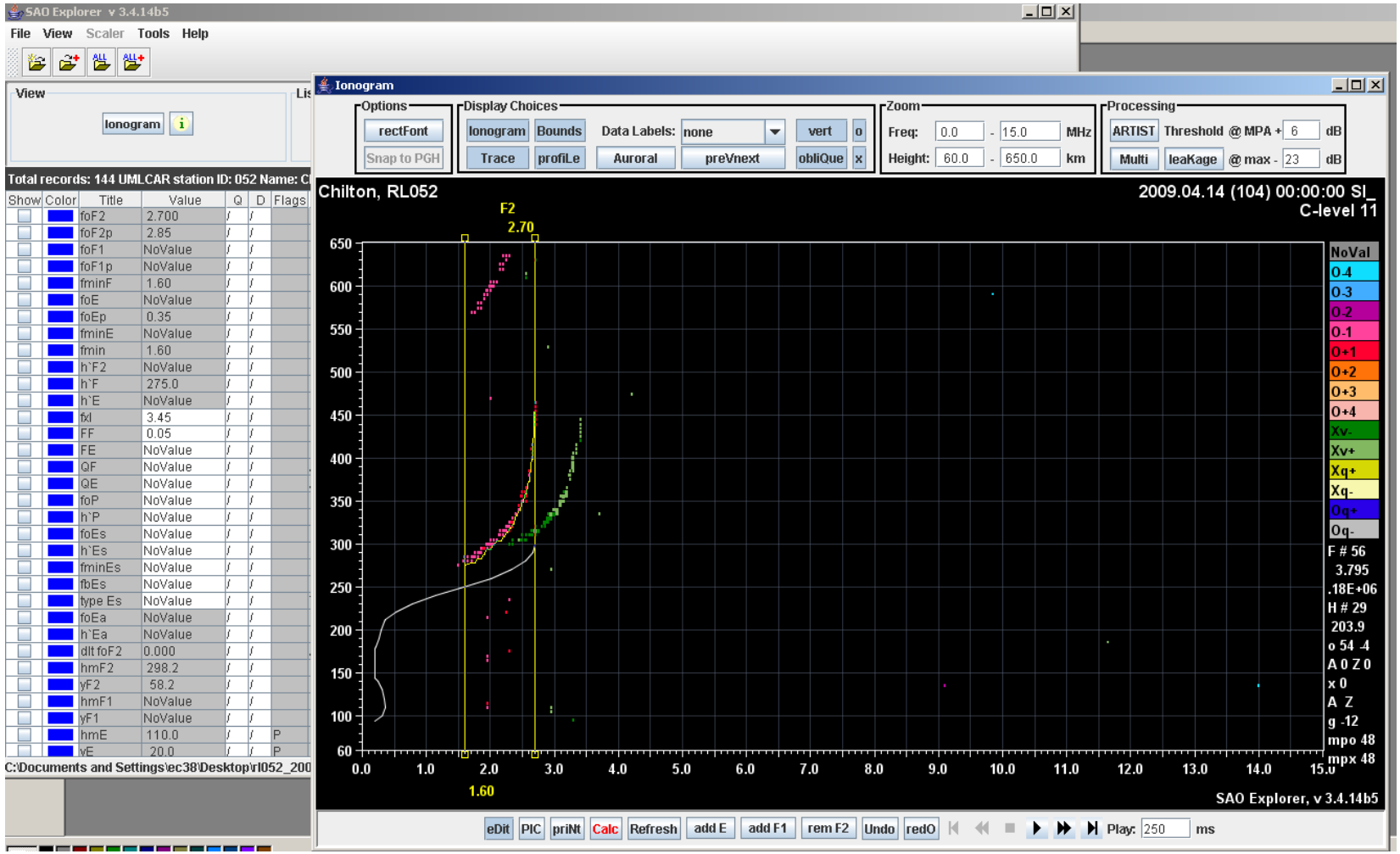
Software preservation properties are related to concepts in OAIS.

Preservation methodology and software

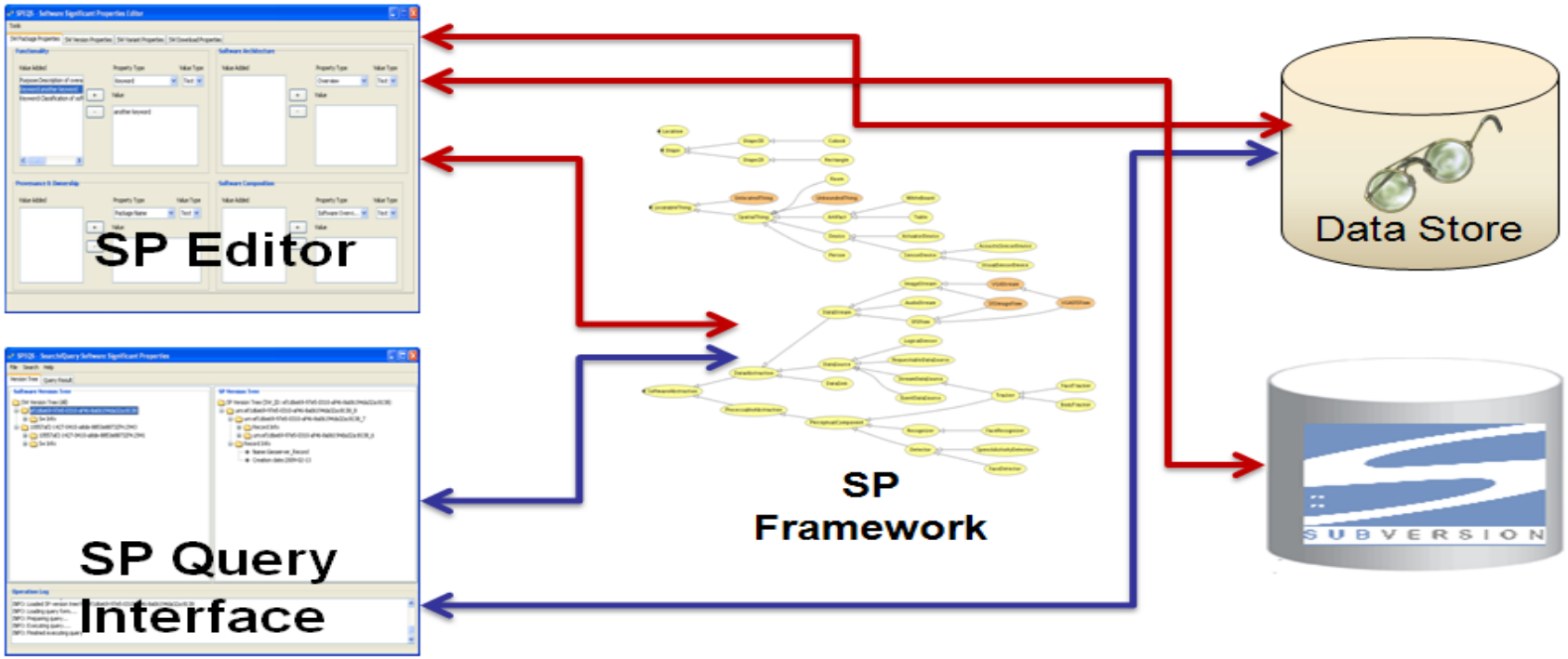




Case study



Significant Properties Editing and Querying for Software (SPEQS)



- Java-based Eclipse plug-in
- Enables capturing software preservation properties during its development
- Demonstrates the concept of preservation tools that could be integrated within existing software development systems

Summary

- Exploration of the s/w preservation space
- Defined reasons, audience, some basic concepts
- Defined a framework which enables s/w to be included in OAIS preservation framework
- Fits in a OAIS compatible preservation methodology
- Validated in some practical scenarios

Questions?

http://sigsoft.dcc.rl.ac.uk/twiki/bin/view

http://www.e-science.stfc.ac.uk/projects/information/software.htm