

CTS2 Implementation Phast (So Far)

PHAST

CTS2 Implementation Phast

- Phast is a terminology provider and distributor in hospital pharmacy medicine terminology (CIOsp) and Medical Devices Terminology - CIOdm using proprietary means.
- Needed a better way to handle terminologies (Text files vs. Web Services).
- Wanted a more robust and internationally-aligned mechanism of distribution → started implementing CTS2 .
- Implementation as close as possible of the SFM
 - See functional requirements matrix
 - Some need for improvement were detected in SFM (Concept Domain Hierarchy in the Conceptual Model, alignment of the SFM with Core Principles)
 - STS considers **binding** as the center of a terminology server (expected services in the solution deployment in France – CDA Implementation Guides - Volets de contenu,).

CTS2 Implementation Phast

- Implementation of HL7 CTS2 SFM based on HL7 v3 RIM and HL7 Data Types R2 (HL7-ISO Data Types)
- Product named Standard Terminology Services (STS) - beta-test.
- STS workshops for hands-on implementation feedback.
- The technical implementation specifications are in French and are not formally written in an implementation guide (*fiches techniques*). A great amount of editorial work is ahead in order to collect all the information from different documents, translate it, and publish it.
- Information and downloadable components:
[http://wiki.phast.fr/index.php?title=Standard Terminology Service - STS](http://wiki.phast.fr/index.php?title=Standard_Terminology_Service_-_STS)

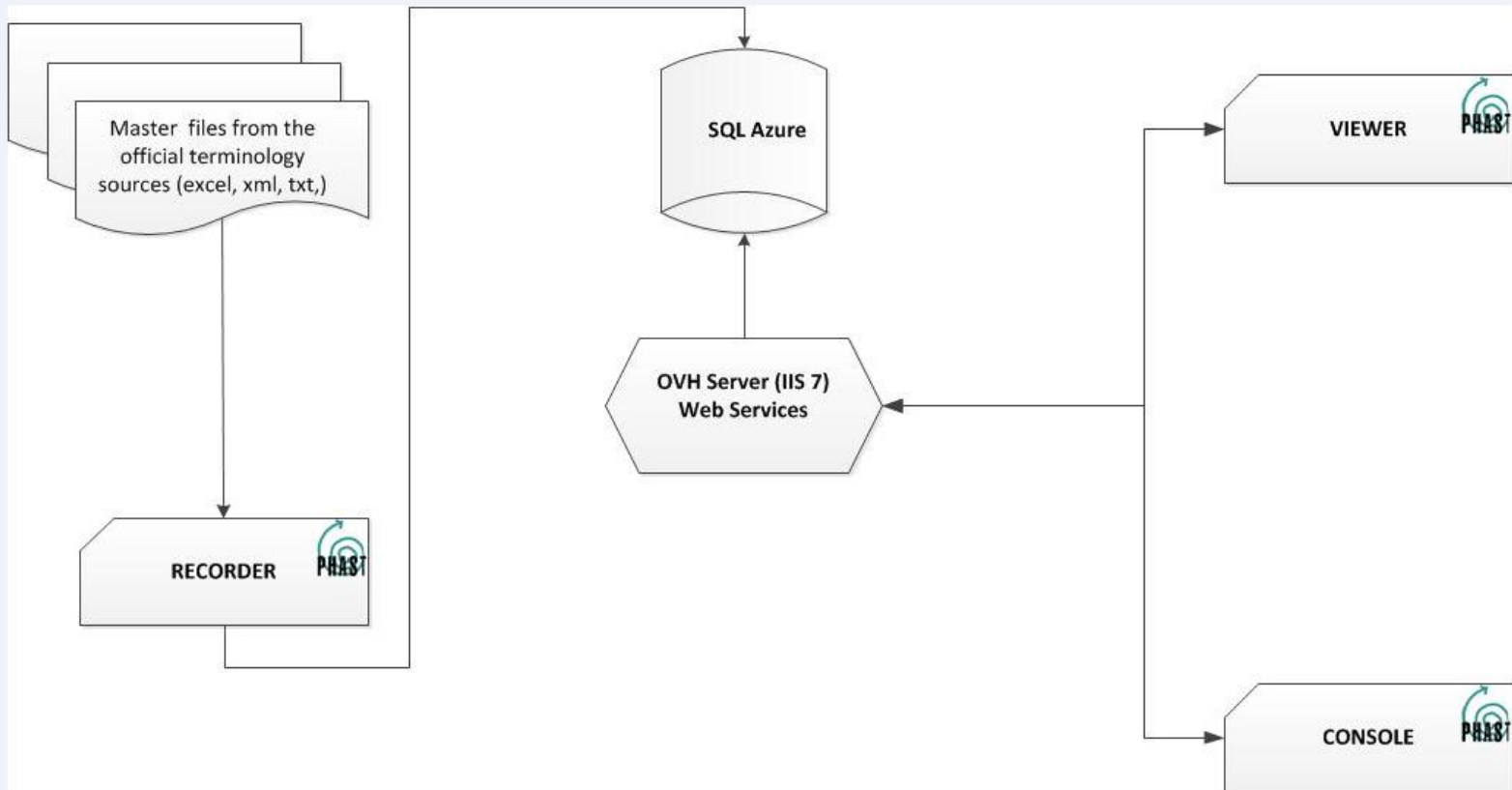
CTS2 Implementation Phast

- Only the “**read-only**” functions, namely the CTS2 Query Profile and all the operations associated with it (22 operations) are implemented and functional, although the WSDL was written for all the functions in the CTS2 so that the implementers can have access to them.
- The CTS2 Terminology Administration Profile (11 operations) and the CTS2 Terminology Authoring Profile (23 operations) are not yet active although the WSDL is written for them. For the moment, from a practical point of view, the “write” functions are not needed immediately by the French hospitals since the data is administered and integrated directly into the database by the Recorder.

CTS2 Implementation Phast

- Wanted to check progress of SFM CTS2 - expired in Oct 2011, wanted to pick up the thread.
- Phast became an OMG member in March 2011
- Presented implementation efforts for the first time in Vocabulary WGM in Orlando, April 2011
- Discussion with Mayo clinic with regards to harmonization efforts
- Became member of the OMG FTF in December 2011
- Proposal to work with Mayo in San Diego – September 2011 (“*the three things*”):
 - Connecting Phast viewer to Mayo terminology server
 - Implementing two read-only services in STS as specified in OMG CTS Implementation guidelines.
 - Working together in contributing to the normative version CTS2.
- Phast would like to contribute to the normative version CTS2 and the OMG specifications in order to improve CTS2 in general.

STS Architecture



Web Services

- The CTS2 operations are implemented as functions of the STS Web service.
- All the functions are defined, and for each, the corresponding Web service operation can be invoked with the appropriate input data.
- All the CTS2 operations are present in the list of functions on the web site and on the Forms in the Console (56 operations).
- The website where the Web service is available is:
http://extension.phast.fr/RLIO_WS/RLIO.asmx
- Only the active functions that have been implemented will be able to return data. There are only 22 active functions.
- The purpose of having both active and inactive Web service functions listed is so that the programmers needing them can see them and can obtain the WSDL
- The Console will demonstrate if a function is active or not (if not active, nothing will happen).

The Database

- The STS database is based on HL7 CTS2 SFM and the HL7 RIM layered on top of the ISO data types
- Advantages to this approach
 - The concepts corresponding to real world objects such as medication or a medical device can have a structured description, allowing for links to additional data saved under the Common Product Model format whenever the need arises. (a medication has attributes such as active ingredient, packaging, pharmaceutical form, etc. and each attribute has different code systems)
 - The CTS2 functional specifications allow for bindings and relationships between
 - ❑ Value set
 - ❑ Concept Domains
 - ❑ Usage context
 - The templates are implemented using their usage context; however one may need additional information that is obtained via the link between a template and a particular RIM class.
- Azure SQL database

Recorder

- Executable program proprietary to Phast.
- Can write directly into the database since the “**write**” functions of STS are not active yet.
- Can receive information either manually or provided by the Feeder, will be able to receive information when the “write” functions will be activated as well.
- Works by periodically checking its working directory for new files to be integrated in the database.
 - Each file may represent either a whole terminology or simply a value set
 - Each type of file can have a different method for integrating in the database
- The Recorder also archives all the processed material, depending on its source (if the material is created by a user such as translation of display names vs. material produced by official terminology providers).

Console

- The Console is a program used as a test tool.
- Each **console form** is a function described in the Web service allowing the capturing input data for the Web service, and returning data so that the tester can verify that the functions supported by the Web service are working properly.
- The Console is a program which is a testing tool for the Web Services and giving access to the the data present in the Azure database.
- You can install the Console on any PC following the instructions on the Phast wiki:
http://wiki.phast.fr/index.php?title=Standard_Terminology_Service_-_STS#How_to_install_the_Console_on_your_machine.

Viewer

- Lets a user see the contents of STS database.
- The Viewer is the only part of the STS architecture which is intended for the end user in order to establish trust.
- An advanced user of STS will probably only use it when some manual verification may be needed, for example if a user of an STS client comes across an unexpected code s/he can use the Viewer to check the actual content of the data base.
- http://extension.phast.fr/CTS2/RLIO_CTS2_Main.aspx

Artifacts – Functional Requirements Comparison Matrix

- First version of the matrix posted.
- Name of the operations in STS = Name of the operations in CTS2
- Some operations added – business scenarios to be added?
- Work in progress
 - Specifications
 - Connecting the viewer to the Mayo server (links provided to Nicolas).