

# Release Notes for LiU Archetype Editor v0.5

## Updates

New versions of the editor and more information can be found at either of these pages:

- <http://www.imt.liu.se/mi/ehr>
- [http://svn.openehr.org/liu\\_knowledge\\_tools/TRUNK/project\\_page.htm](http://svn.openehr.org/liu_knowledge_tools/TRUNK/project_page.htm)

## Installation

1. Extract the ZIP-file in which the LiU Archetype Editor is packaged.
2. The directory you chose to extract to will contain a directory with archetypes ('archetypes'), a Java Executable Archive file (JAR-file) and a batch-file that can be used to start up the program if double-clicking the JAR-file does not work.

To run the editor on your system you must have Java Runtime Environment 5.0 (1.5.0) or newer. Java Runtime Environment (JRE) 5.0 can be downloaded here: <http://java.sun.com/j2se/1.5.0/download.jsp>. If you have a fully working installation of Java Runtime Environment 5.0 or later the application will start when you double-click the JAR-file. If that does not work, you can run the program from the system console with the command 'java -jar filename.jar' or try double-clicking the batch-file (.bat) if you are using Windows.

## General usage

- All the archetypes that exist in the 'archetypes' directory and which are not inside a folder stating otherwise can be opened with the editor. Besides these archetypes, there might exist some at the [openEHR Clinical Knowledge Repository](http://svn.openehr.org/knowledge/project_page.htm) ([http://svn.openehr.org/knowledge/project\\_page.htm](http://svn.openehr.org/knowledge/project_page.htm)) that are supported.
- Archetype terms can be looked up and bound to UMLS if you have an account at <http://www.nlm.nih.gov/research/umls> and you have granted your IP address in your settings for the UMLS Knowledge Source Server <http://umlsks.nlm.nih.gov>.
- If you have granted your IP address but have problems connecting, try changing to the standard UMLS Knowledge Source Server: `//umlsks.nlm.nih.gov/KSSRetriever`.
- The different views in the editor can be cycled through with Alt + Left/Right. Other shortcut keys are shown in the menus.
- Please note that everything related to the 'Experimental' menu is under development and not fully working. A new version of the editor that adds more functionality to the MoST and SNOMED CT terminology services will probably be released later.

## Changes since v0.4

- Disabled the TermViz browser which will be included in a later version of the editor.
- Disabled the MoST-wizard since there is no public server to connect to yet.
- The only way to make use of the binding suggestions from the MoST system is to load XML-files created by the MoST system.
- Log4j's GUI log viewer application LogFactor5 can now be opened in the view menu. You can view the log files which are created in the root directory of the editor.
- Added archetype package URI field in description view.

- Added translations and revision history tabs in the description view. The user-interface in the translations tab allows translators to input information about the translation of the archetype but limitations in openEHR software makes this part useless until software upgrades can be made. Version controlled information about archetypes can be viewed in the revision history tab.
- Assumed values supported for more AOM and AP classes now.
- Switched the interface view to a Swing-based simulation of an entry form for the corresponding archetype. Note that this view is in development.
- Some usability improvements, code and performance improvements and bug fixes.

## openEHR software used

The LiU Archetype Editor uses third-party software provided by openEHR. This software is often referred to as the Java reference implementation of openEHR (read more about it here: [http://svn.openehr.org/ref\\_impl\\_java/TRUNK/project\\_page.htm](http://svn.openehr.org/ref_impl_java/TRUNK/project_page.htm)). The current version of the software that the editor uses is 1.0.1-RC1 and primarily the editor uses the following components:

- **openehr-aom** Java implementation of the Archetype Object Model (AOM)
- **openehr-ap** Java implementation of the openEHR Archetype Profile (AP)
- **adl-serializer** ADL serialization from AOM, AP objects
- **adl-parser** Archetypes in ADL format to AOM, AP objects

## Limitations

- Archetypes cannot be version controlled from within the archetype editor as of yet, but plans to add support for an archetype service have been discussed.
- There are limitations regarding support for some of the latest archetypes available (<http://svn.openehr.org/knowledge/archetypes/dev/index.html>). These limitations primarily exist because of deficiencies in third-party software.
- Archetype slots are not supported, which means the editor does not fully support composition, section, action and instruction archetypes due to third-party software limitations.
- The editor has a field for archetype package URI, but it is not yet supported by third-party software.
- The interface view is far from fully developed and the interface generated from some archetypes or parts of archetypes may be blank, incomplete and incomprehensive.
- Demographic archetypes are not supported by the editor.

## Future releases

Below follows a priority list of some of the major things targeted for future implementation.

1. Include the TermViz terminology visualization tool in the editor which allows browsing and making terminology bindings from SNOMED CT in a convenient way.
2. Make the editor validate archetypes against the openEHR specifications, XML-schema or the reference model classes of the Java reference implementation. The current validation is hard-coded and will throw away unknown elements.
3. Translate the editor into other languages (use Java Internationalisation - I18N).
4. Implement a connection to an archetype service which allows archetypes to be version controlled.
5. Implement a choice to either set occurrences of archetype slots (use nodes) or use the referenced node's occurrences.

6. Switch Log4j's logging application LogFactor5 to the newer application Chainsaw and make the editor alert the user when something has been written to the log.
7. Add support for Demographic archetypes.

## **Known bugs**

- If some text in the description view is very long, e.g. concept, then the form may grow to be wider than the application frame. This is ugly, but not critical.

## **Acknowledgements**

This work was performed in the framework of the EU-funded Network of Excellence entitled Semantic Interoperability and Data Mining in Medicine (<http://www.semanticmining.org>).