ProposalPluginApi

Why Plugins

- Seperate Functionality from core
- Create a 3rd party plugin ecosystem

Technical foundation

- can likely be implemented using UsecaseFramework
- fallback:/ pseudo protocol
- XPatch at runtime (???)

Problems

- Extension versus inheritance

To be clarified

- What does a plugin consist of
- How should plugins be handled in Lenya
- Do plugins apply only to authoring or to live as well?
- Are plugins the same as blocks in Cocoon?

Project decisions

- Should plugins be a compile time or a runtime feature?

Prototyping Plan

(by J. Wolfgang and Torsten (without "h"))

- Define a PluginExperimentPublication in trunk
- Implement a publication specific generic (i.e. editor independent) EditDocument usecase which will use a EditPlugin through a well defined interface (contract) to handle the actual editor (see below)
- Implement a publication specific LenyaCMSMenuGenerator Avalon component which
  - will read an XML file with structure similar to the **result** of today's Default publication generic.xsl.
  - will replace placeholders in the XML menu skeleton with menu options supplied by plugins: for example EditorPlugins
  - the rest of the menu.xmap pipeline will remain unchanged for now
- Develop a publication specific BXE EditorPlugin as an example to proof the concept