

Design - Module Framework and Registry

Overview

DSpace needs a clear definition of what constitutes a "DSpace module", so that third-parties can create, maintain and distribute their own "modules" as add-ons to DSpace. These third-party modules also need a way to be easily distributed and advertised via some form of "module registry". The registry would allow users to find and select third-party modules to install into their own DSpace, as they see fit. DSpace would also need mechanisms to install, upgrade or uninstall individual modules. A "DSpace module" would also need a specific version number, and a dependency definition (e.g. "requires DSpace 6.0 or above").

Use Cases / Needs

- Define a family of specifications/interfaces for functional extensions to 'core' DSpace (working title: 'modules'), and refactor existing bundled code to conform to new model (if appropriate/cost-effective)
- Provide infrastructure/tools for a module registry, where users can discover, and install modular extensions. Likely include both modules maintained by committers and community contributions.
- End goal would be to find a way to modularize components of DSpace so that institutions can mix & match modules to meet their local needs.
 - Certain modules would come "out-of-the-box" (the ones most folks will want to use by default & that make DSpace easy to get started with). Some of these may be mandatory (if DSpace really needs them to function well), while others are just "recommended"
 - Additional modules would be available via the module registry.

Planning Phase - Prototyping and Selecting a Framework

Timeline is TBD based on establishment of a Module Framework Team

- Establish a dedicated "Module Framework" technical team to prototype potential platforms. This team will consist of volunteer or "donated" development resources
- Define a list of possible "module framework" options to analyze (paying special attention to similar functionality in other Java-based projects)
- Prototype at least one "module framework", along with a basic ("Hello World" like) sample module for testing / verification
 - If multiple feasible options exist, more than one framework should be prototyped, implementing the same sample module.

Development Phase

details needed