WoodyTODO

The Woody TODO list

Implementation-level things

- evaluate other, more standard, expression engines such as Jexl and JXPath and compare with the currently used xreporter-expression package.
- Add an setValidationError or addValidationError method to the Widget interface (partly done: the field widget already has this)
- variable fields: fields which are not editable by the user (DONE: these are the output widgets)
- calculated fields
- default values for fields
- checking validity of id's (should not contain spaces or dots, or characters not allowed in request parameters)
- Including one form in another form (subforms)
- It should be possible to set SelectionLists on widget-instance level (currently they are read once on widget-definition level). (DONE)
- think about the design of the various WoodyFileFormats
- make an utility that checks that all (required) fields defined in the form definition are present in the form layout template
- (low) allow src attribute on selection-list to be a class that implements a certain interface, for higher performance than executing a pipeline and interpreting its XML output. (DONE: selection list implementations are pluggable now)
- as an alternative to the former: add `model` attribute to `selection-list`, referring to a collection stored in the object model by the flowscript. (DONE: flow-jxpath selection list implementation)
- validation widget: a widget that has no value of its own, but just serves to perform validation on or between other widgets (for validation rules that don't belong to a specific widget).
- it should be possible to run over the widget tree without knowing what widgets are in there (getChildren on ContainerWidget interface or something) (DONE)
- for fields with a selection list, it should be checked that the submitted value is effectively in the selection list (to avoid problems with people playing with request parameters). Since this could potentially be expensive (if the selection list is dynamically generated), maybe do this as a seperate validation rule.

- adapt the AggregateField so it can aggregate the other way around as well `(see [mail-list-proposal|http://marc.theaimsgroup.com/?l=xml-cocoon-dev&m=105712568410208&w=2]) -- ([mpo|MarcPortier])`

- think about transfering (part of) the woody form-model information over to the client as a javascript model that can be interpreted by a set of client-side javascript functions inside the browser

```
-- ([mpo|MarcPortier])
```

- make the 'binding' isolate the woody model from the business model --> fieldbinding should perform a clone()

```
-- ([mpo|MarcPortier])
```

- cache compiled bindings
- create an "enum" base type and an "enum" convertor for types that implement the TypesafeEnumPattern – (UgoCei)
- ...