

# magnolia

*Simple is beautiful*

## **Magnolia GUI - Concept Presentation (first edition)**

# Overview & Status

- concept covers ideas & „discussion input“ for the following topics
  - Data Model (-)
  - Metaphors & Basic Concepts (++)
  - Features (++)
  - Structure (+)
  - Interaction (0)
  - Design (-)
- status
  - see <http://wiki.magnolia.info/display/DEV/Genuine+Concept>
  - many ideas and „TODO“s, most likely to much for first sprint
  - meant as an overview („big picture“) or status quo

# Data Model

- model („picture“) of the application that exists in the user’s mind
- important for the user to understand, otherwise he can not work efficiently
- how this model looks like (for the user) depends on how the user interface communicates it (this is called „mapping“)
- Goal: ensure that the mapping of the underlying application model is recognizable through the UI
  - precondition is that the semantics are well defined
  - does the GUI communicate those semantics correctly?
- examples:
  - a Magnolia installation (normally) consists of one author and multiple public instances
  - a paragraph is associated with a dialog

# Metaphors & Basic Concepts

- several concepts/metaphors of the current Magnolia administration interface have been proofed successfully
  - Admin Central metaphor
  - actions and the function bar (toolbar)
  - templates, paragraphs and dialogs
  - hierarchical editing vs. editing in preview mode
  - URIs and virtual URIs
  - [...]
- however, they still can and maybe should be improved for mostly usability reasons („to be discussed“)

# Metaphors & Basic Concepts

## Admin Central

- improvement: a single Admin Central for multiple instances
  - abstracts from deployment infrastructure
  - single entry point to Magnolia
  - bulk changes over multiple instances
  - completely separate application life-cycle (updates)
  - requires webservice approach
- new feature: dashboard
  - direct entry points for the user
  - „information at a glance“
  - customizable depending on user preferences and/or role
  - rich features (excerpt)
    - last edited pages, inbox, server health & statistics, help

# Metaphors & Basic Concepts

## **Actions and Function Bar**

- enhancements to ensure consistency; some remarks
  - actions available in context menu are a subset of those in the function bar
  - actions are accessible through keyboard shortcuts
  - actions can be used for bulk changes
  - actions can be found at single place: the function bar (toolbar)
  - toolbar has more responsibilities (e.g. creating contents in website tree)
  - toolbar is context sensitive and context aware (e.g. not often used features are hidden)
  - toolbar is also available in preview editing mode

# Metaphors & Basic Concepts

## Templates, Paragraphs and Dialogs

- there are „inconsistencies“ or pitfalls
  - the data structure stored in the JCR depends on the templates (hard coded in cms:newBar)
  - each paragraph has to have a dialog, which is not always reasonable
  - configuring templates/dialogs/paragraphs is error-prone
- some ideas for improvements
  - provide input forms for creating/editing definitions (++)
  - revise model, especially (+)
    - introduce configurable „zones“ (e.g. „mainColumn“)
    - consider security configuration or custom logic for the mechanism of choosing a template/paragraph for a page/zone/paragraph
  - separate layout and style (-)

# Metaphors & Basic Concepts

## Hierarchical vs. Preview Editing

- problems with (current) preview editing mode
  - works with JSP templates only (taglib)
  - navigation/page structure not editable
  - complicated layout makes page editing difficult
  - depends on template, e.g. removing a `<contentNodeCollection>` makes content inaccessible
- solution: allow editing of contents in website tree
  - generic (template independent)
  - keeps/shows navigation context, page structure editable
  - faster for advanced users
  - allows searching/filtering
  - allows drag & drop between pages, DMS and website tree

# Metaphors & Basic Concepts

## URIs and virtual URIs

- several concepts in Magnolia use (virtual) URIs: website pages, I18n support, virtual URIs, URI-to-repository mappings, URI security, default URI, activation subscriptions, cache and more
- very powerful (if one knows where/how to configure things)
- however, not very user friendly
  - each mechanism configured at a different place („where?“)
  - hard to figure out which (virtual) URI rule applies („why?“)
- possible solution: more „responsibilities“ for the website tree
  - manages virtual URIs
  - manages URI-to-repository mappings
  - manages URI based I18n support
  - manages/indicates default URI
  - maybe even indicates activation subscriptions and cache

# Features

- some notes
  - not necessarily new features, some are revised a bit
  - most of them still have TODO markers
- by category
  - general: customizability (+), skinning support (0), keyboard support (+), drag & drop support (+)
    - editing: actions (+), bulk changes (+), toolbar (+), context menus (+), property bar (?), masked views/advanced view filters (-)
    - navigation: breadcrumb (+), location bar (0)
    - help: help page & contextual help (0), tool tips (+)
  - content management
    - pages:

# Features

## Category: General

- customizability (+), skinning support (0), keyboard support (+), drag & drop support (+)
- editing: actions (+), bulk changes (+), toolbar (+), context menus (+), property bar (?), masked views/advanced view filters (-)
- navigation: breadcrumb (+), location bar (0)
- help: help page & contextual help (0), tool tips (+)

# Features

## Category: Administration

- configuration dialogs (+)
- tools dialogs (+)
- user preferences (+)
- security: hierarchical groups (+), advanced security configuration (-)
- modules: module manager (?), module configuration dialogs (+)
- automatic publishing (-)

# Features

## Controls

- see Wiki for a complete list, this section covers most important new features
- action sources
  - keyboard shortcut support (+)
- editable value holders
  - validation (+), formatters (0), converters (0)
  - auto-completion/suggestion (0)
  - multi-value controls (+)
- structural components
  - customization (column order, sorting, width, etc.) (0)
  - (live-) filtering (+)
  - mouse-wheel support (+)
  - paging (+), partial loading (+)/"load in background" (-)

# Features

## Controls (continued)

- layout components (0)
- informational components
  - progress bar (+)
  - status bar/message panel (+)
  - tooltip (0)
- some notes:
  - those controls are consistently used in all dialogs
  - some features depend on the underlying library (e.g. ExtJS)

# Structure

- see poster
- some important remarks
  - configuration is not covered by a single tree anymore, but instead split over multiple screens
  - screens & navigational structure are organized after use case/semantics
    - e.g. setting up templates is different from configuring the mail server (in terms of change frequency or user role)
  - goal: hide unnecessary complexity of the configuration hierarchy
  - however, we should also keep the „classical tree view“ for advanced users

# Interaction

- goal: review important interaction scenarios to ensure consistency in the application's behavior
- work in progress
- managing content
  - creating pages/page structure, creating content
  - finding pages or content
  - editing pages/page structure, editing content, performing bulk changes
  - observing progress of action execution
  - content life-cycle: activating, reviewing and publishing
  - exporting & importing data
  - browsing versions

## Interaction (continued)

- managing security
  - creating a security subject (user, group, role)
  - checking a users effective permissions
  - refactoring an existing security configuration
- (to be continued)

# Next Steps

- discuss features
- UI prototypes: proof-of-concept for scribbles
- design