Using Knowledge Networks to support Innovation

Henno Gous

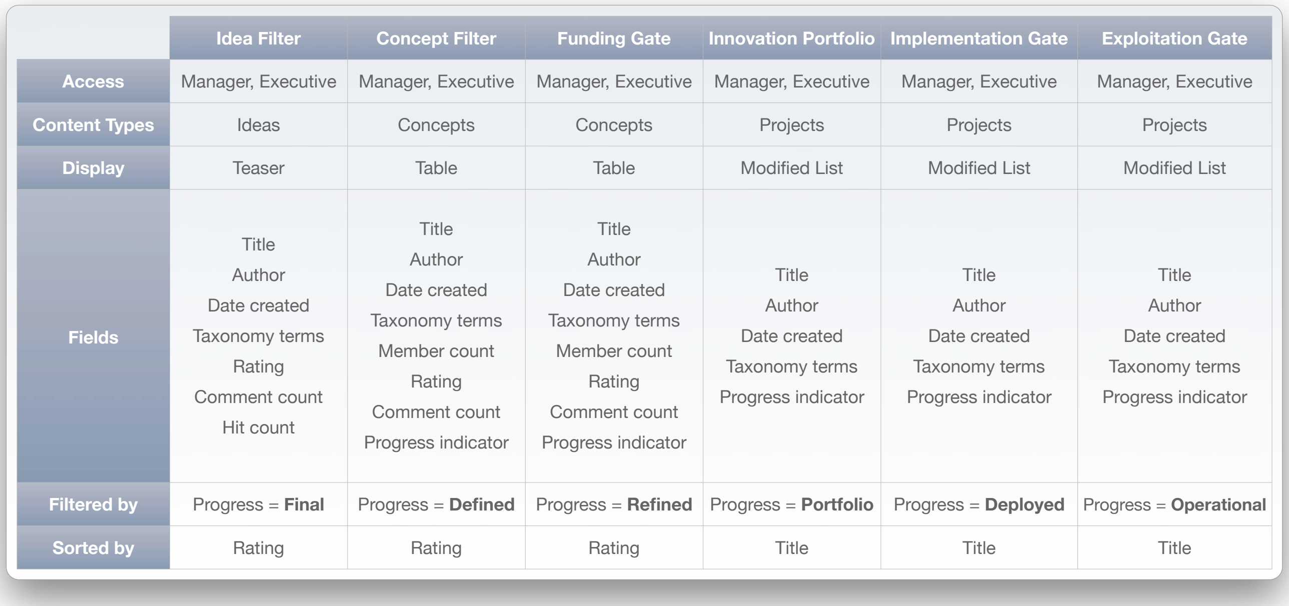
Thesis presented in partial fulfilment of the requirements for the degree of

Master of Engineering Science

at Stellenbosch University

Study Leader: C.S.L. Schutte

March 2009



	Idea Filter	Concept Filter	Funding Gate	Innovation Portfolio	Implementation Gate	Exploitation Gate
Access	Manager, Executive	Manager, Executive	Manager, Executive	Manager, Executive	Manager, Executive	Manager, Executive
Content Types	Ideas	Concepts	Concepts	Projects	Projects	Projects
Display	Teaser	Table	Table	Modified List	Modified List	Modified List
Fields	Title Author Date created Taxonomy terms Rating Comment count Hit count	Title Author Date created Taxonomy terms Member count Rating Comment count Progress indicator	Title Author Date created Taxonomy terms Member count Rating Comment count Progress indicator	Title Author Date created Taxonomy terms Progress indicator	Title Author Date created Taxonomy terms Progress indicator	Title Author Date created Taxonomy terms Progress indicator
Filtered by	Progress = **Final**	Progress = **Defined**	Progress = **Refined**	Progress = **Portfolio**	Progress = **Deployed**	Progress = **Operational**
Sorted by	Rating	Rating	Rating	Title	Title	Title

Figure 70 - Custom views of the network knowledge base may be used to provide necessary information to accompany filters and gates in the Innovation Life Cycle

By creating certain view outlines, user needs for data display is anticipated. It is however impossible to anticipate all user requirements, and therefore a search function is enabled with the core Search module. The way in which knowledge objects were seamlessly integrated with content earlier, now allows search results to reflect users, idea, concepts, blog posts, documents and forum to be included in search results for a single query. The SQL Search module expands this functionality by allowing users to specify taxonomy terms, content types, etc. that should or should not be included in the results, which equates to emulating predefined views on-the-fly.

5.5. Messaging

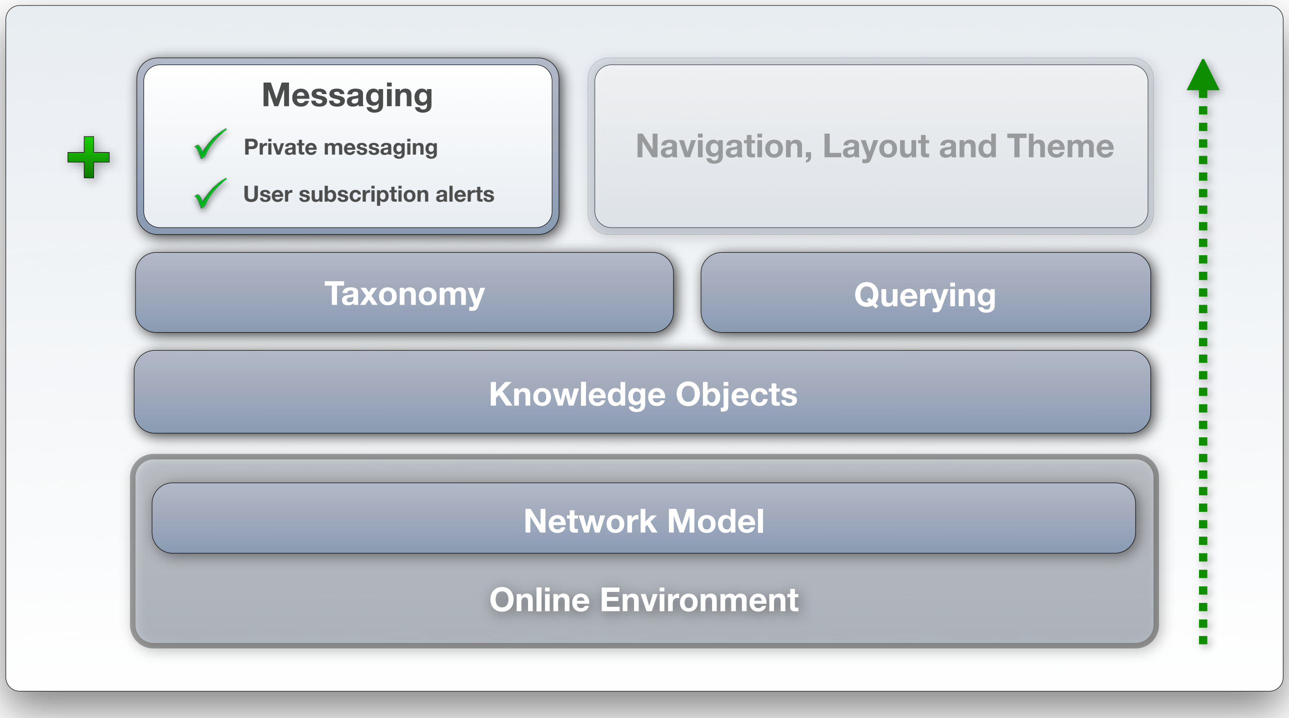


Figure 71 - Design layer 5: Messaging

Functionality Specification: Private messaging

Addressing requirement 5, and stated in Chapter 4.4.6.1 as:

Communication has been highlighted as an important requirement for the proposed platform solution, especially for enabling social networking on the platform and promoting the smooth execution of the Innovation Life Cycle. Private messaging and contact forms that send an e-mail to users should be included to provide an efficient communication system throughout the platform. These functions will compliment the support for discussion that is created by employing commenting and forums.

Functionality Specification: User subscription alerts

Addressing requirement 15, and stated in Chapter 4.4.6.2 as:

Users should be able to subscribe to alerts that are triggered by activity within the network. This could be a new comment in a certain thread, a new post being classified in a certain knowledge category, or a favorite user making a new contribution to the network.

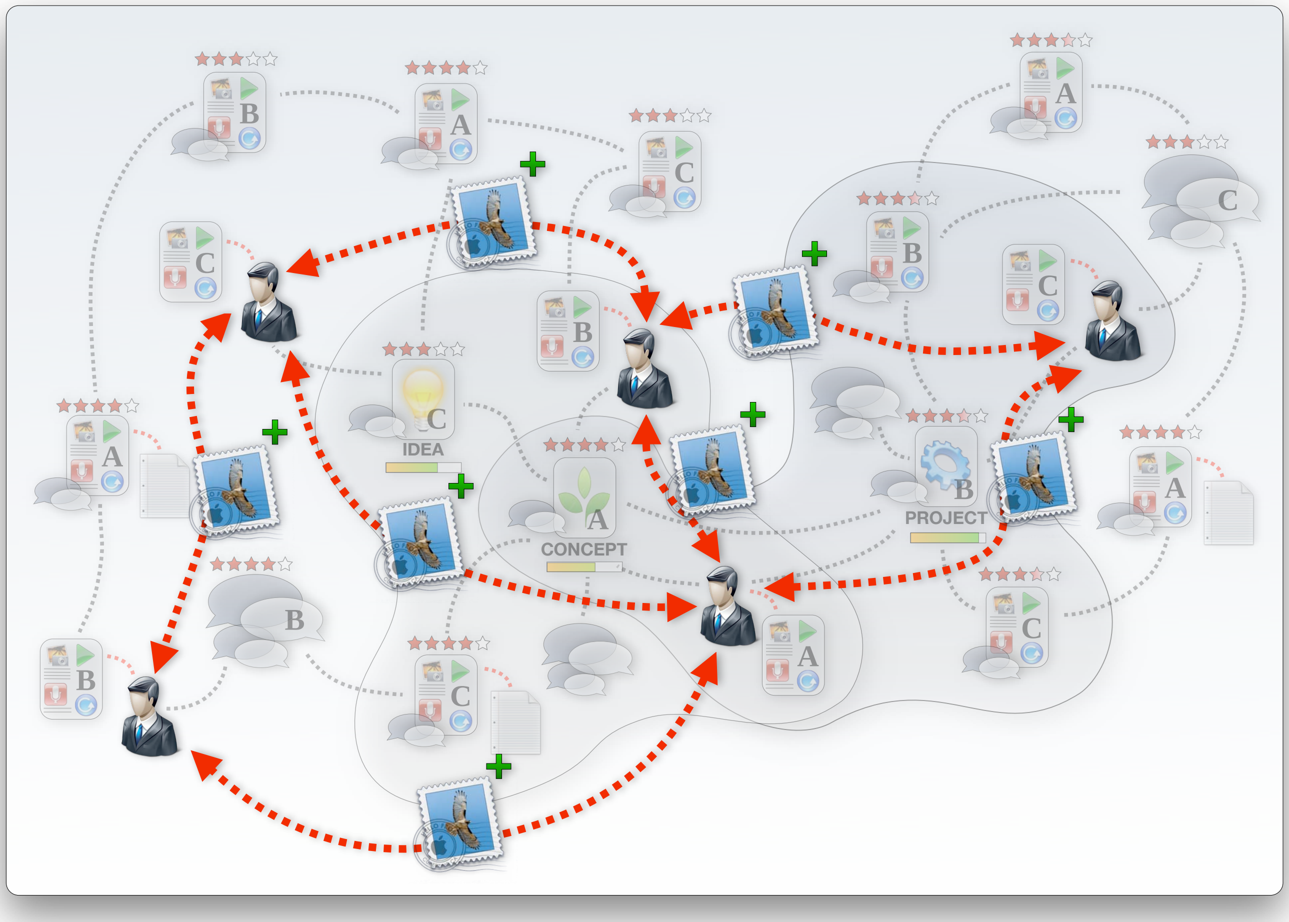


Figure 72 – Providing adequate communication channels support socialization

The Drupal core includes a **Contact** module that provides system-wide contact forms through which users may send an e-mail to each other. This does however not provide sufficient communication channels to the system.

The **Messaging** module provides a framework for a number of communication methods that includes private messages, SMS services and HTML-based e-mails. The module does not in itself provide the actual services, but rather functions as an interface between the Drupal core and modules that provide the communication methods.

Privatemsg is an example of such a module that adds private messaging capabilities to the information system. Private messages may be set as the default communication method within the Messaging module's communication framework. The same input formats that are available to create a media-rich environment for adding content are also available for the composition of private messages, and users may organize their private messages in folders.

The value of the private messages as a method of communication is increased by adding "Write to author" links on individual content items. This allows network members to contact authors directly to initiate a discussion that may result in implicit and tacit knowledge transfer. Contact details for users (e.g. telephone number, e-mail address, preferred instant messaging service and username) are made available on profile pages.

The Token module is another example of a “helper” module that allows other modules to perform powerful functions. The Token module captures core information during system activity and makes it available to other modules. This information includes node types, node titles, author names, organic group names and taxonomy terms involved in the activity.

The Notifications module may be configured to monitor the information intercepted by the Token module and alerts users when activity that corresponds to that information occurs. This enables the system to offer a subscription service that keeps users up to date with developments related to knowledge objects or areas that they find interesting.

At the bottom of the display area for a content item, a range of subscription options is made available. These subscriptions include alerts for developments in the content item itself (i.e. edits and comments), or for new postings by the same author, new postings that correspond to a specific taxonomy term or new postings of the same content type. Notifications are especially helpful for alerts on activity in organic groups, as it keeps all group members up to date with the latest developments in Concept and Project Objects.

Users may maintain their own subscriptions by picking their preferred notification method and are free to unsubscribe from alerting services.

5.6. Navigation, Layout and Theme

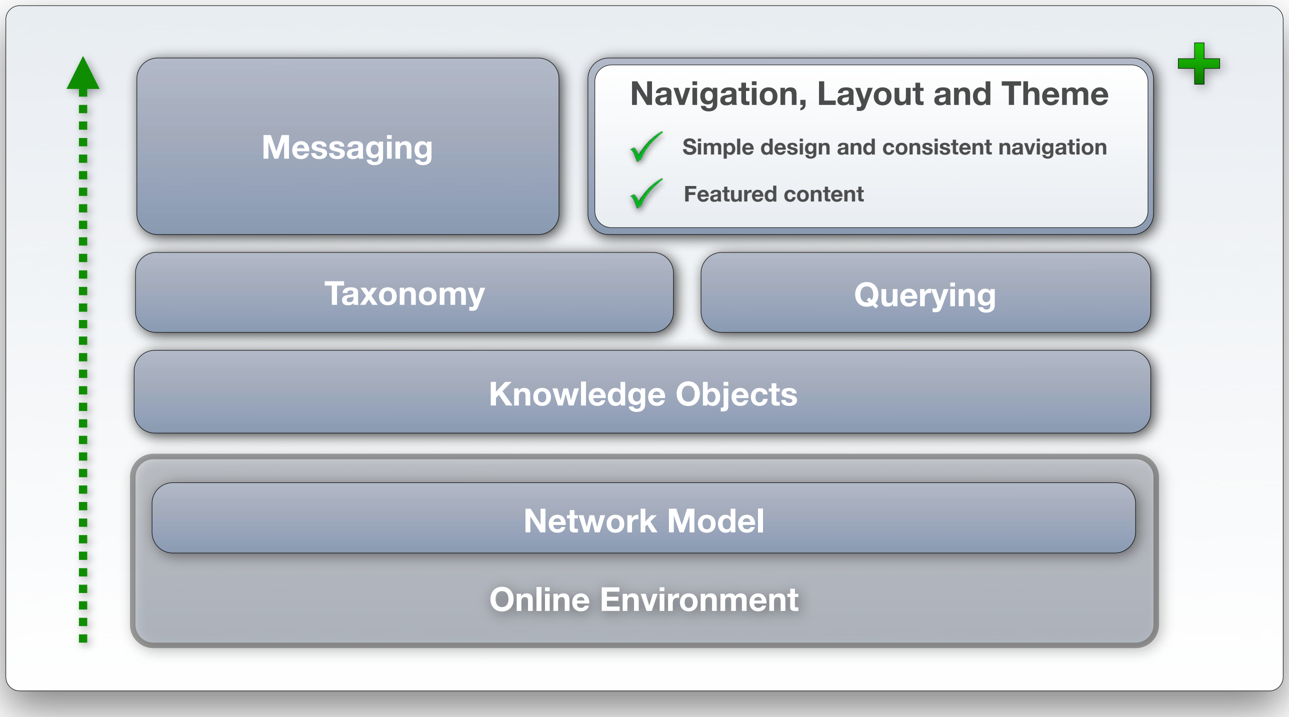


Figure 73 - Design layer 6: Navigation, Layout and Theme

Functionality Specification: Simple design and consistent navigation

Addressing requirements 11 and 13, and stated in Chapter 4.4.7.1 as:

The knowledge network that must be implemented is a rather complex structure of knowledge objects, relationships, communication functions, etc. It is therefore imperative that the platform has a look-and-feel that is simple and easy to use and understand. The presentation of information should be done in a way that is comprehensive, but at the same time relevant. Correct contextual use of menus and information blocks will ensure that the right information is shown to the right users at the right time.

Functionality Specification: Featured Content

Addressing requirement 14, and stated in Chapter 4.4.7.2 as:

To ensure that newly contributed knowledge object are noticed by users browsing the platform, content should be ordered with the latest items at the top of lists, unless otherwise specified by the nature of the list. Items that should always appear at the top of lists, regardless of its age, should also be supported.

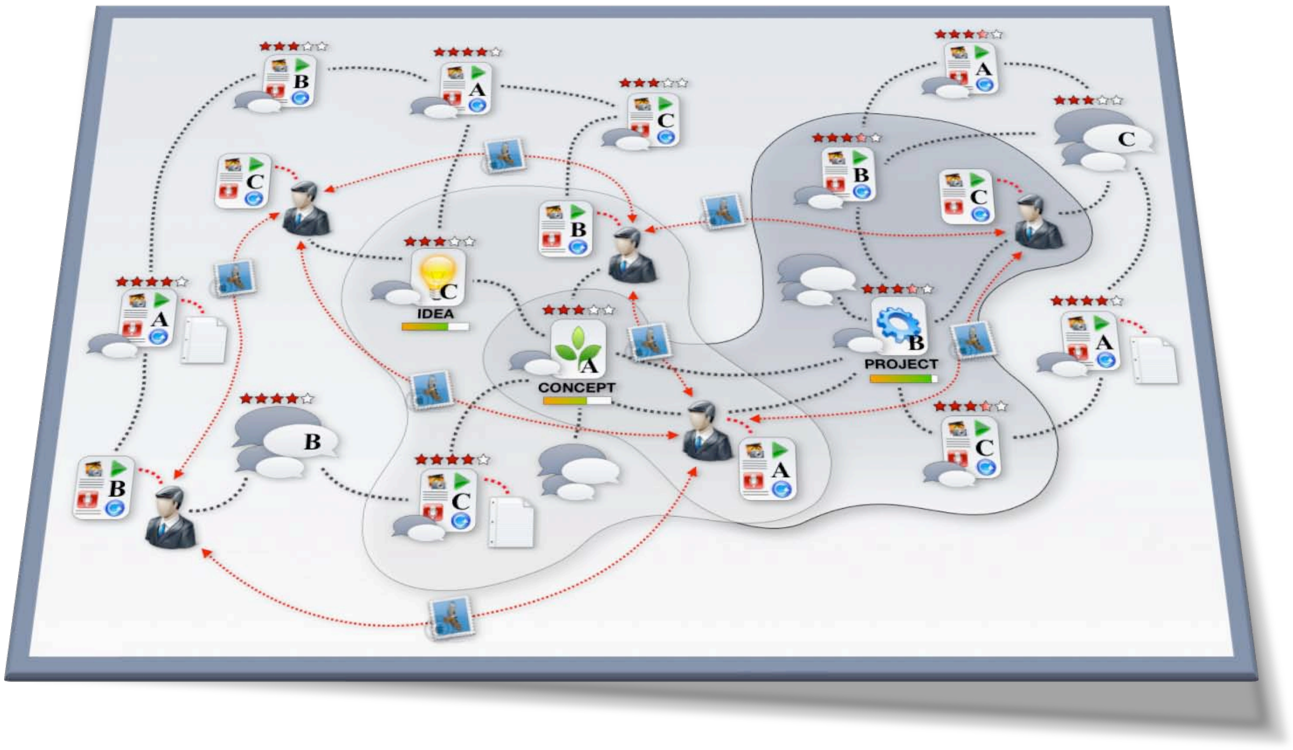


Figure 74 – Presenting the Information System's functionality in a simple interface adds value to the design

A simple and intuitive interface is required to allow users to generate the most value from an Information System that offers a wide range of functions. Drupal offers a templating system that can interchange graphical presentation of the system without any changes to the content items. This is especially helpful when it is needed to revamp the look-and-feel of the system without risking the loss of any content.

The core Drupal package includes several templates that all provide extremely simple interfaces. The Garland theme is the best of these as it offers simple presentation of information while maintaining a pleasing environment. The Drupal core package includes a **Color** module, which allows administrators to customize the color scheme of templates that are configured to include the Color module.

Configuring the layout of the interface is simplified by Drupal's use of "blocks". Elements of the system are assigned to blocks, which may then be placed in positions allowed by the template. In the case of the Garland template, these positions include a header, left and right sidebars, the main content display area and a footer. The **Panels** module allows advanced element positioning by creating functional element groupings for placement in template positions.

Menus that are maintained by the Drupal core are used to navigate the system. These menus include a user menu for personal links (right sidebar), a menu for adding content (right sidebar) and a menu for options specifically related to the Innovation Life Cycle (right sidebar). The header includes a menu for differentiating between different types of knowledge objects, and the left sidebar contains menus that reflect the taxonomy scheme (created with the Taxonomy Menu module), helping users to navigate the network knowledge base. A search box is also located in the left sidebar. Implementing the DHTML Menu module makes all the menus collapsible and adds the ability to open menu options without page reloads.

The Drupal core Path module allows administrators to create user-friendly URL's for pages (e.g. http://www0.sun.ac.za/bingtest/innonet/?q=project1 in stead of http://www0.sun.ac.za/bingtest/innonet/?q=node/95. This is useful for reference purposes, as is done in this document.

In the case of a complex taxonomy scheme being implemented, the tagging interface becomes cumbersome and results in an extremely long page. The Taxonomy Super Select module solves this problem by replacing the default tagging interface with a user-friendly collapsing interface.

5.7. System Design Specifications

The design specifications that were derived in Chapters 5.1 to 5.6 may be seen as a version of the Information System Architecture that was specified in Chapter 4.7, translated into technologies, frameworks and modules (refer to Figure 75). These System Design Specifications are used to guide the system development process in Chapter 6.

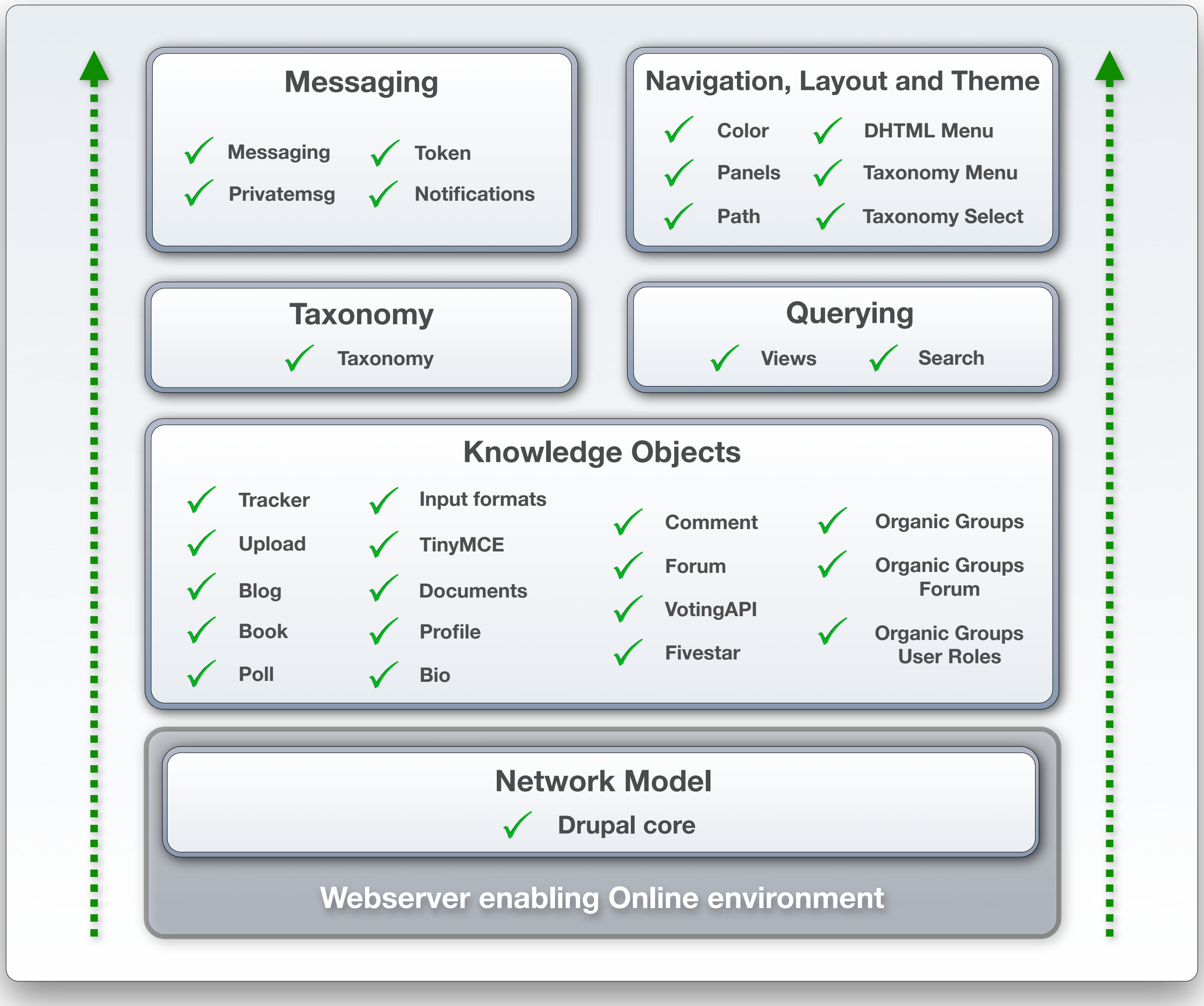


Figure 75 - Layered presentation of System Design Specifications

Chapter 6

Development

SDLC Step 6: Development

"Converts a design into a complete information system."

- Wikipedia

In Chapter 6, the detailed system design that was developed in Chapter 5 is converted to a complete information system. Only the approach to the development process is discussed, with some of the details being presented in a series of tutorial videos that are available online at:

http://www0.sun.ac.za/bingtest/innonet/development

System development is approached with the familiar layered System Design Specification depicted in Figure 76 as a guide. Technologies, frameworks and modules are installed and configured in the order shown below, although some variation in the sequence does occur due to practical implications.

The implementation of each layer of the System Design Specification is discussed in a separate tutorial video. All of these videos are available online at:

http://www0.sun.ac.za/bingtest/innonet/?q=development.

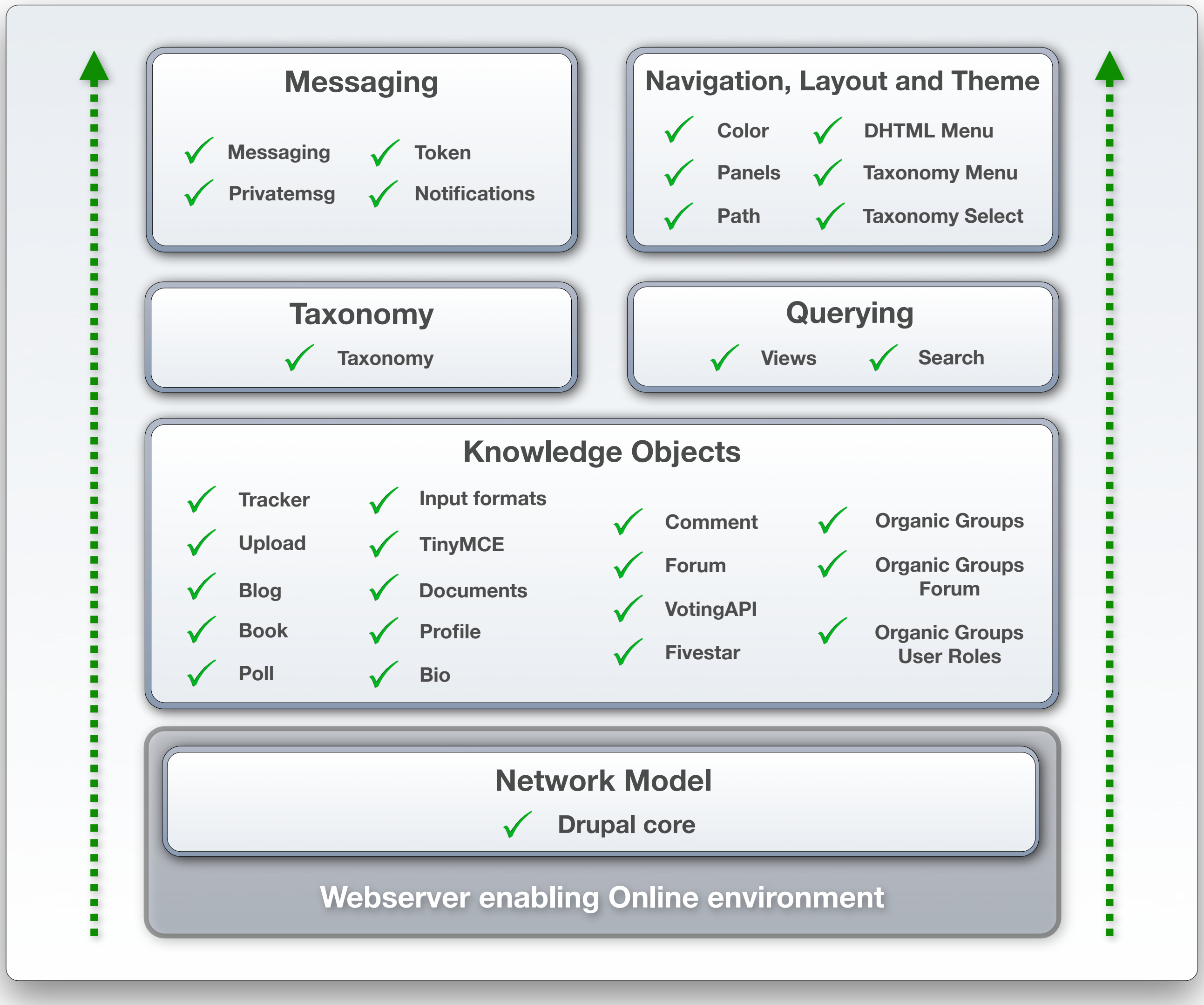


Figure 76- The layered System Design Specification that is used as a guide for the development process

It follows from the layered System Design Specification in Figure 76 that the development process will be approached in the same 6 phases that were followed during System Design in Chapter 5:

1. Network model within Online environment
2. Knowledge Objects
3. Taxonomy
4. Querying
5. Messaging
6. Navigation, Layout and Theme

A brief background of the technologies and modules that are implemented is provided, along with references to more complete discussions on element functionality. Note that the use of an extendable Content Management System (i.e. Drupal) allows for the high-level development of a complex web-application without any physical programming.

Note: In the discussion of these six Development phases, some uncaptioned diagrams are used to aid the representation of functionality that is implemented in each phase.

6.1. Network Model

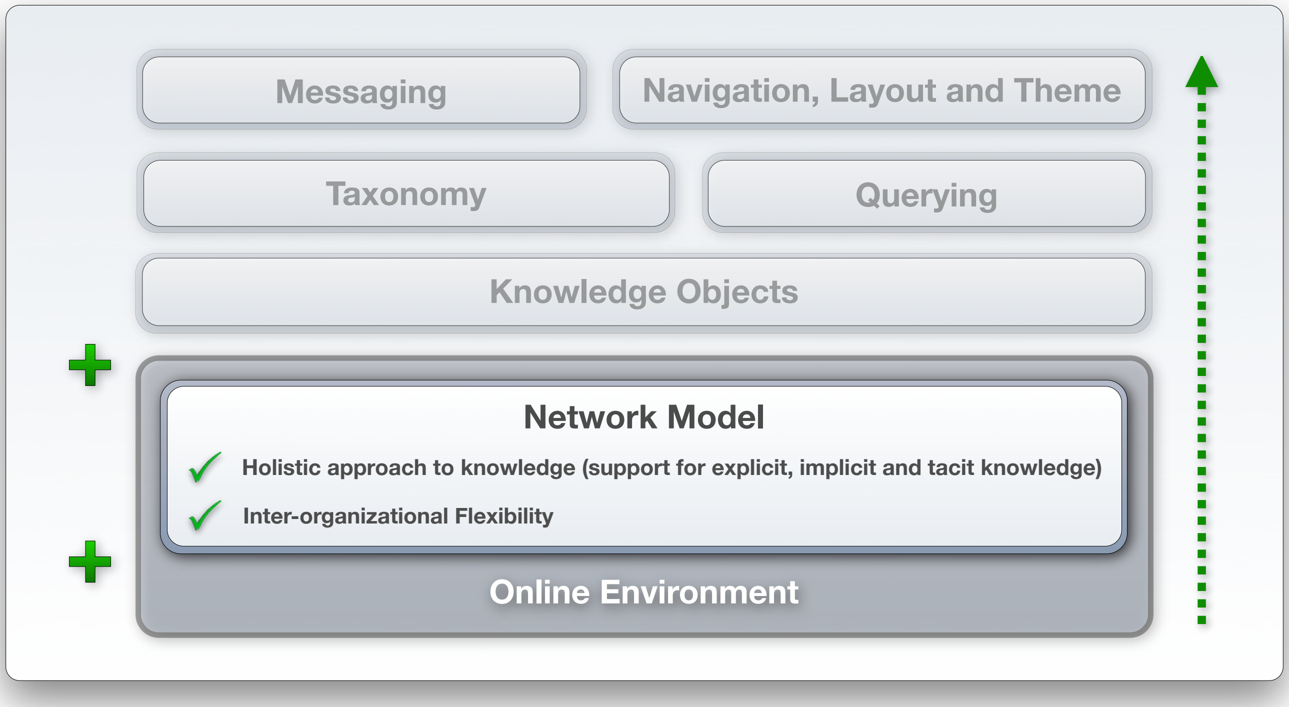

Messaging
Navigation, Layout and Theme
Taxonomy
Querying
Knowledge Objects
Network Model
✓ Holistic approach to knowledge (support for explicit, implicit and tacit knowledge)
✓ Inter-organizational Flexibility
Online Environment

Figure 77 - Development phase 1: Network model within an Online environment

The installation and configuration of elements included in Development phase 1 are covered in Tutorial Video 1: Network Model.

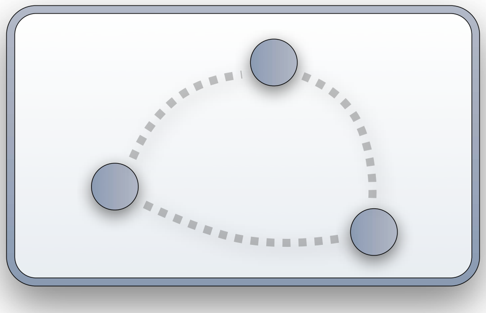
6.1.1. Webserver enabling Online environment

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.1.

6.1.1.1. Webserver configuration

Provides the online environment that is required by the Knowledge Network Architecture.

6.1.2. Network Model



For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.1.

6.1.2.1. Drupal core

The Drupal core initiates the basic features necessary for the implementation of the network model with its dynamic approach to content management and the ability to handle various content types. Its approach to access and user management furthermore provides the basis for inter-organizational flexibility.

6.1.2.2. Admin Menu

A contributed module that simplifies the Drupal core's administration interface and is used as a tool in the development process (Kudwien [25]).

6.2. Knowledge Objects

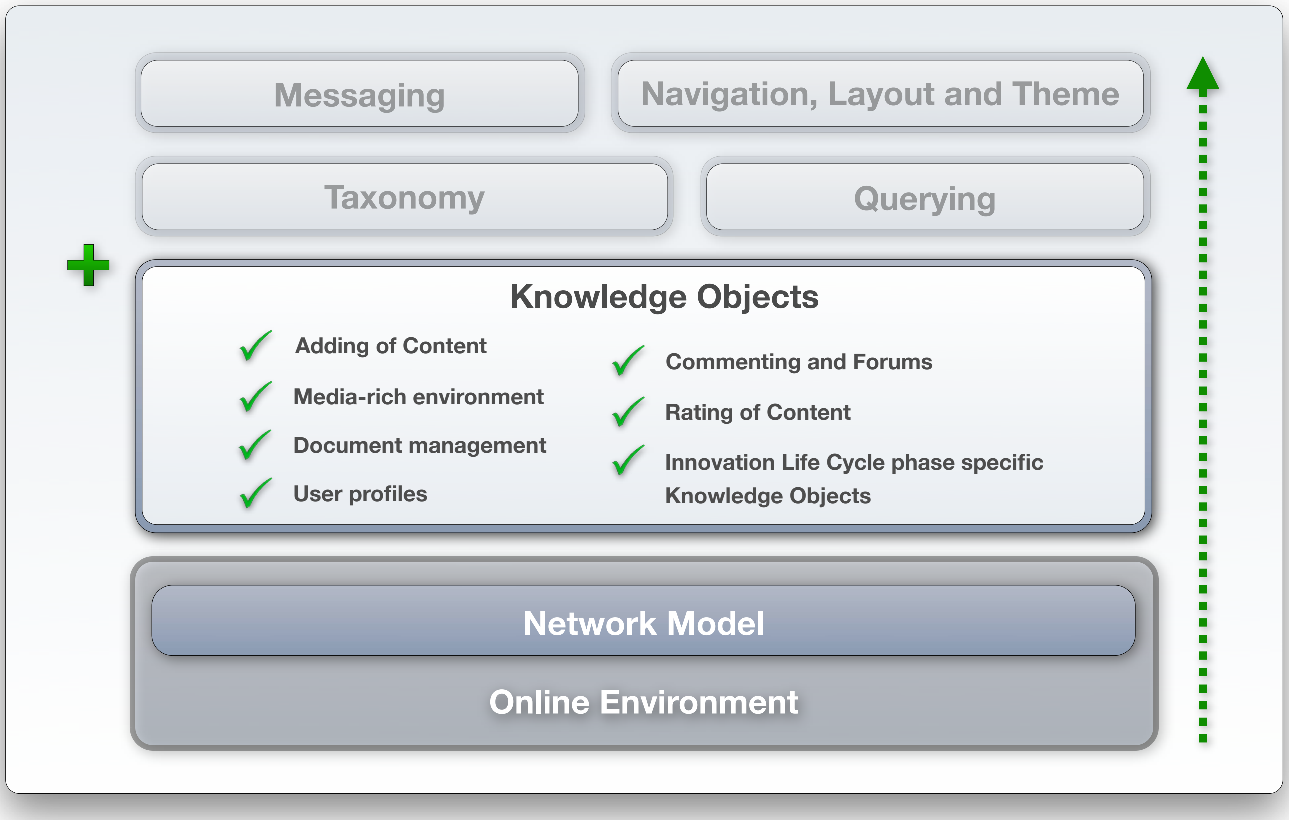
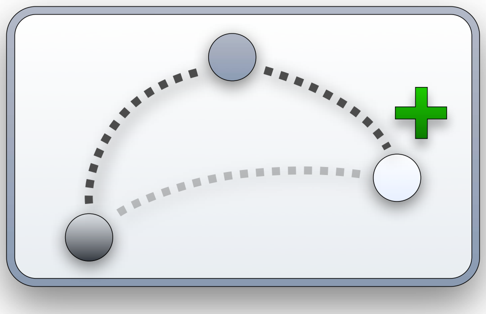


Figure 78 – Development phase 2: Knowledge Objects

The installation and configuration of elements included in Development phase 2 are covered in Tutorial Video 2: Knowledge Objects.



6.2.1. Adding of Content

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.1.

6.2.1.1. Tracker

A Drupal core module that enables version control.

6.2.1.2. Upload

A Drupal core module that allows users to attach files to posts.

6.2.1.3. Statistics

A Drupal core module that enables the logging of all system activity.

6.2.1.4. Blog

A Drupal core module that adds the *blog* content type and collects all such posts from a single user to form a personal blog.

6.2.1.5. Book

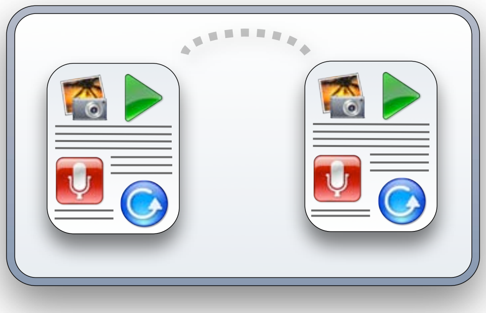
A Drupal core module that adds the *book* content type and allows users collaboratively build a multi-post structure.

6.2.1.6. Poll

A Drupal core module that adds the *poll* content type and allows users to create simple polls as content items.

6.2.2. Media-rich environment

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.2.



6.2.2.1. Input Formats

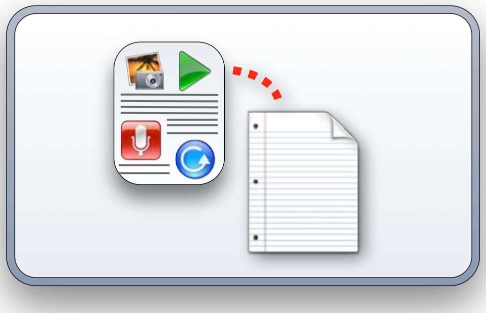
The Drupal core includes a number of default input formats that may be configured to allow users to take advantage of a media-rich environment, i.e. text, images, video, audio.

6.2.2.2. TinyMCE

A contributed module that embeds the TinyMCE what-you-see-is-what-you-get editor to allow users who are not familiar with HTML syntax to take advantage of the features enabled by the customized input formats (Reinen [41]).

6.2.3. Document Management

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.3.



6.2.3.1. Content Construction Kit (CCK)

Contributed module that allows administrators to edit the structure of content types by appending custom fields (Chaffer [8]). Used to customize the *document* content type, which adds documents to the network as knowledge objects.

6.2.4. User Profiles

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.4.

6.2.4.1. Profile

A Drupal core module that allows administrators to configure the structure of user profiles by adding categories with fields.

6.2.4.2. Bio

A contributed module that creates the *biography* content type and links a content item of this type to each user profile. Users may then maintain this content item as their personal biography (Robbins [44]). Biography nodes attached to user profiles integrate with other content items (e.g. via the taxonomy scheme), thereby adding users to the network as knowledge objects.

6.2.4.3. Profile Category Weight

A contributed module that allows administrators to configure the order in which information categories are displayed on user profiles (Barreiro [2]).

6.2.4.4. Register With Picture

A contributed module that allows users to upload an avatar upon registration of their user account (Milano [28]).

6.2.5. Commenting and Forums

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.5.

6.2.5.1. Comment

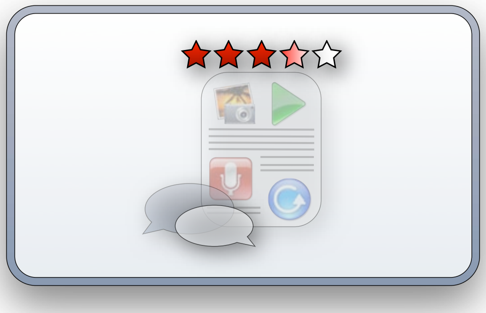
A Drupal core module that implements a powerful commenting model within the system and allows users to comment on content.

6.2.5.2. Forum

A Drupal core module that adds the *forum topic* content type and allows users to initiate forum discussions. Administrators may organize forum discussions in “containers”.

6.2.6. Rating of Content

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.6.



6.2.6.1. VotingAPI

A contributed module that creates a framework for the storage and retrieval of votes, which may be used by other modules (Eaton [17]).

6.2.6.2. Fivestar

A contributed module that uses the VotingAPI framework and adds a simple rating widget to content types as configured by the administrator (Haug [22]).

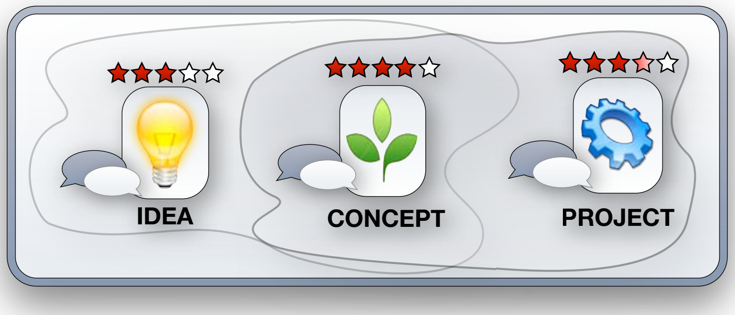
6.2.7. Innovation Life Cycle phase specific Knowledge Objects

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.2.7

6.2.7.1. Organic Groups

A contributed module that allows the formation of node clusters around a central node (Weitzman [60]), thereby creating the dynamic content structure needed for the Concept and Project Objects. CCK is used to customize the structure of the *concept* and *project* content types (created by the Drupal core), of which instances will be used to form the central nodes of organic group clusters. Users may join these groups to become subscribed members, and posts from a wide variety of types may be posted into the group.

Note: The Drupal core is also used to create a new *idea* content type that is customized with the Content Construction Kit. Instances of this content type will form Idea Objects.



6.2.7.2. Organic Groups Forum

A contributed module that adds a dedicated forum discussion to each organic group cluster, thereby extending the communication functionality of the Concept and Project Objects (Constantine [10]).

6.2.7.3. Organic Groups User Roles

A contributed module that allows organic group administrators to assign roles to group members (Parker [37]). These roles may differ from between group types, and a user may be assigned different roles in different groups.

6.3. Taxonomy

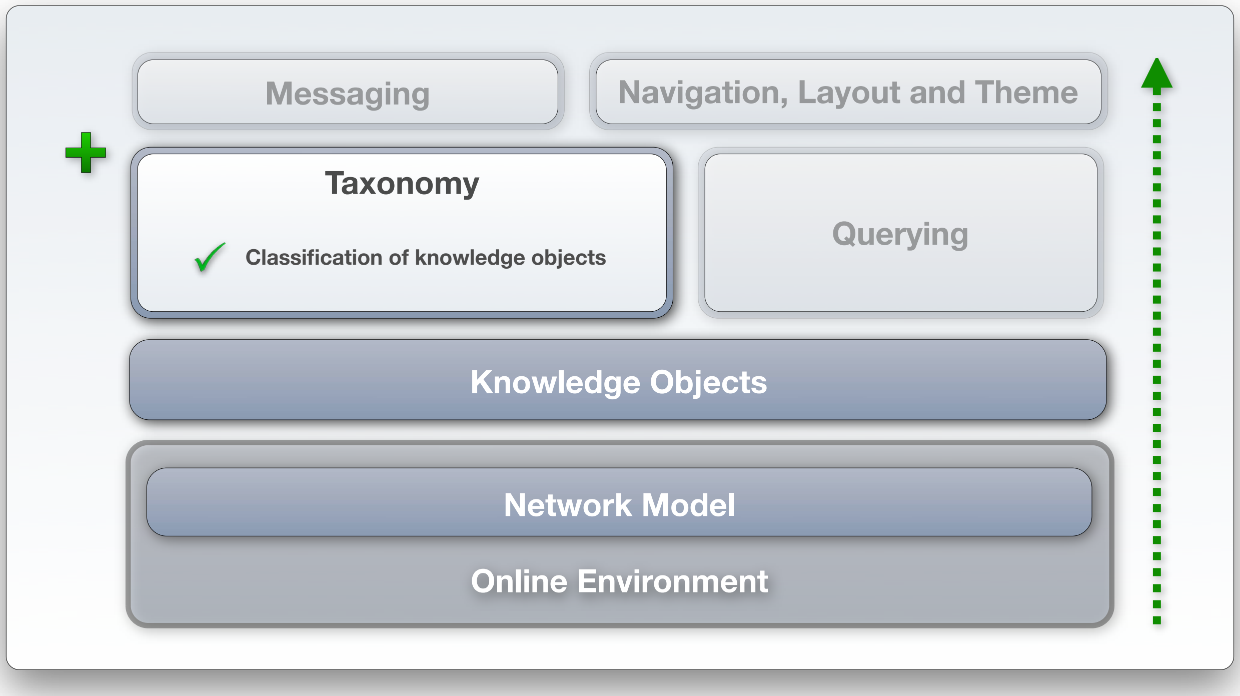
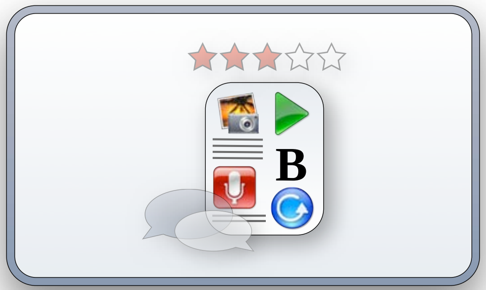


Figure 79 - Development phase 3: Taxonomy

The installation and configuration of elements included in Development phase 3 are covered in Tutorial Video 3: Taxonomy.

6.3.1. Classification of Knowledge Objects

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.3.



6.3.1.1. Taxonomy

A Drupal core module that allows users to classify content by tagging items with keywords. These keyword terms are arranged in vocabularies that may include a predefined set of terms or set up to allow free tagging. Terms in vocabularies may form complex hierarchies and items may be tagged with multiple terms.

6.3.1.2. Similar By Terms

A contributed module that displays a list of content items that are classified similarly to the one that is currently being displayed (Robbins [45]). This serves as suggestions that want to explore related content.

6.4. Querying

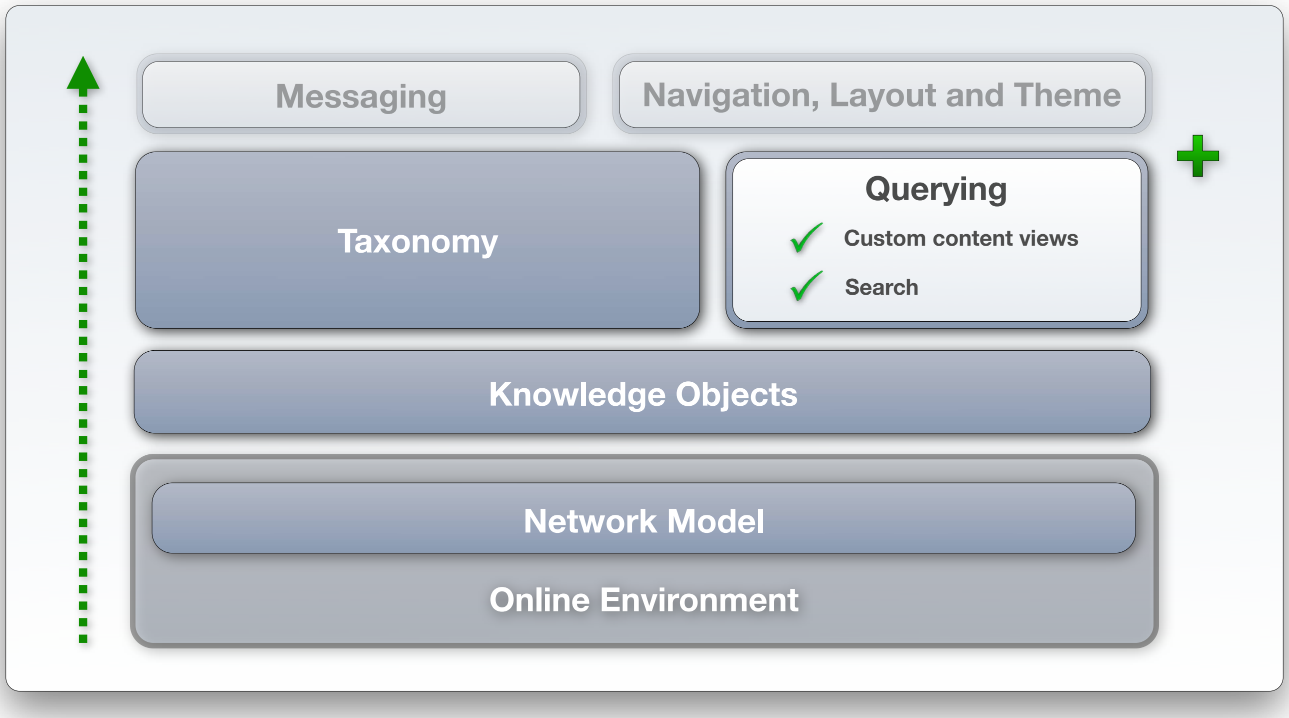
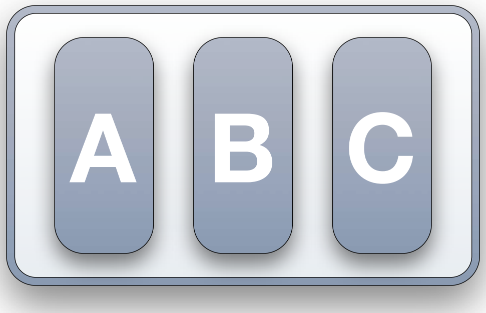


Figure 80 - Development phase 4: Querying

The installation and configuration of elements included in Development phase 4 are covered in Tutorial Video 4: Querying.

6.4.1. Executive content views

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.4.



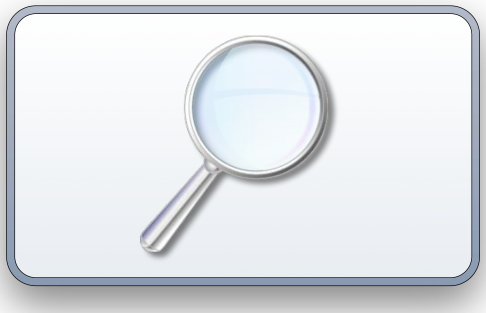
6.4.1.1. Views

A contributed module that performs as a query builder to extract custom views of content within the database (Miles [30]).

6.4.2. Search function

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.4.

6.4.2.1. Search



6.5. Messaging

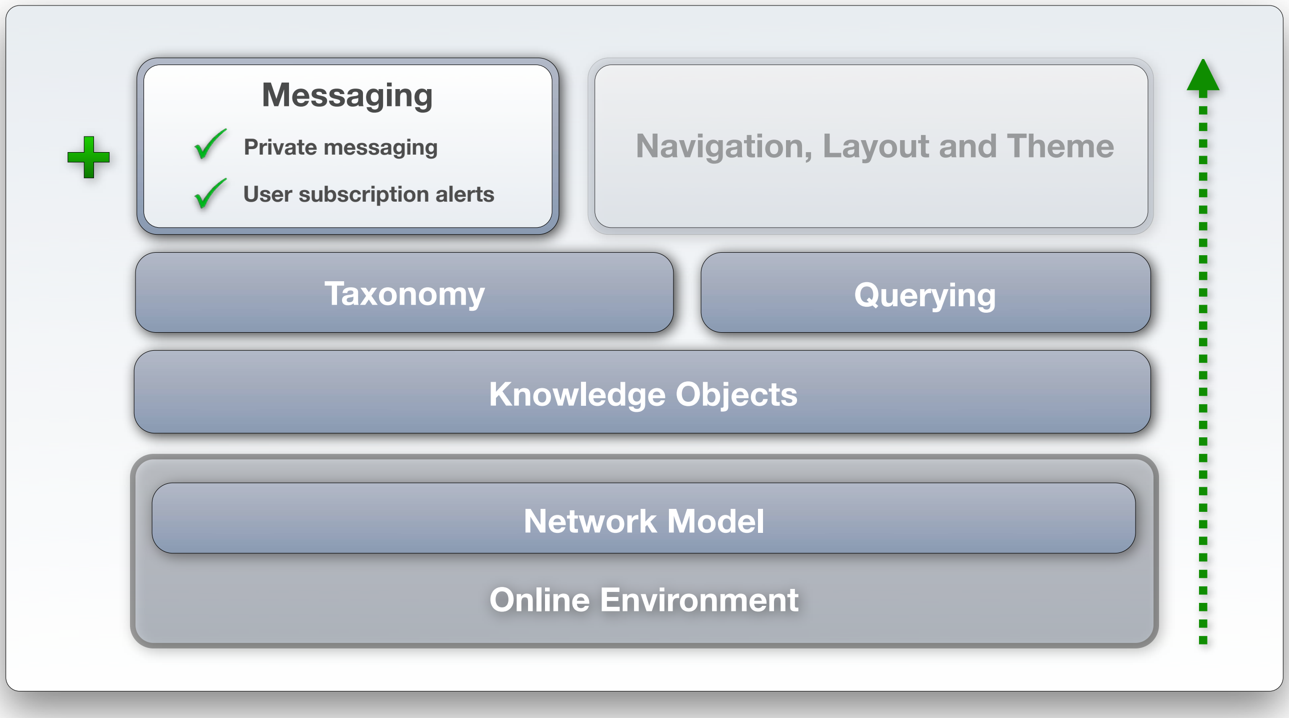


Figure 81 - Development phase 5: Messaging

The installation and configuration of elements included in Development phase 5 are covered in Tutorial Video 5: Messaging.

6.5.1. Messaging

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.5.1.

6.5.1.1. Messaging

A contributed module that creates a framework of communication methods between users (Reyero [42]).

6.5.1.2. Privatemsg

A contributed module that uses the framework created by the Messaging module and adds private messaging between users as a communication method to the system (Terenchuk [55]).

6.5.2. User subscription alerts

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.5.2.

6.5.2.1. Token

A contributed module that intercepts core system information and makes it available to other modules (Eaton [16]). This information includes node types, node titles, author names, organic group names and taxonomy terms.

6.5.2.2. Notifications

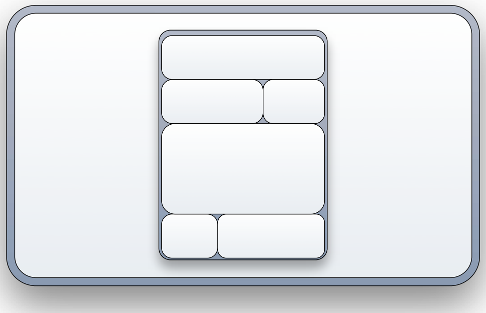
A contributed module that may be configured to monitor the information intercepted by the Token module and alerts users when activity that corresponds to that information occurs (Reyero [43]). This allows users to e.g. subscribe to all posts from a specific author, taxonomy term or belonging to a certain organic group and be notified when any such activity occurs. This function may, amongst other uses, be used to notify members of a concept development group of the posting of new content into the group.

6.6. Navigation, Layout and Theme

The installation and configuration of elements included in Development phase 6 are covered in Tutorial Video 6: Navigation, Layout and Theme.

6.6.1. Simple design and consistent navigation

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.6.



6.6.1.1. Color

A Drupal core module that allows administrators to adjust the color scheme of graphic themes which employ the Color module to determine their look-and-feel.

6.6.1.2. Panels

A contributed module that allows administrators achieve advanced customized page layouts (Miles [29]). By posting content to a specific “panel” and positioning the “panel” itself, content display is customized.

6.6.1.3. Path

A Drupal core module which allows administrators and authors to specify customized user-friendly URL’s that link to system content, e.g. http://www0.sun.ac.za/bingtest/innonet/?q=development.

6.6.1.4. DHTML Menu

A contributed module which allows menus displayed in side-bars to collapse and expand without reloading the page, which reduces page-loads and increases layout efficiency (Ilyaran [24]).

6.6.1.5. Taxonomy Menu

A contributed module that creates menus (with parent and child items) according to taxonomy vocabulary configurations (Halumi [21]). This enables users to navigate to the network knowledge base with the taxonomy scheme as a guide, making it much easier to find content on specific topics. Menus based on the taxonomy scheme maintain a consistent user mental model of the knowledge base.

6.6.1.6. Taxonomy Super Select

A contributed module that simplifies the interface for assigning taxonomy terms during content creation (Christmas [9]). In the case of a complicated taxonomy vocabulary configuration, the default Drupal interface may become cumbersome, and this module addresses this issue.

6.6.2. Featured content

For a discussion of the functionality of elements in this sub-layer, refer to Chapter 5.6.

The Drupal core allows content items to be configured as “Sticky at the top of lists”, which will keep the item at the top of content listings, regardless of the sort criteria.

Chapter 7

Showcase

SDLC Step 7: Integration and Test

"Illustrates that the developed system conforms to requirements as specified in the Information System Architecture."

- Wikipedia

Chapter 7 presents a demonstration of the functionality of the developed information system. In doing this, Chapter 7 links to Chapter 2 by proving the hypothesis stated in that Chapter. A number of identified key functionalities are used as arguments in this proof.

7.1. Approach to the Testing phase of the SDLC

The philosophy towards to the testing phase of the developed Information System's life cycle is heavily influenced by both the needs of the research method, as well as the scope of the project.

Time constraints, and subsequently project scope, does not allow for extensive physical testing of the system within a production environment. The method followed in this research project does however state a hypothesis that may be proved with successful demonstration of a number of identified key system functionalities (refer to the refined problem statement in Chapter 3.3 and hypothesis in Chapter 3.4).

The aim of this chapter is therefore to illustrate these functionalities with the use of sample data. Successful functionality demonstration will be seen as proof of the hypothesis and therefore the solving of the problem (refer to Chapter 3.3). Comments on the testing of the Information System will be provided in Chapter 8.

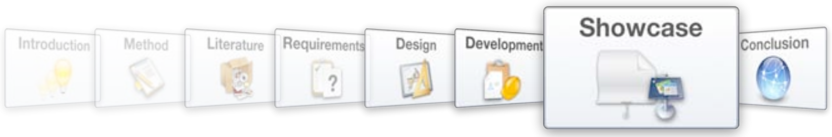
7.2. Demonstration outline

The demonstration of the developed Information System's functionality will be done according to the outline provided in Figure 83, aligning it with the proof of the hypothesis (refer to Chapter 3.4):

Figure 83 highlights the key functional segments that need to be demonstrated for proof of the hypothesis:

1. Support for Knowledge Work Processes within an Integrated Knowledge Network
    a. Knowledge Network Architecture
    b. Knowledge Work Processes
2. Support for the full Innovation Life Cycle
    a. Idea Phase
    b. Concept Phase
    c. Project Phase

Note that this navigation structure differs from that used in Chapters 4, 5 and 6, as the structure followed in those chapters combined the needs of a Knowledge Network and Innovation for system development purposes. Chapter 7 is aimed at proof of the hypothesis.



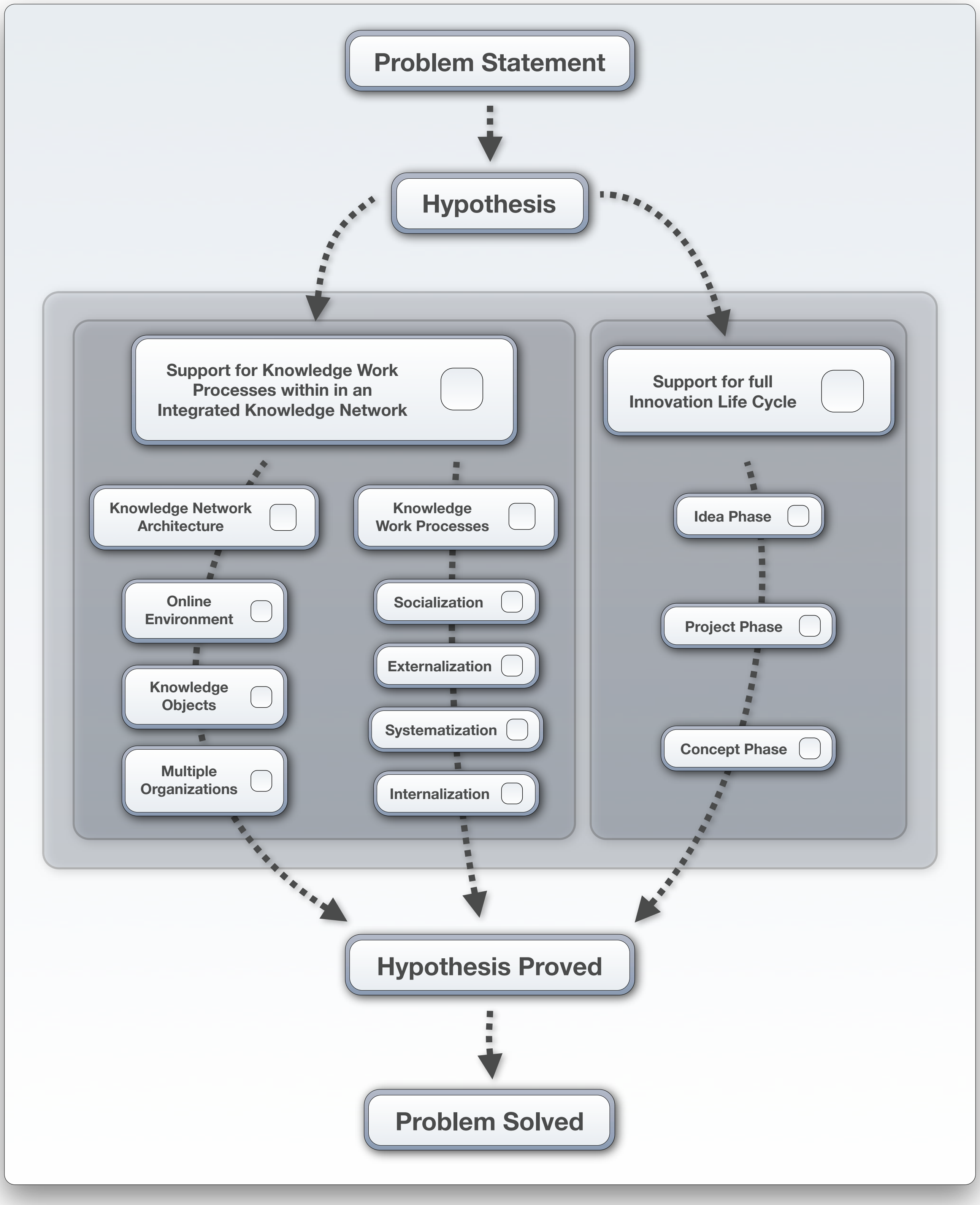


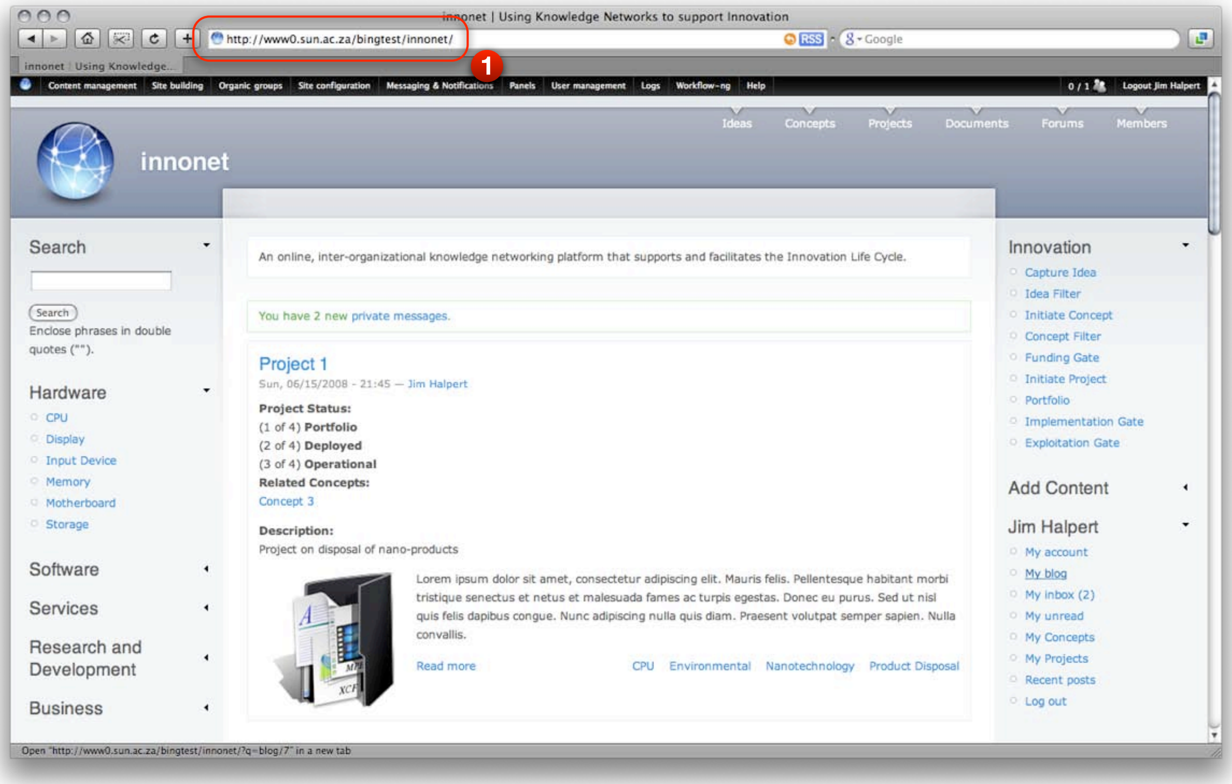
Figure 83 - Demonstration outline, aligning Chapter 7 with the proof of the hypothesis

7.3. Demonstration of support for Knowledge Work Processes within an Integrated Knowledge Network

7.3.1. Knowledge Network Architecture

The Knowledge Network Architecture includes an ICT and an organizational component. The ICT component requires that a network model with a holistic approach to knowledge be implemented in an online environment (refer to Chapter 5.1). From an organizational perspective, multiple organizations should be accommodated to enable an Integrated Knowledge Network.

7.3.1.1. Online Environment



Enclose phrases in double quotes ("").

An online, inter-organizational knowledge networking platform that supports and facilitates the Innovation Life Cycle.

You have 2 new private messages.

Project 1

Sun, 06/15/2008 - 21:45 — Jim Halpert

Project Status:
(1 of 4) **Portfolio**
(2 of 4) **Deployed**
(3 of 4) **Operational**
Related Concepts:
Concept 3

Description:
Project on disposal of nano-products

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris felis. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec eu purus. Sed ut nisl quis felis dapibus congue. Nunc adipiscing nulla quis diam. Praesent volutpat semper sapien. Nulla convallis.

Open "http://www0.sun.ac.za/bingtest/innonet/?q=blog/7" in a new tab

Figure 84 - Online environment that may be accessed with any web-browser

An online environment is provided through an installation of the Drupal Content Management System on a webserver that supports the PHP scripting language and mySQL databases.

This installation may be accessed with any web-browser at the following URL:

http://www0.sun.ac.za/bingtest/innonet

7.3.1.2. Knowledge Objects

a) Adding content of various types in a media-rich environment

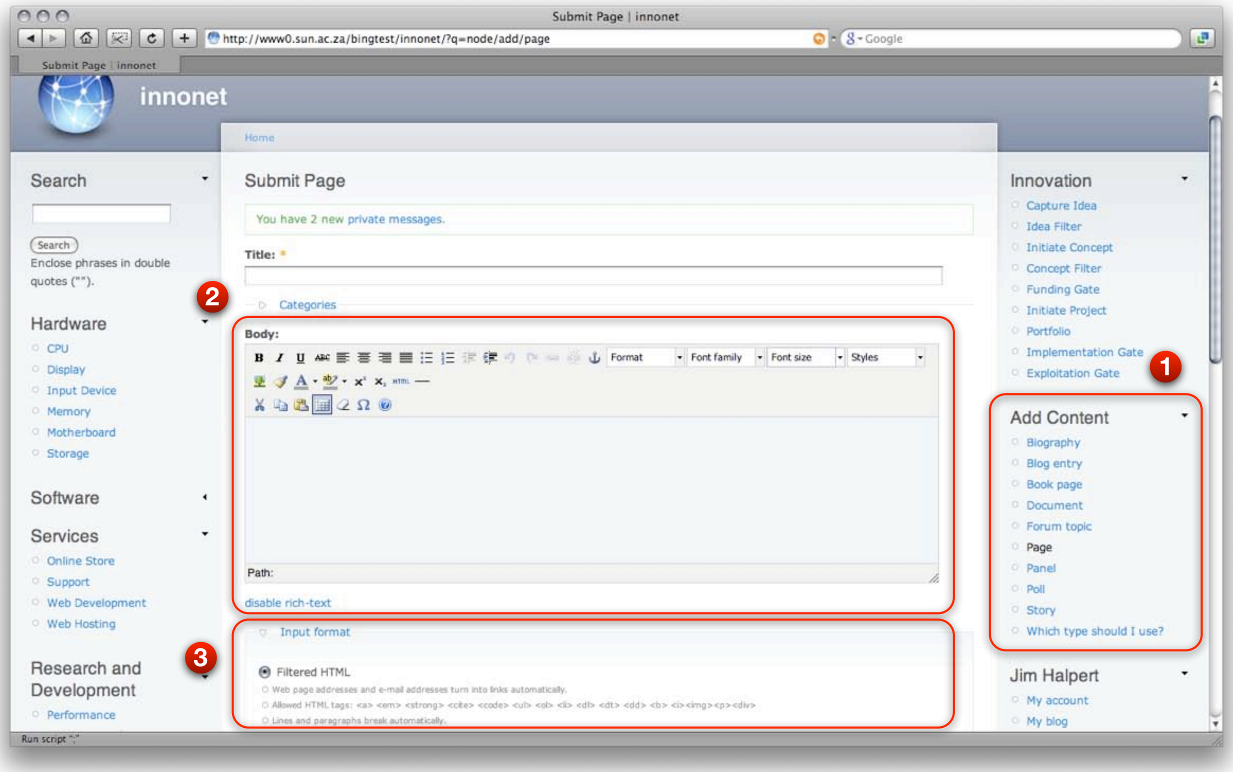
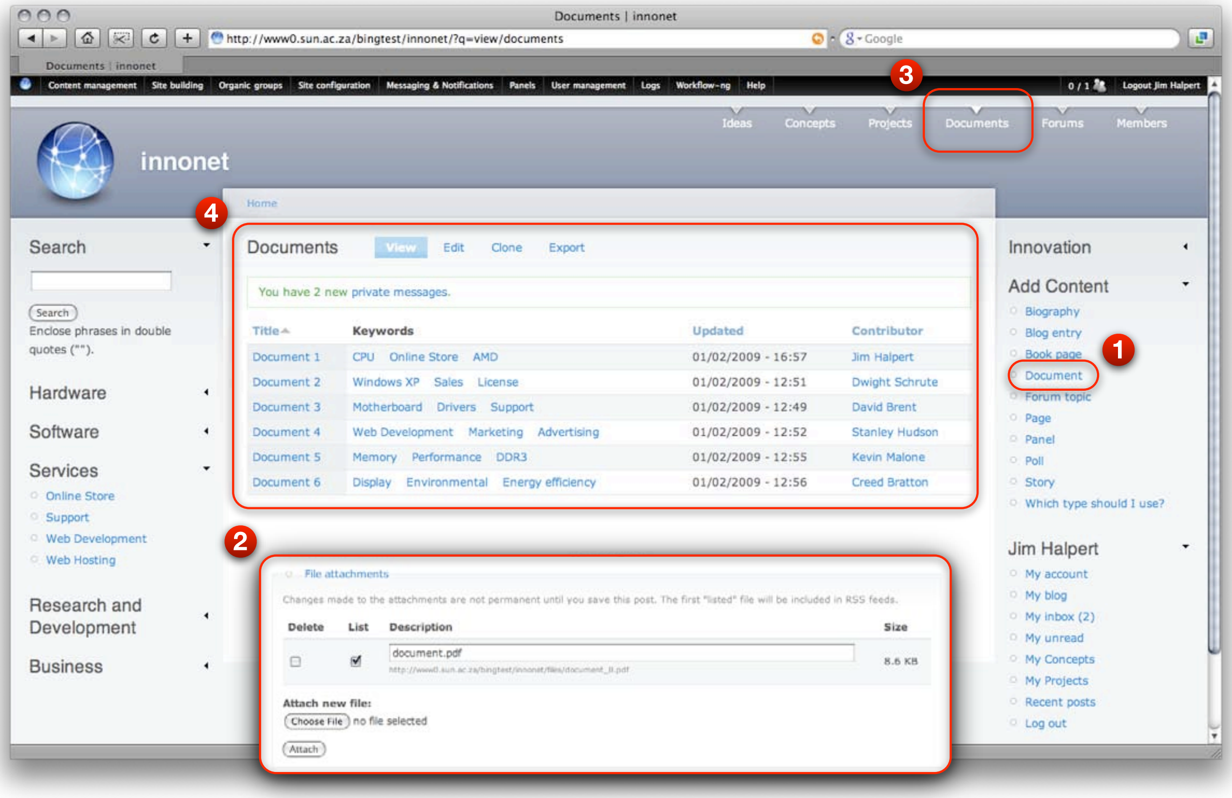


Figure 85 - Various types of content may be added in a media-rich environment

Network members may add various types of content to the system through the “Add Content” menu in the right sidebar (see number 1 in Figure 85). Available content types include blog entries, book pages, forum topics, pages, polls and stories (refer to Chapter 5.2.1). The TinyMCE editor (number 2 in Figure 85) is used to ease use of the Filtered HTML input format (number 3 in Figure 85) that is employed as default input format by the system (refer to Chapter 5.2.2). This interface allows users to combine text, images, audio and video in their postings.

The Drupal content management system handles content in such a way that it forms knowledge objects that carry explicit knowledge within the network knowledge base.

b) Document Management



You have 2 new private messages.

Title	Keywords	Updated	Contributor
Document 1	CPU Online Store AMD	01/02/2009 - 16:57	Jim Halpert
Document 2	Windows XP Sales License	01/02/2009 - 12:51	Dwight Schrute
Document 3	Motherboard Drivers Support	01/02/2009 - 12:49	David Brent
Document 4	Web Development Marketing Advertising	01/02/2009 - 12:52	Stanley Hudson
Document 5	Memory Performance DDR3	01/02/2009 - 12:55	Kevin Malone
Document 6	Display Environmental Energy efficiency	01/02/2009 - 12:56	Creed Bratton

File attachments

Changes made to the attachments are not permanent until you save this post. The first "listed" file will be included in RSS feeds.

Delete	List	Description	Size
☐	✓	document.pdf http://www0.sun.ac.za/bingtest/innonet/files/document_0.pdf	8.6 KB

Attach new file:
Choose File no file selected
Attach

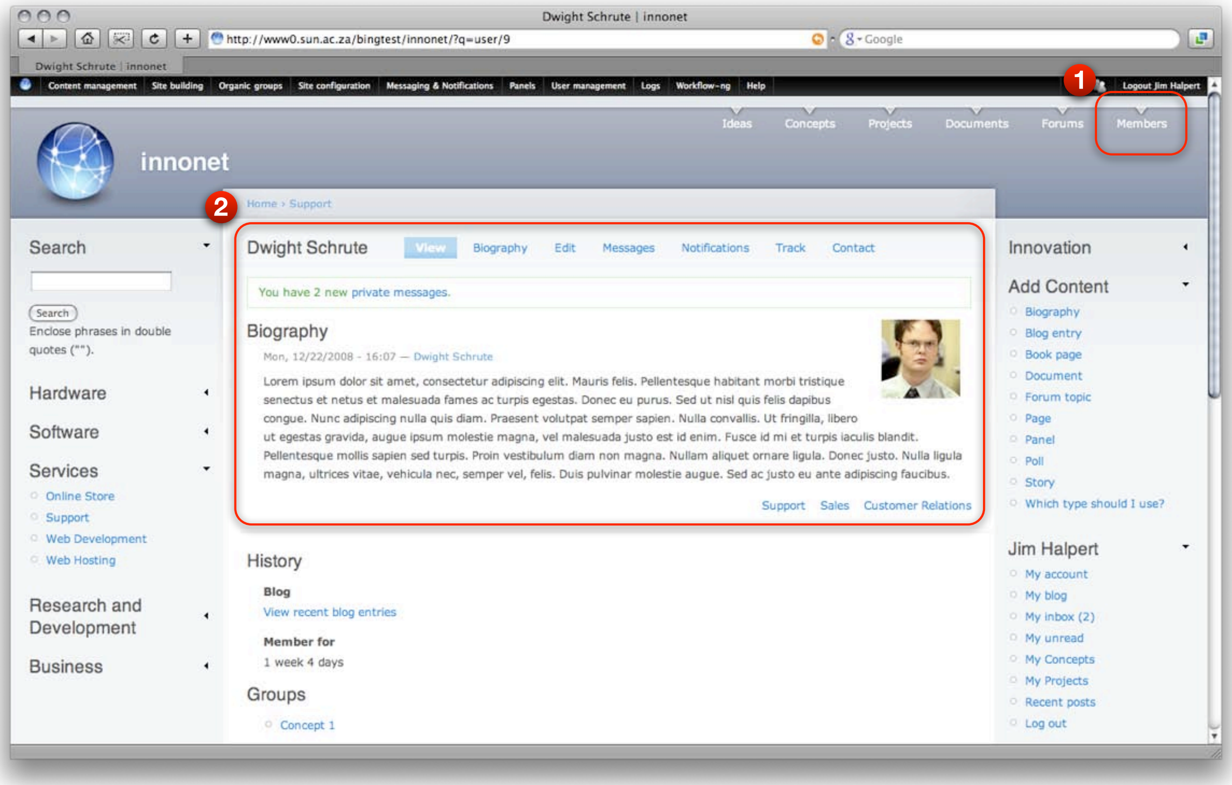
Figure 86 - Documents are handled as content within the system

The Drupal core is used to create a content type dedicated to document management (refer to Chapter 5.2.3). Users may add documents to the network knowledge base by creating a node of the "document" content type through the "Add Content" menu (see number 1 in Figure 86) and then attaching the relevant file before submission of the node (number 2 in Figure 86).

A complete listing of all the documents in the network knowledge base (number 4 in Figure 86) may be obtained through the "Documents" link in the top navigation menu (number 3 in Figure 86).

The Drupal-based system is therefore manipulated in such a way that documents form knowledge objects that carry explicit knowledge within the network knowledge base.

c) User Biographies



Enclose phrases in double quotes ("").

Dwight Schrute

You have 2 new private messages.

Biography

Mon, 12/22/2008 - 16:07 — Dwight Schrute

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris felis. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec eu purus. Sed ut nisl quis felis dapibus congue. Nunc adipiscing nulla quis diam. Praesent volutpat semper sapien. Nulla convallis. Ut fringilla, libero ut egestas gravida, augue ipsum molestie magna, vel malesuada justo est id enim. Fusce id mi et turpis iaculis blandit. Pellentesque mollis sapien sed turpis. Proin vestibulum diam non magna. Nullam aliquet ornare ligula. Donec justo. Nulla ligula magna, ultrices vitae, vehicula nec, semper vel, felis. Duis pulvinar molestie augue. Sed ac justo eu ante adipiscing faucibus.

Support Sales Customer Relations

History

Blog
View recent blog entries

Member for
1 week 4 days

Groups

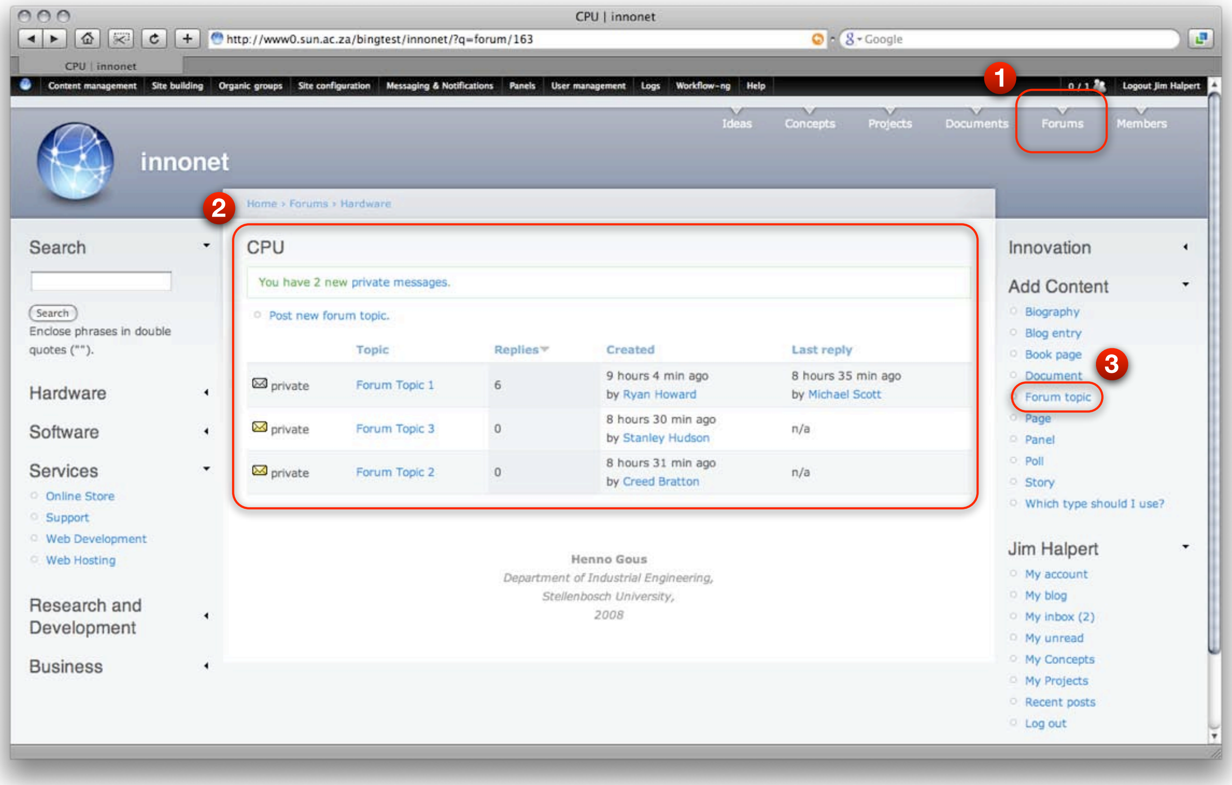
- Concept 1

Figure 87 - User biographies form pointers to tacit knowledge in the network knowledge base

Users are allowed to create a single biographic node during the registration of their system user account. This node integrates with other content within the network knowledge base, and therefore forms a pointer to a tacit knowledge source (the user) within the network knowledge base (refer to Chapter 5.2.4).

A listing of all the biographical nodes within the knowledge base may be obtained through the "Members" link in the top navigation menu (see number 1 in Figure 87). Users may update their own biographies, and browse through other biographies to get acquainted with the tacit knowledge sources in the system (number 2 in Figure 87).

d) Forum Discussions


CPU | innonet

http://www0.sun.ac.za/bingtest/innonet/?q=forum/163

Content management | Site building | Organic groups | Site configuration | Messaging & Notifications | Panels | User management | Logs | Workflow-ng | Help | Logout Jim Halpert

innonet

Ideas | Concepts | Projects | Documents | Forums | Members

Home › Forums › Hardware

Search

Enclose phrases in double quotes ("").

Hardware

Software

Services
- Online Store
- Support
- Web Development
- Web Hosting

Research and Development

Business

CPU

You have 2 new private messages.

Post new forum topic.

	Topic	Replies	Created	Last reply
private	Forum Topic 1	6	9 hours 4 min ago by Ryan Howard	8 hours 35 min ago by Michael Scott
private	Forum Topic 3	0	8 hours 30 min ago by Stanley Hudson	n/a
private	Forum Topic 2	0	8 hours 31 min ago by Creed Bratton	n/a

Henno Gous
Department of Industrial Engineering,
Stellenbosch University,
2008

Innovation

Add Content
- Biography
- Blog entry
- Book page
- Document
- Forum topic
- Page
- Panel
- Poll
- Story
- Which type should I use?

Jim Halpert
- My account
- My blog
- My inbox (2)
- My unread
- My Concepts
- My Projects
- Recent posts
- Log out

Figure 89 - Forum topics form knowledge objects in the network knowledge base

Following the “Forums” link in the top navigation menu (see number 1 in Figure 89) produces an overview of all the ongoing forum discussions in the system (refer to Chapter 5.2.5). Users may browse through the existing discussions and reply to a specific forum topic (number 2 in Figure 89) or start a discussion of their own by creating a topic from the “Add Content” menu (number 3 in Figure 89).

These forum discussions are nodes within the network knowledge base and form collaborative knowledge objects.