Wikis for Unprescribed Information Systems

Marco André Gonçalves Pinheiro
Instituto Superior Técnico
marco.pinheiro@ist.utl.pt

Abstract In the digital era, most organizations use information systems as support for activities at operational, strategic and management level. Traditionally, these systems are prescribed as a result of governance’s decisions. However, on a more meticulous evaluation, some of them are not prescribed, said emergent, which are not derived directly from the initial design. Regardless of its medium nature (electronic or not), these systems arise within the organization alongside the prescribed ones and are in part promoted by initiatives of non-manager agents. Examples of unprescribed systems and work are: annotations and notes in printed documents; post-it with information necessary to accomplish tasks; coordination and overview of the progress of a complex task through structured spreadsheets; and running logs where information is written down from different sources that are not integrated. Such systems have inherent disadvantages, for example, fail to support traceability and updating information, are not oriented to sharing information and do not support well collaboration among an large group of users. This thesis seeks to provide a model of how emergent and prescribed information and artifacts can be supported by an emerging platform. After modeling in Tropos the features of unprescribed systems of a case study and to consider a scenario of how emerging aspects arise, software architecture is generated focused on three main concepts: artifacts, data, and import/export. After that, XWiki is extended to enhance its features to support emergency resulting in an emerging platform that is subsequently validated by simulating the scenario used previously to expose the problem.

Keywords: Unprescribed Information Systems, Emergent Artifacts, Emergent Information, Wiki

1 Introduction

The current Information Systems (IS) in organizations may be classified according to which and how the technology is used by two distinct types: prescribed and unprescribed. Prescribed Information Systems (PIS) in organizations are those whose implementation is required by the top management to support the defined goals of the organization. On the other hand Unprescribed Information Systems (UIS) arise as an alternative or complement to the necessary work to support the daily tasks of the organizational actors, responding to contingencies, exceptions and daily opportunities (Orlikowski, 1996).

The goals and needs that lead users to interact with the IS may converge or diverge from the organization’s objectives. The figure 1 shows the nature of this alignment when technology and method of use are classified as emerging or prescribed. When IS (prescribed or unprescribed) are used respecting the goals and tasks prescribed by the top management, we can say that the use is prescribed and convergent. The emergent use of IS (prescribed and unprescribed) can be subdivided in two types:

Unprescribed use

Convergent

Divergent

Prescribed use

Convergent

Unprescribed IS technology

Prescribed IS technology

Figure 1 - IS classification according with technology and use
convergent and divergent. An emergent and convergent use corresponds to the emergence of new ways of using IS to achieve organizational goals, so an emergent and divergent use corresponds to a use whose intention is not necessarily aligned with the organizational goals.

2 Goals

2.1 Objective

This thesis aims at studying the relationship between prescribed and unprescribed information systems and its characteristics taking into account the co-existence of prescribed and emerging information. Accordingly, a case study with emergent scenarios is modeled in Tropos in order to capture the requirements of the emerging system and thereby obtaining a model of features and functionality intended from an emerging system.

On the other hand, aims to assess the potential of wikis as a platform to support the needs of users in the context of an emergent scenario. A wiki is then extended to close gaps and improve the support for user-development of emerging systems.

2.2 Contribution

We expect that the utilization of a digital platform designed to support unprescribed systems will increase the digital support for emergent needs and tasks taking advantage of the digital systems inherited benefits, such as support to traceability and data interoperability between prescribed and unprescribed systems, sharing information, support to manage emergent information in a structured way and support for collaboration among an large group of users.

3 State of the Art

3.1 Unprescribed Information Systems

The perspectives on organizational change shared the assumptions that the change was only episodic, rapid, radical and necessarily planned, and their drivers were (depending on perspective) the managers or the technology itself (Orlikowski, 1996). But the concept of change has evolved from an episodic event in the background to a constant activity during the life of an organization. Consequently, the existing set of perspectives has become inadequate to support this new vision.

In contrast to the deliberate change strategies, there are also the emerging ones. While the change is a deliberate creation of a new organizational pattern according to the originally conceived, the emerging change is the realization of a new organizational pattern without any specific intention or preconceptions via the actions performed by humans, even if unintentional or perceived.

The emerging non-electronic systems, adopted by users, are often improvised from artifacts known. Folders, files, stacks, folders, chalkboards, calendars, post-it notes are examples of such artifacts. In a study, two types of articles were highlighted as the most important: folder and stack. The elements of a folder have a title and are placed in a specific order (e.g. alphabetical). In a stack, elements have a sequence but are placed at the top in no particular order.

On the other hand, the emerging electronic systems, adopted by users, are often improvised using productivity software, such as spreadsheets, due to ease of use experienced by most users. The worksheets can keep and organize information on a complex task whose status is updated over time.

Electronic mail is an example of an emergent system due not to technology itself but because of the non-prescribed use. More than
a means of asynchronous communication, email has become an agenda, a manager of notes, contacts and to-do list, and even a place to store files and transferring them.

Malone identifies two basic functions provided by the articles. The first feature is the search. The agents organize documents so they can find them later. The second feature is the recall. There are artifacts visible at the top of the tables in most offices that are there in order to visually remind users to non-performed tasks. Dix et al distinguish two roles that can be staged by the same artifact: trigger and placeholder. The essential difference between them relies on that the former express that there is something to be performed and the latter one express what is need to be completed.

3.2 Wikis

Wikis are collections of linked pages that can be edited by anyone, anytime, from anywhere.

The wikis move away the paradigm of control and approach the vision of sharing, collaboration and democratize the access to information. Deliberately, complex procedures and structures were removed from the wiki model. At this level, users have the role to learn and adopt the style of using the wiki and define the way of organizing information. This flexibility allows the platform to be used for a variety of emerging needs.

Most wikis offer a common set of features: editing documents with text, tables, links, and other more complex artifacts; simple markup language; WYSIWYG editors, RSS notifications, tagging, hierarchy organization, control access, search, and the flexible paradigm of browsing documents through links, not restricting users to one application model that potentially doesn’t satisfy their needs.

Another wikis has more complex features as: scripting, extensions packages, management of objects, classes and properties.

Users can use this set of features to accomplish their needs: to remember and prioritize tasks and to recall information saved on structured wiki pages.

3.3 Case Study

The case study is an ethnography about a sales department. The unit applies efforts using information systems to make their work appear complained with the rules, procedures and prescribed goals. If fact, the employees perform other work that top management has not delegated and that was actually delegated to other departments. Arises the why an apparent compliance is needed.

Therefore employees face three challenges of self-coordination that (Cunha, 2005) pointed: (i) remember task and their status, (ii) remember information required to complete each task, and (iii) prioritize tasks. In order to overcome these challenges, users have adopted various artifacts and techniques. Figure 2 shows the strategic dependency model for describing the dependencies of employee on the emergent platform.

Though founded in different formats, the main artifacts boils down to to-do lists (and post-its), stack of documents and running logs. The table 1 shows the artifacts and their formats to support each challenge.

![Figure 2 – Challenges that leads employee to use emergent artifacts](image-url)
3.4 Scenario

In this scenario M-TEL (a fictional company), Mark (an employee of the sales force department), and Supermarkets Inc. (a client) are introduced. This department aims to carry out sales whose monthly total per customer is below a limit and other prescribed procedures as using Siebel to create and maintain essentially four types of entities: sales, sales opportunities, tasks and contacts from customers.

Mark receives and email from client with a request for quotation. According with the rules, Mark should transfer the customer to fields force department because he knows in advance that the quote will reach such limit imposed by management. So it avoids using Siebel to record the pending task. To remember the task, he put the email printed on the top of a stack that contains other requests for dealing. Another way to deal with unprescribed tasks is to write down requests in post-it and put them on desktop.

To prioritize tasks, Mark decides to relocate post-it according to his perceived importance. On the other hand, he removes documents from stack that are no longer useful for following tasks and can be archived or deleted. The stack is thus renewed and cleaned of unnecessary items for the rest of the working. Unfulfilled requests and priority documents appear at the top.

After choose and task to fulfill, Mark starts processing the request for quotation that appears at the top of the stack. Mark must create a new sales opportunity on Siebel according with procedures, but cannot describe it with much detail because this request should be redirected to fields force department. To remember information to complete the order if client ask for, Mark annotates the document with the account and opportunity number that appear in Siebel creating a link between the non-technological physical artifact and the digital entity. Put the document in a file with other documents organized by customer account.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Artifact</th>
<th>Artifact Format</th>
<th>Nuances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remember tasks</td>
<td>To-Do List</td>
<td>Page of a notebook</td>
<td>• Scratch when complete;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simple sheet of paper</td>
<td>• One item per line;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page of a calendar</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Post-it</td>
<td>• One for each task;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Discard when complete;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Position of the artifact;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inbox</td>
<td>• Inability to take notes along the information to which they refer;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Users report difficulty of use (in order of recall tasks and information); there was fear of forgetting something important;</td>
</tr>
<tr>
<td>Remember information</td>
<td>To-Do Stack</td>
<td>Email marked with different colors that simulate stacks</td>
<td>• View and separate documents to refresh the memory of that has to do for the rest of the day;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Printed documents with notes</td>
<td>• Collection of documents and information from different sources: email, phone calls with customers, vendors or representatives</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• operational, prescribed IS, among other;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Ability to make notes on paper and along the information referred to;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Each document is oriented to the task, since the document itself can be the area for task support (and not just access to information)</td>
</tr>
<tr>
<td></td>
<td>Folders or Dossiers</td>
<td>Documents organized by customer account</td>
<td>• Tabs;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Organized by customer account;</td>
</tr>
<tr>
<td></td>
<td>Running log</td>
<td>Page of a notebook</td>
<td>• Not organized;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simple sheet of paper</td>
<td>• Not correlated information on the same page;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spreadsheet</td>
<td>• Organized;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Tables paradigm;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• A spreadsheet by subject</td>
</tr>
<tr>
<td></td>
<td>Post-it</td>
<td></td>
<td>• Correlated information on the same sheet</td>
</tr>
<tr>
<td>Prioritize tasks</td>
<td>To-do list numbered</td>
<td></td>
<td>• Numbering</td>
</tr>
<tr>
<td></td>
<td>Post-it positioned in a particular place</td>
<td></td>
<td>• Positioning of the artifact</td>
</tr>
</tbody>
</table>

Table 1 – Artifacts and their formats to support each challenge.
4 Solution

4.1 Overview

After problem analysis, three topics emerged: artifacts, data, and import/export processes.

The artifact concept covers the model of user interaction with data and the work practices tailored to the needs of users. The data concept comprises the topics related with emergent and prescribed information and operations on it. On the one hand, artifacts use data and represent it in a specific format. On the other hand, the import/export process replicate internally entities that reside in external systems, and copy entities created in the emerging platform to outside.

The objectives of the emerging platform stands out for its ability to work with data from different external sources/systems and worked/used in accordance with the practices identified.

4.2 Quality Attributes

There are four quality attributes: (i) ease of modification, (ii) interoperability and (iii) ease of use, and (iv) security. In detail:

- Easy to modify the artifacts, data, and data type at runtime by end-user;
- Easy to modify the types of artifacts and connectors available during implementation;
- Securing access to emergent artifacts and data;
- Interoperability with external data sources, including import and export of data in systems prescribed;
- Easy the use of the artifacts proposing useful features that appear in the physical format of the artifacts;
- Usability of the data by providing operations over it.

4.3 Software Architecture

On the software architecture of the emerging platform there are three main: module decomposition, which modules each module uses, and the components and connectors viewtype.

Regarding the decomposition style, figure 3 shows that the emerging platform decomposes into three modules related with the topics already raised in section 4.1. In module “data source” the decomposition isolated what is dependent on technology integration and external source – the “connectors” module – from what is not – “import” and “export” modules. This ensures that it is easy to modify or implement new connectors. In the last level of decomposition of the module “data source”, each module is a well-defined step in import/export, such as “typification” and “loading” modules. To meet the ease of modifying data types at runtime by end-users, the module “data” are decomposed into, besides other, in the “configuration data type” module. The module “operations” allows easy modify and interact with data.

![Figure 3 – One level decomposition.](image)

Regarding the uses style, the module “data” are used on two fronts: first by module “artifacts” and moreover the module “data source”. The view is directed more toward the mechanism of import/export and the identification of which modules are used by. The “mapping” module is responsible for mapping each value in a property’s data type. The module “loading” (in import) and module “data collector” (in export) use the module
“operations”. This module encapsulates the CRUD operations on data.

Regarding the components and connectors, as this is a web application arises two components that run on a client browser: “artifact” component and “configuration import/export” component. The “artifact” component consumes and produces data along the interactions with the user. The “configuration” component produces the configuration object and communicates with the engine in order to configure it for its execution. As figure 4 depict, on the server side there is a data repository. However, both artifacts and import/export engine do not communicate directly with the repository. There is a component called “data API” which contains the interface to domain hiding domain particularities and communication to repository performed through SQL. This component facilitates the modification and use of data.

The graphical WYSIWYG editor lets you change the content of articles including artifacts getting immediately the visual result. In addition, the menu allows you to add artifacts by inserting macros.

The XWiki allows the user to manage objects, classes and properties. This aspect is very important in the architecture developed for the emerging platform.

The programmable aspect comes broadly in XWiki and allows users to execute scripts in documents through the groovy and velocity syntaxes and using the APIs that manipulate objects and classes.

The XWiki platform can be extended by:

- Writing scripts in wiki pages;
- Writing applications (set of wiki pages with an application logic);
- Writing modules in JAVA that offer a coherent set of services;
- Writing new skins or extending existing ones;
- Extending existing Service APIs when they provide extension points
- Extending and configuring the WYSIWYG editor.

5 Implementation

5.1 Technology

In developing the emerging platform, various technologies are used: MySQL, Jetty, Google Web ToolKit (GWT), SmartGWT, and XWiki. The XWiki is a second generation wiki, i.e. a platform for developing web applications using the collaborative wiki paradigm. It has relevant features as an emerging platform and it is extended in order to improve support for emerging systems. Thus, it is worth mentioning the following aspects in the context of emerging and state the degree of extensibility.

The XWiki allows the user to manage objects, classes and properties. This aspect is very important in the architecture developed for the emerging platform.

The programmable aspect comes broadly in XWiki and allows users to execute scripts in documents through the groovy and velocity syntaxes and using the APIs that manipulate objects and classes.

The XWiki platform can be extended by:

- Writing scripts in wiki pages;
- Writing applications (set of wiki pages with an application logic);
- Writing modules in JAVA that offer a coherent set of services;
- Writing new skins or extending existing ones;
- Extending existing Service APIs when they provide extension points
- Extending and configuring the WYSIWYG editor.

5.2 Implementation

In terms of artifacts, table 2 summarizes the technical and functional aspects of three topics: (i) user interface in normal mode and in-line edit mode that toggles between navigation features and/or editing, (ii) XWiki macros are used to enter artifacts into documents since they are transformed into HTML and JavaScript during rendering, (iii) domain that serves to keep the artifact state that is independent of it, i.e. the same entity state can be rendered in any artifact of the same type.

Regarding the data, the meta-model supports the definition of entity types with attributes and relationships between types and then instantiate them. The XWiki allows the user to
perform these operations at run time through the user interfaces or APIs. The extension to the document editor allows the user to choose a data type and browse its instance taking into account the links between them. During this navigation is possible to choose the attributes to be included in the document. These attributes are inserted into the document through the reference artifact.

Concerning the concepts of connectors, importing and exporting, there is a user interface where you can configure each aspect and step of the process: (i) type of connector such as CSV file, (ii) configurations inherent of the type connector such as the delimiter and the lines to ignore, (iii) page that hosts the entities, (iv) transfer strategy in accordance with table 1, (v) name of the attributes that correspond to the moments of last modification, import and export, (vi) mapping between the attributes of the internal structure and sequence of values in the external format. During the mapping you can map directly an attribute to a value or another entity imported before using input values to find it.

### 6 Validation

Validation is done through the configuration and use of emerging platform in order to support the scenario described in Section 3.2. The validation is divided into three situations, each related to one of three major challenges that lead to the appearance of emerging systems.

#### 6.1 Remember Tasks

Mark receives and email from client with a request for quotation. According with the rules, Mark should transfer the customer to fields force department because he knows in advance that the quote will reach such limit imposed by management. So it avoids using Siebel to record the pending task.

Thus, Mark authenticates to the emerging platform and after opening the MTEL workspace creates a new page. He copies the peripherals and services required for the document and save it with the same title as the email. So the title becomes an implicit reference to link document and email.

In order to remind him of the task, it places the document in the top of the stack localized at workspace. Thus, even in the context of the

<table>
<thead>
<tr>
<th>Artifact</th>
<th>User Interfaces – Features</th>
<th>Macros</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Show content</td>
<td>Enable inline edit mode of artifacts embedded in the document (Modification of the contents through normal editing mode)</td>
<td>Content (text and artifacts)</td>
</tr>
<tr>
<td>Workspace</td>
<td>Show placed artifacts</td>
<td>Insert, delete and relocate artifacts by dragging and dropping</td>
<td>—</td>
</tr>
<tr>
<td>Panel</td>
<td>Show and hide panels with artifacts</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Stack</td>
<td>Browse the stack from top to bottom and vice versa click an object to view it</td>
<td>Insert, delete and reorder documents</td>
<td>Macro name: Parameters:</td>
</tr>
<tr>
<td>Post-it</td>
<td>Show content</td>
<td>—</td>
<td>Macro name: Parameters:</td>
</tr>
<tr>
<td>Reference</td>
<td>Show value of referenced attribute</td>
<td>Modify value of referenced attribute</td>
<td>Macro name: Parameters:</td>
</tr>
<tr>
<td>Annotation</td>
<td>View, add and delete notes from the document</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Tag</td>
<td>View, insert and delete tags from the document</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 2 – Technically and functional aspects of artifact in emerging platform.
newly created document he choose the stack and press the button add through the extended functionality of the panel. The document displayed is then inserted at the top of the stack chosen (figure 5). Mark wants to insert the price list document in stack so as to facilitate the preparation of the budget when he returns to this theme later. So opens the workspace in inline editing mode enabling editing features of artifacts. He can now add documents to the top of the stack and rearrange them.

Another way to deal with unprescribed tasks is to write down requests in post-it and put them on workspace (figure 6). During the inline editing is possible adding artifacts through the add button. Select the type of artifact and configure the settings, in this case, the post-it and its contents. Still in the inline editing mode he can relocate any artifact including the post-its.

6.2 Prioritize Tasks

To prioritize tasks, Mark decides to relocate post-it according to his perceived importance. On the other hand, he removes documents from stack that are no longer useful for following tasks and can be archived or deleted.

Mark uses the stack editing features to reorder documents with the drag-and-drop feature and remove unnecessary ones. In this case, the documents when being removed from the stack is not needed to be archived in folders or files because the document exists on its own regardless of its location on stacks.

As indicated previously, the edit mode of the workspace allows relocate the post-it in a sequence according to a criterion set by the user consciously or unconsciously.

Thus, unfulfilled requests and priority documents appear at the top.

6.3 Remember Information

After choose and task to fulfill, Mark starts processing the request for quotation that appears at the top of the stack.

After he double-clicks on the document to open it, notes prices next to each peripheral. To do this, select the text referring the peripheral and with a keystroke arises the annotation artifact where he can write the price. Mark writes in an email response from the notes made – there is an option to view all notes in the document.

Mark must create a new sales opportunity on Siebel according with procedures, but cannot describe it with much detail because this request should be redirected to fields force department. To add emergent information to the sales opportunity entity, Mark first imports the entity from Siebel to the platform through import/export mechanisms and then edits the entity in platform.

Mark adds in the document the sales opportunity as a reference and, in order to find the document later, add tags to it with the account and opportunity number that appear in Siebel (figure 7)

![Figure 5 – A Stack of documents.](image)

Figure 5 – A Stack of documents.

![Figure 6 – A Post-it.](image)
7 Conclusions

7.1 Contributions

The main objective of this dissertation is to provide users a platform as a basis for developing a system able to meet their emerging needs.

The main contributions are:

- Modeling in Tropos artifacts features and rationale use – to-do lists, post-it, stack of documents, email and spreadsheets;
- Development of a specific software architecture for emerging platforms;
- Integrate wikis with emergent systems features;
- Provide users with a flexible platform for the development of emerging systems and handling of emerging information.

7.2 Future Work

In future work, the following topics would be covered:

- Analyze the benefits of hiding the metadata from users or leave full control over them;
- Extended integration with data sources allowing to perform operations on the domain and replicate them on external systems, with the authorization of users;
- References with bidirectionality can identify documents and artifacts that relate to a particular entity finding correlate documents;
- Feature to store import/export settings and run them automatically according to scheduling rules.

8 References


Casual Organization of Information. In CHI, pp. 627-634.
