A FRAMEWORK FOR ENCAPSULATING BEST BUSINESS PRACTICES FOR ELECTRICITY SUPPLY INDUSTRY INTO GENERIC PATTERNS

C. Rolland*, G. Grosz*, P. Loucopoulos, S. Nurcan***

* Centre de Recherche en Informatique
Université Paris 1 - Panthéon - Sorbonne
90, rue de Tolbiac 75013 Paris, France
{rolland, grosz, nurcan}@univ-paris1.fr

** Department of Computation, UMIST
P.O. Box 88, Manchester M601QD, U.K.
pl@sna.co.umist.ac.uk

Abstract : *This work presents a Web based tool implementing a framework that is being used to fulfil one of the main objectives of the ESPRIT project ELEKTRA*[1] *: the discovery of the generalised patterns of change management for re-using them in similar settings in other electricity companies. The term 'pattern' refers to knowledge that may be repeatable from one situation to another, and shareable by different users. The issues of defining (a) the structure of a knowledge repository of good business practices for managing change in organisations and (b) a Web based tool for providing a means for retrieving and accessing this knowledge are developed in this paper. The tool allows to introduce, in a company involved in a change management, a common repository of reusable knowledge easily accessible and customisable by the stakeholders.*

Keywords : *Generic Pattern, Reuse Process, Change process, WWW*

1. Introduction

Recent years have witnessed an increasing interest in the use of patterns within the software development community and in particular by those advocating object-oriented approaches and re-use (Coad, 1992), (Beck, 1997), (Buschmann, *et al* 1996), (Coplien, Schmidt, 1995), (Gamma *et al*, 1994), (Vlissides *et al*, 1996), (Hay, 1996), (Fowler, 1997), (Rolland, Plihon, 1996), (Harmsen *et al*, 1994). These efforts have in common their attempt to exploit knowledge about *best practice* in some domain. Best practice knowledge is constructed in 'patterns' that are subsequently used as the starting point in the domain activities. Different studies have shown that it seems possible that generic knowledge can be exhibited for the modelling of change in the *Electricity Supply Industry* (ESI) sector, e.g. (Yajima,1997).

In ELEKTRA, we propose to describe the reusable knowledge encapsulated in patterns in terms of the EKD (Enterprise Knowledge Development) methodology concepts (i.e. enterprise goals, enterprise processes etc.). EKD is an approach that provides a systematic and controlled way of analysing, understanding, developing and documenting an enterprise and its components by using Enterprise Modelling (Loucopoulos *et al*, 1997), (Bubenko et al. 1997), (Rolland *et al*, 1997a), (Rolland et al, 1997b), (Rolland *et al*, 1998a). In the context of this project, the patterns will be specific to the ESI sector and will not be applicable to any other sector. The framework however, for organising these patterns, is independent of any application and could potentially be used in other domains. The rationale is that all concepts defined and used within this framework are based on the EKD method which is totally domain independent.

An important aspect of the ELEKTRA project concerns the dissemination of best business practices within the ESI sector companies. To this end, we encapsulate the knowledge describing the best business practices within a pattern repository and propose a tool called 'Electronic Guide Book' that allows to retrieve and access this knowledge over the Web. The 'Electronic Guide Book' can either be used in a standalone manner or in connection with other tools supporting the change management. In the latter case, the 'Electronic Guide Book' can be seen as an on-line powerful and intelligent help that can be activated by an EKD engineer at any time while using other change management support tools. In the former case, the 'Electronic Guide Book', that indeed can be used through the Internet, acts as an hypertext encyclopaedia that allows any EKD engineer to gain knowledge on the change management, as well as on the way these knowledge shall be re-used.

Section 2 discusses the notion of a pattern, proposes a pattern template and the corresponding HTML representation. Section 3 describes the structure of the pattern repository allowing an easy navigation and retrieval of patterns and illustrates the HTML representation of the indexing hierarchy used in the pattern repository. Section 4 describes the process of retrieving generic patterns through HTML hyper-links navigation and presents the 'Electronic Guide Book' through a scenario of use.

2. The notion of a pattern and its HTML representation

There are three key ideas concerning patterns that can be found in the literature (Coad *et al*, 1996), (Alexander *et al*, 1977), (Alexander, 1979), (Gamma *et al*, 1994). First is the fact that a pattern relates a *problem* to a *solution* to this problem. The second point draws our attention to the fact that what is important in pattern identification is not so much to recognise the commonalties among candidate elements but to identify the discriminant criteria. Finally, the genericity may be at different levels: (i) A pattern can be used for different applications in a specific domain repeatable in different organisations/settings (application independent but domain specific). Example: generic patterns for ESI sector ; (ii) A pattern can be used in

[1] This work is partially supported by the ESPRIT project ELEKTRA (N°22927) funded the by the EC in the Framework IV programme.

different domains (domain independent pattern). Example: generic patterns for goal refinement. For the pattern to be generic and reusable, the pattern structure should be able to state (a) that a given problem exists within a stated range of applications and (b) that in the given context, a given solution solves the given problem.

2.1. The pattern template

The pattern 'template' is a predetermined structure according to which the generic knowledge has to be represented. Clearly, there is a need to make a difference between the *reusable knowledge* contained in a pattern and its *description*. The former is represented using an EKD model fragment in our framework. We call it the *body* of the pattern. The latter, namely *descriptor,* aims to describe the context in which the body of the pattern can be reused (situation) and the problem it can resolve (usage intention). Therefore, a pattern is completely defined with its descriptor and its body (see figure 1).

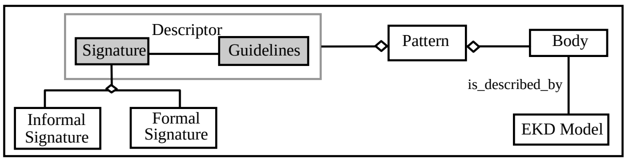


Figure 1 - The pattern template

2.1.1. The body

We propose a refinement of the conceptual modelling approach by using the EKD concepts to describe the body of a pattern (Loucopoulos *et al*, 1997). The structuring framework for the generic patterns is domain independent, the only dependency being in its use of EKD descriptions. Figure 8 shows the body of a pattern describing dependencies between roles involved in the electricity consumption measurement.

2.1.2. The descriptor

Our proposal is to have a pattern descriptor defined as an aggregation of a signature and guidelines (figure 1). The former describes in which situation and for which problem it is relevant to reuse the body of the pattern whereas the latter are recommendations on the way the body of a pattern can be reused. A *signature* aims at describing the characteristics of a pattern, when it can be used, why, etc.. It has a formal part and an informal part as shown in figure 1. As we will develop in sections 3 and 4, *formal signatures* are used in the reuse process in order to retrieve patterns which are appropriate for a given situation having a given usage intention in mind. Some additional requirements for describing patterns are encapsulated in the *informal signature*. We have chosen to combine a *faceted approach* with a *contextual approach* (Grosz *et al.*,1997) to describe the *formal signature* which is therefore composed of a *situation* part -the context- and a *usage intention* part -the problem.

- The *situation* precisely describes the applicability conditions which must hold for reusing the generic pattern. It comprises two facets :
  - The *type*: the type of the pattern (Actor/Role, Role/Activity, Object, Rule, Goal, Change process patterns) (see section 2.2).
  - The *domain*: it describes the activity domain for which the pattern is applicable to (Customer servicing, restructuring, etc.).
- The *usage intention* expresses the goal to be achieved by the use of the generic pattern. It has two facets:
  - A *verb* (invoice, change, ...), and
  - A *set of parameters* described as a set of sub-facets: source, target and manner. Each parameter plays a different role with respect to the verb, some of them having sub-types (e.g. target has two sub types : object and result).
  - – The *target* designates what is affected by the usage intention. We distinguish two types of targets: objects and results. As opposed to *objects*, the *results* are affected by the usage intention. They do not exist prior to the usage intention.

    Measure $_{\text{verb}}$ *(electricity consumption)*$_{\text{result}}$
  - – The *source* identifies the origin of what is affected by the usage intention.

    Measure$_{\text{verb}}$ (*electricity consumption*)$_{\text{result}}$ *(from meter reading)* $_{\text{source}}$
  - – The *manner* parameter is used to express in which way the usage intention is achieved.

    Measure$_{\text{verb}}$ (*electricity consumption*)$_{\text{result}}$ (*from meter reading*)$_{\text{source}}$ *(automatically)*$_{\text{manner}}$

2.2. Pattern typology

2.2.1. Product patterns

Since we have chosen to represent the bodies of patterns as EKD models fragments, we adopted the EKD typology of models for patterns. Therefore, the body of a pattern can be either a goal pattern represented with the EKD goal model concepts or a business process pattern represented with the EKD business process model concepts. A business process pattern is specialised in turn into actor/role pattern, role/activity pattern, object pattern and rule pattern. All these patterns describe ESI structural models and are referred to as *product patterns*. However, EKD addresses both the description of the business processes (and their associated goals) and the description of the change process itself. The former leads to product patterns whereas the latter introduces the needs for another type of pattern called, *change process patterns*.

2.2.2. Change process patterns

Our proposal is to use an extended goal model as a means to represent the body of generic change process fragments. Following the semantics of an EKD goal graph, a change process pattern is represented as a goal hierarchy using the AND, OR and AND/OR connectors. In order not to confuse between company business goals and the company change goals, we decide to rename for the latter the concept of 'goal' as 'change intention'. The leaves of this goal hierarchy are called operationalisable change intentions which correspond to intentions that do not require any further decomposition, it means that their realisation can be expressed in terms of one or several product patterns (i.e. a set of EKD models).

2.3. The HTML representation of a pattern

Figure 8 shows an example of an HTML page representing a pattern with respect to our pattern template. The body of the pattern which is an actor-role model is designed in such a way that, according to our belief, it is applicable to all ESI organisations dealing with the problem of measuring electricity consumption.

3. The pattern repository structure and its HTML representation

Composed patterns are meaningful. The first possibility is to define a pattern as an aggregation of other patterns, i.e. to compose complex patterns using more elementary patterns. The drawback of this solution is that it hinders a flexible reuse process. A more flexible solution consists of (i) keeping patterns at the knowledge level in the repository, and (ii) expressing their possible composition through an indexing hierarchy (meta-knowledge) level. This is the option that we propose (Rolland *et al*., 1998b).

3.1. Relationships among patterns and their formal signatures

Since a formal signature is intentional, the proposal is to express the relationships among patterns using intention connectors. We identify three types of connectors, AND, OR (exclusive OR) and AND/OR (inclusive OR). This way of expressing the possible combinations of patterns leads to a hierarchical representation of the domain knowledge which is intentional as it is expressed with domain goals (i.e. usage intentions of formal signatures). Our belief is that this representation eases the retrieval of relevant patterns for a given situation having a given usage intention in mind.

3.2. Repository organisation

The repository of the ESI patterns is composed of : (i) the patterns part (defined at the knowledge level), and (ii) the indexing hierarchy part (defined at the meta-knowledge level). The patterns part represents the ESI knowledge for a single problem. The descriptor of the pattern is an expression of the problem whereas the body is the reusable solution provided to this problem. The indexing hierarchy part describes this knowledge in terms of the ESI domain goals (usage intentions) that can be fulfilled by reusing the patterns with the corresponding applicability situations. The hierarchical organisation of the pattern formal signatures helps structuring the problem in intentional terms that are (or should be) easily understood by the domain experts because they describes their context of work and their problem.

In figure 2, the top level usage intentions figures out what is the overall objective of the ELEKTRA project, namely to 'Manage change in the ESI sector' and tells us that there are two problems in managing the change which are supported by pattern based solutions in the repository: 'Understand ESI models' and 'Manage the change process'. While browsing through the indexing hierarchy associated to the former usage intention, the possible paths lead to a set of product patterns, whereas while browsing through the hierarchy associated to the latter, the possible paths lead to change process patterns. The leaves of these change process patterns make explicit references to product patterns.

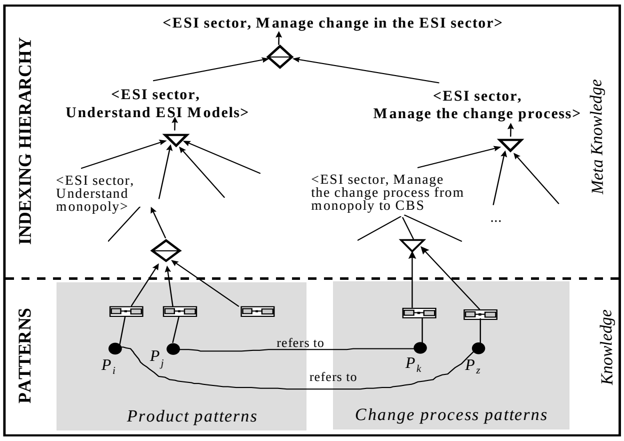

INDEXING HIERARCHY
PATTERNS
<ESI sector, Manage change in the ESI sector>
<ESI sector, Understand ESI Models>
<ESI sector, Manage the change process>
<ESI sector, Understand monopoly>
<ESI sector, Manage the change process from monopoly to CBS
...
Meta Knowledge
Knowledge
refers to
refers to
P_i
P_j
P_k
P_z
Product patterns
Change process patterns

Figure 2 - The pattern repository structure

3.3. The HTML representation of the pattern repository

Figure 3 shows examples of two HTML pages representing respectively an AND-ed node (background window) and an AND/OR-ed node (foreground window) in the indexing hierarchy.

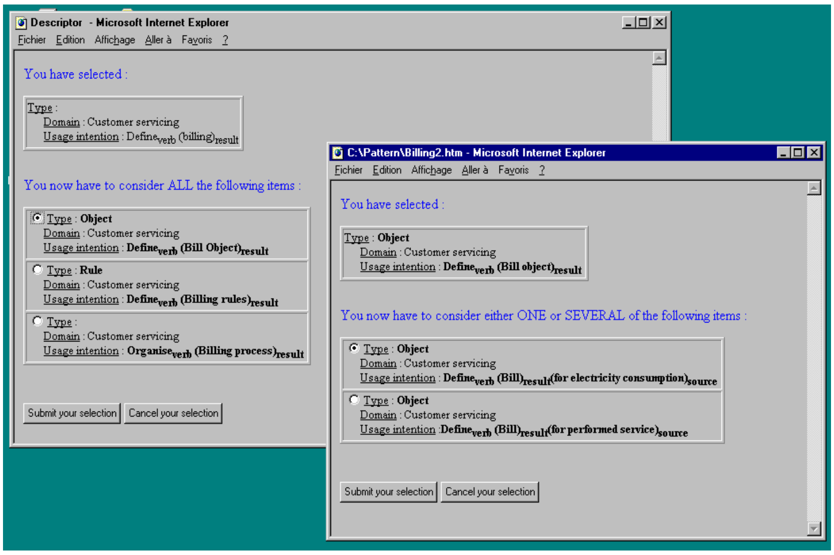
4. The 'Electronic Guide Book' for retrieving patterns

As illustrated in figure 4, the process of reusing a generic pattern is a three steps process which consists of : (1) the retrieval and selection of the generic patterns from the repository of generic patterns, (2) the storage of the selected generic patterns in the work space, and (3) the customisation/expansion/refinement of the generic pattern. The 'Electronic Guide Book' that is described in this section handles the first step of this process and proposes browsing and querying facilities tailored to the pattern repository.

4.1. The architecture of the 'Electronic Guide Book'

Our approach emphasises the integration of Web based and hypertext based technologies for facilitating the reuse of generic knowledge. The hypertext technology helps users by streamlining access to and providing rich navigational features around related information, thereby increasing user comprehension of information and its context (Bieber, 1995). Used in the context of reuse, hypertext facilities permit to access to generic knowledge in a simple manner, providing different levels of details, allowing to combine textual descriptions along with graphical ones. The Web based technology offers accessibility, portability and eases the distribution. There are many potential approaches for integrating Web servers with knowledge bases. Our approach uses conjointly traditional CGI, applets and client scripting written as Visual basic scripts and Java scripts.

The ‘Electronic Guide Book’ is composed of two inter-related modules: the *Repository Browser* which runs at the client site and *Pattern Search Engine* which runs at the server site (figure 4). The *Repository Browser* is the interface provided to the domain expert, it consists in a set of HTML forms which allows him/her to interact with the ‘Electronic Guide Book’ and to access the pattern repository through either Internet or locally. These HTML forms can be displayed by any traditional web navigator (form-compatible Web browser), such as Microsoft Internet Explorer® or Netscape®, among others. We embed, directly into Repository Browser HTML forms, VBScript® code represented as standard ASCII text. VBscript is a simple subset of Microsoft Visual Basic for Applications (VBA) (Brophy, 1996). This code runs entirely within the Repository Browser at the client site and aims at helping CGI applications by pre-processing user’s actions and information entered through the HTML forms. Some client-side applications, for instance a graph browser written in the Java language, are downloaded through the network from the Web server. The *Patterns Search Engine* is composed of a Web server, a set of CGI applications, and the pattern. repository. It runs at the server site. The Web server is responsible for all exchanges between the client(s) and CGI applications (arrows labelled 1.1 and 1.2 in figure 4) that run on the server. It receives request from client(s) in the form of formal signatures elements and sends the resulting Web pages that are generated "on the fly" from the pattern repository (arrows labelled 1.1 and 1.6 in figure 4). We chose to use the software called "Website ®" for the development of our Web server. Website ® provides a fully graphical "property sheet" administration tool that makes web server construction and configuration very easy to use, it also offers an integrated toolkit for developing CGI applications using Visual Basic. The Web Browser and Server communicate each other by using the Hypertext Transfer Protocol (HTTP). The interactions between the pattern repository and the Web server are handled by a set of Common Gateway Interface (CGI) applications. The CGI applications are organised in two modules: the Dynamic Form Processing module and the Query module. The first one transforms repository output data, embeds them into HTML forms (arrows labelled 1.4 and 1.5 in figure 4) and returns them to the Web Server. The second module manages all queries that emanate from the domain expert. This module interacts directly with the pattern repository by transforming users queries into SQL queries (see arrows 1.2 and 1.3 in fig. 4). The pattern repository is implemented with Microsoft Access ® RDBMS. It is structured according to the formalism presented in section 2. Thank to the multimedia ability of Microsoft Access, images are directly stored into the database.


Descriptor - Microsoft Internet Explorer
Fichier Edition Affichage Aller à Favoris ?

You have selected :

Type :
Domain : Customer servicing
Usage intention : Define$_{verb}$ (billing)$_{result}$

You now have to consider ALL the following items :

◉ Type : **Object**
Domain : Customer servicing
Usage intention : **Define$_{verb}$ (Bill Object)$_{result}$**

○ Type : **Rule**
Domain : Customer servicing
Usage intention : **Define$_{verb}$ (Billing rules)$_{result}$**

○ Type :
Domain : Customer servicing
Usage intention : **Organise$_{verb}$ (Billing process)$_{result}$**

Submit your selection
Cancel your selection
C:\Pattern\Billing2.htm - Microsoft Internet Explorer
Fichier Edition Affichage Aller à Favoris ?

You have selected :

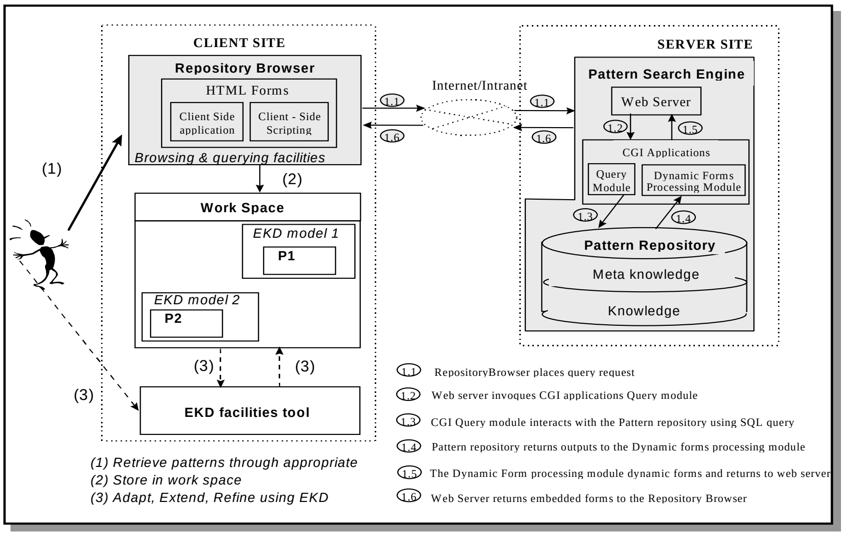
Type : **Object**
Domain : Customer servicing
Usage intention : **Define$_{verb}$ (Bill object)$_{result}$**

You now have to consider either ONE or SEVERAL of the following items :

◉ Type : **Object**
Domain : Customer servicing
Usage intention : **Define$_{verb}$ (Bill)$_{result}$(for electricity consumption)$_{source}$**

○ Type : **Object**
Domain : Customer servicing
Usage intention :**Define$_{verb}$ (Bill)$_{result}$(for performed service)$_{source}$**

Figure 3 - Examples of HTML pages representing the indexing hierarchy



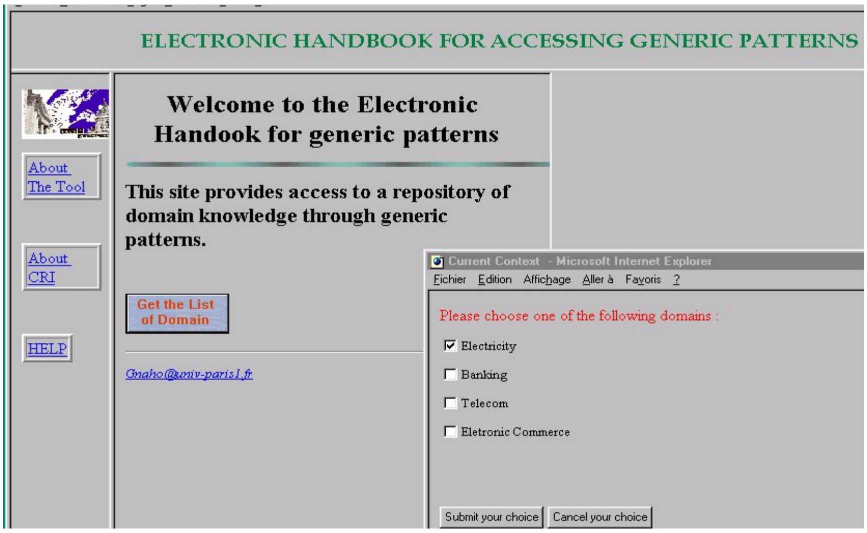
(1) Retrieve patterns through appropriate
(2) Store in work space
(3) Adapt, Extend, Refine using EKD

1.1 RepositoryBrowser places query request
1.2 Web server invoques CGI applications Query module
1.3 CGI Query module interacts with the Pattern repository using SQL query
1.4 Pattern repository returns outputs to the Dynamic forms processing module
1.5 The Dynamic Form processing module dynamic forms and returns to web server
1.6 Web Server returns embedded forms to the Repository Browser

Figure 4 - Global Architecture of the ‘Electronic Guide Book’

4.2. Using the ‘Electronic Guide Book’

We shall now detail the different windows that the domain expert goes through while using the ‘Electronic Guide Book’. For the sake of readability, the presentation follows an imaginary scenario of use. The background window of figure 5 presents the home page of the ‘Electronic Guide Book’. The main suggestion is to get the list of domain for which generic patterns have been defined within the repository. After the domain expert had clicked on the button, a query is sent to the Patterns Search Engine to retrieve the list of domains for which patterns have been defined. The result is sent to the Repository Browser which displays the foreground window of figure 5. The domain expert has now to select the one he/she has interest in. Then, as shown in the



ELECTRONIC HANDBOOK FOR ACCESSING GENERIC PATTERNS

Welcome to the Electronic Handook for generic patterns

This site provides access to a repository of domain knowledge through generic patterns.

Gnaho@univ-paris1.fr

Current Context - Microsoft Internet Explorer

Please choose one of the following domains :

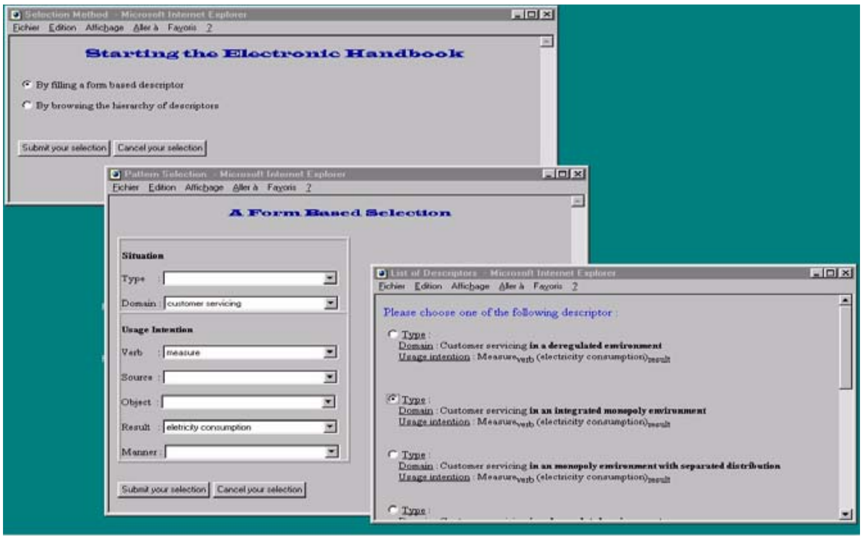
☑ Electricity

☐ Banking

☐ Telecom

☐ Eletronic Commerce

Figure 5 - The home page of the ‘Electronic Guide Book’



Selection Method - Microsoft Internet Explorer

Starting the Electronic Handbook

● By filling a form based descriptor

○ By browsing the hierarchy of descriptors

Pattern Selection - Microsoft Internet Explorer

A Form Based Selection

Situation

Type :

Domain : customer servicing

Usage Intention

Verb : measure

Source :

Object :

Result : eletricity consumption

Manner :

List of Descriptors - Microsoft Internet Explorer

Please choose one of the following descriptor :

○ Type :
Domain : Customer servicing **in a deregulated environment**
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$

● Type :
Domain : Customer servicing **in an integrated monopoly environment**
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$

○ Type :
Domain : Customer servicing **in an monopoly environment with separated distribution**
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$

○ Type :

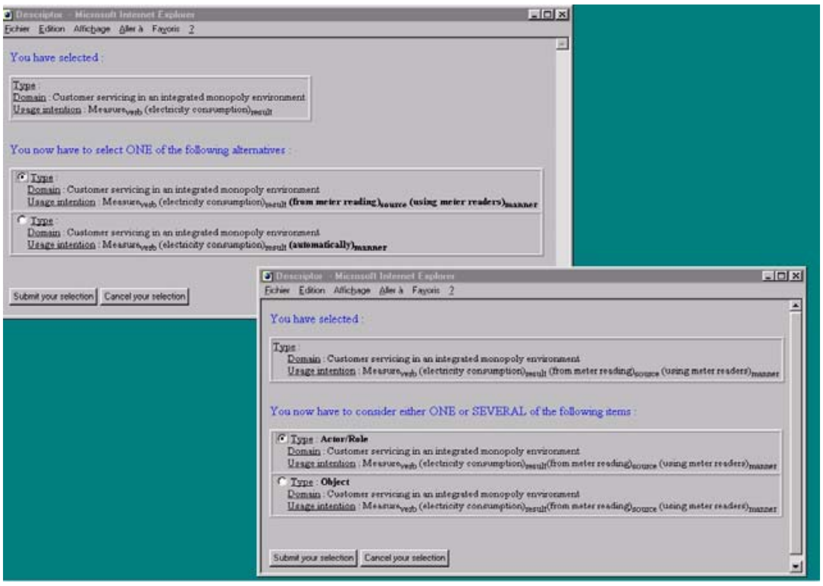
Figure 6 - Submitted query and associated matching formal signatures

background window of figure 6, two options for querying or browsing the pattern repository are proposed. Assume the domain expert selects the query facility, he/she is now asked to fill a form based on the formal signature structure. For each field, a list of possible values is accessible. Selecting an item from this list guarantees the conformity of the values towards those stored in the repository. The domain expert fills the properties he/she knows the value of. For instance, he/she may fill the property ‘domain’ = ‘*customer servicing*’ and describe partially its usage intention, ‘*Measure*$_{verb}$ (*electricity consumption*)$_{result}$. After the form has been submitted (click on the ‘submit your choice’ button), the Pattern Search Engine searches within the repository for the formal signatures that match the properties that were given. The search through the indexing hierarchy is performed in a top-down manner. In order to minimise the number of retrieved formal signatures for the sake of guiding the domain expert, the search engine only retrieves the most general formal signatures that match the query. The foreground window of figure 6 displays the resulting list of formal signatures. All elements that complete the initial elements of the query are displayed in bold. The domain expert is asked to select one of these retrieved formal signatures. Based on his/her selection, the Pattern Search Engine continues to descend the indexing hierarchy to retrieve more specialised formal signatures that refine the selected one as shown in the background window of figure 7. In our example, two are matching. Because these two are alternative, the domain expert is asked to select one of them. Assume, he/she selects the first one. The next steps consist in descending the hierarchy until product or change process patterns are found. The next step, shown in the foreground window of figure 7, leads the Pattern Search Engine to reach leaves of the hierarchy.

Assume the domain expert selects the first item of the list, the associated product pattern is then displayed as shown in figure 8 with all its components (i.e. formal signature, informal signature, guidelines and body). The domain expert has the ability to display the example(s) associated to this pattern in a different window. If this pattern satisfies his/her expectations, he/she can download the body of the pattern into his current work space. At this point, he/she can choose to stop the retrieval because he/she found what he/she was looking for. Otherwise, he/she has the possibility of continuing the retrieval thanks to the ‘what to do next’ button. Clicking on this button will lead the tool to propose the pending options that were listed for this query. In our example, this would first lead the tool to suggest the study of the ‘object’ product pattern (the second item in the foreground window of fig. 7).

5. Conclusion

A critical factor in being able to share best business practice (including best practice for system development) is the appropriate reuse of existing ‘chunks’ of best practice. Patterns provide the mechanism for achieving this. The generic patterns will have to be built by empirical observation and tested on a range of examples within case studies. Prior to discovering patterns however in these case studies, there is a need to establish the framework for maintaining and using the knowledge pertinent to the patterns. This paper has attempted to provide first a framework for defining generic patterns for the Electricity Supply Industry (ESI) sector and second the associated ‘Electronic Guide Book’. For engineers involved in change management, the ‘Electronic Guide Book’ represents an opportunity of reusing already tested generic knowledge in similar settings, by customising them with respect to the particularities of the studied organisation. Both the patterns framework and the tool are independent of any application and could potentially be used in other domains. The 'Electronic Guide Book' is built over the web and provides a uniform interface across platforms that allows domain experts, working in incompatible environment, to share generic knowledge. Moreover, the web environment provides a medium for an easy distribution and use of the 'Electronic Guide Book'. Studying the feedback from the users should lead to validate and complete the pattern repository.


Descriptor - Microsoft Internet Explorer
Fichier Edition Affichage Aller à Favoris ?
Submit your selection
Cancel your selection
Descriptor - Microsoft Internet Explorer
Fichier Edition Affichage Aller à Favoris ?
Submit your selection
Cancel your selection

You have selected :

Type :
Domain : Customer servicing in an integrated monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$

You now have to select ONE of the following alternatives :

● Type :
Domain : Customer servicing in an integrated monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$ **(from meter reading)$_{source}$ (using meter readers)$_{manner}$**

○ Type :
Domain : Customer servicing in an integrated monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$ **(automatically)$_{manner}$**

You have selected :

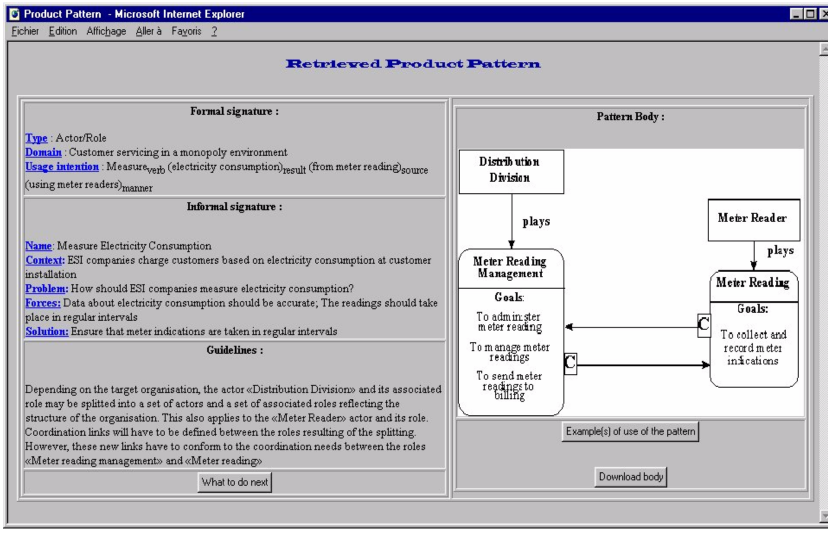
Type :
Domain : Customer servicing in an integrated monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$ (from meter reading)$_{source}$ (using meter readers)$_{manner}$

You now have to consider either ONE or SEVERAL of the following items :

● Type : **Actor/Role**
Domain : Customer servicing in an integrated monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$(from meter reading)$_{source}$ (using meter readers)$_{manner}$

○ Type : **Object**
Domain : Customer servicing in an integrated monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$(from meter reading)$_{source}$ (using meter readers)$_{manner}$

Figure 7 - Descending the hierarchy of formal signatures


Product Pattern - Microsoft Internet Explorer
Fichier Edition Affichage Aller à Favoris ?
What to do next
Distribution Division
plays
Meter Reader
plays
Meter Reading Management
Goals:
To administer meter reading
To manage meter readings
To send meter readings to billing
Meter Reading
Goals:
To collect and record meter indications
C
C
Example(s) of use of the pattern
Download body

Retrieved Product Pattern

Formal signature :

Type : Actor/Role
Domain : Customer servicing in a monopoly environment
Usage intention : Measure$_{verb}$ (electricity consumption)$_{result}$ (from meter reading)$_{source}$ (using meter readers)$_{manner}$

Informal signature :

Name: Measure Electricity Consumption
Context: ESI companies charge customers based on electricity consumption at customer installation
Problem: How should ESI companies measure electricity consumption?
Forces: Data about electricity consumption should be accurate; The readings should take place in regular intervals
Solution: Ensure that meter indications are taken in regular intervals

Guidelines :

Depending on the target organisation, the actor «Distribution Division» and its associated role may be splitted into a set of actors and a set of associated roles reflecting the structure of the organisation. This also applies to the «Meter Reader» actor and its role. Coordination links will have to be defined between the roles resulting of the splitting. However, these new links have to conform to the coordination needs between the roles «Meter reading management» and «Meter reading»

Pattern Body :

Figure 8 - The retrieved product pattern

6. References

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. (1977*)* A Pattern Language, Oxford University Press, New York.

Alexander, C. (1979) The Timeless Way of Building, Oxford University Press, NY.

Beck, K. (1997) Smalltalk Best Practice Patterns. Volume 1: Coding, Prentice Hall, Englewood Cliffs, NJ.

Bieber, M. and Kacmar, C. (1995) Designing Hypertext Support for Computational Applications, Communication of the ACM, 38(8), 1995, 99 - 107.

Brophy, K., Koets, T., (1996) Le programmeur VBScript S & SM, France 1996.

Bubenko, J., Strina, J. (1997) EKD User Guide, research report, ELEKTRA project.

Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P. and Stal, M. (1996) Pattern-Oriented Software Architecture - A System of Patterns, John Wiley.

Coad, P. (1992) Object-Oriented Patterns, in *Communications of the ACM*, Vol. 35, No. 9, 152-159.

Coad, P. et al. (1996) Object Models - Strategies Patterns and Applications, Yourdon Press Computing Series.

Coplien, J.O. and Schmidt, D.O. (1995) (ed.) Pattern Languages of Program Design, Addison-Wesley, Reading, MA.

Fowler, M. (1997) Analysis Patterns: Reusable Object Models, Addison-Wesley.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994*)* Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, MA.

Grosz, G., Rolland, C., Schwer, S., Souveyet, S., Plihon, V., Si-said, S., Ben Achour, C., Gnaho,C. (1997) Modelling and Engineering the Requirements Engineering Process : An Overview of the NATURE Approach, in *Requirements Engineering Journal*, (2), 115-131.

Harmsen, F. et al (1994) Situational method engineering for informational system project approaches, *in Method and Associated Tools for the Information Systems Life Cycle*, Verrijn-Stuart and Olle (eds.), North Holland.

Hay, D. (1996) Data Model Patterns: Conventions of Thought, Dorset House, NY.

Loucopoulos, P., Kavakli, V., Prekas, N., Rolland, C., Grosz, G. and Nurcan, S. (1997) Using the EKD Approach: The Modelling Component, Research Report (ELEKTRA project), March 1997.

Rolland C., Plihon, V. (1996) Using Generic Method Chunks to Generate Process Models Fragments, in *ICRE'96*, Colorado-Spring, USA, April 1996.

Rolland, C., Nurcan, S. and Grosz, G. (1997a) Guiding the participative design process, in *Association for Information Systems, Americas Conference on Information Systems*, Indianapolis, Indiana, 15-17 Aug. 1997, USA, 922-924.

Rolland, C., Nurcan, S. and Grosz, G. (1997b) A way of working for change processes, in *International research Symposium: Effective Organisations*, Sept. 4-5, 1997, Dorset, UK.

Rolland, C., Nurcan, S. and Grosz, G. (1998a) A unified framework for modelling co-operative design processes and co-operative business processes, in *the 31st Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA, 6-9 January 1998.

Rolland, C., Grosz, G., Loucopoulos, P., Nurcan, S. (1998b) The patterns model, Intermediary Research Report, (ELEKTRA Project), January 1998.

Vlissides, J.M., Coplien, J.O. and Kerth, N.L. (1996) (ed.) Pattern Languages of Program Design 2, Addison-Wesley.

Yajima, M.(1997) Deregulatory reforms of the Electricity Supply Industry, Quorum Books.