

FINAL PROJECT: CORDYS QUICKSTART TOOLBARS

Manuel Nicolas Bustillos Ardaya

**Hogeschool van Utrecht
Faculty of Nature and Technology
Department ICM**

Mediatheek HvU



MANAGEMENT SUMMARY

The conception of this project arises from a necessity caused due to the extensive complexity in the Cordys user interface, a shortcoming that can be seen as a competitive disadvantage, specially considering the importance that human-machine relationship are winning in the software industry. Seeking a solution to what could be considered as a handicap for the new Cordys BCP user, the main objective of this project is to offer new users a low-level entry approach into the Cordys environment that does not demand high knowledge nor extensive time-consuming training. Such achievement would result in a much more attractive product, not only from the aesthetic point of view but also for higher productivity that can be achieved without investing in training for employees.

But the value of the product of this project goes beyond the training purposes, since in the long term, as the user gets to be an advanced user of Cordys BCP, he can still take advantage of these Toolbars for important time saving.

To further assure the value and usability of the Toolbar, pattern-oriented techniques will be used. These patterns will follow the experienced Cordys users behaviour and they will be reflected in the final Toolbars.

The final product of this project is planned to be a part of the Content Package, a new feature to be included in the Cordys BCP 4.1 FP2. The release date for this new version was intended for the end of May/2004, however due to administrative issues it has been pushed back to mid May/2004.

INDEX

MANAGEMENT SUMMARY	2
1. GENERAL.....	6
1.1. INTRODUCTION.....	6
1.2. CORDYS COMPANY BACKGROUND	6
1.3. DESCRIPTION OF THE CORDYS COMPANY	7
1.3.1. <i>Cordys Vision</i>	7
1.3.2. <i>Cordys Mission</i>	8
1.4. CORDYS BUSINESS COLLABORATION PLATFORM (BCP)	9
1.4.1. <i>Cordys Integrator</i>	11
1.4.2. <i>Cordys Orchestrator</i>	12
1.4.3. <i>Cordys Portal</i>	12
1.4.4. <i>Cordys Business Application Connectors</i>	13
1.5. POSITION OF THE STUDENT IN THE COMPANY	13
1.6. STARTING SITUATION	13
2. PROJECT DEFINITION	16
2.1. DEFINITION OF THE PROBLEM	16
2.2. PROJECT ASSIGNMENT	16
2.3. OBJECTIVES.....	17
2.3.1. <i>Project General Objective:</i>	17
2.3.2. <i>Specific Objectives</i>	17
2.4. SCOPE OF THE PROJECT	17
2.5. KNOWLEDGE TRANSFER	18
2.6. PRODUCTS TO BE DELIVERED	18
3. ORGANIZATIONAL ASPECTS.....	19
3.1. RESOURCES	19
3.1.1. <i>Information resources</i>	19
3.1.2. <i>Software resources</i>	19
3.1.3. <i>Hardware Resources</i>	19
3.1.4. <i>Human resources</i>	19
3.2. METHODS, TECHNIQUES, STANDARDS AND DEVELOPMENT PLATFORM.....	20
4. PROJECT MANAGEMENT.....	22
4.1. APPROACH	22
4.2. ACTIVITIES SCHEDULE	22
4.3. ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS.....	23
4.3.1. <i>The project will be executed assuming the following:</i>	23
4.3.2. <i>Dependencies of the project:</i>	23
4.3.3. <i>Constraints of the project:</i>	23
4.4. SUCCESS FACTOR	23
4.5. RISK MANAGEMENT	24
4.6. RELATIONSHIP WITH OTHER PROJECTS.....	24

4.7.	PROJECT COMMUNICATION PLAN	24
4.8.	QUALITY PLAN	25
5.	PROJECT REQUIREMENTS ANALYSIS	26
5.1.	INTERVIEWING SESSIONS	26
5.2.	PATTERN-ORIENTED DOCUMENTS ELABORATION	27
5.2.1.	<i>Introducing the pattern concept</i>	27
5.2.2.	<i>Important considerations for the patterns</i>	28
5.2.3.	<i>Patterns Summary table</i>	28
5.2.4.	<i>Generic Pattern template</i>	29
6.	PROJECT DESIGN	31
6.1.	THE PROCESS MODEL CONCEPT	31
6.2.	THE PATTERN-MODEL RELATIONSHIP	31
6.3.	THE MODEL-TOOLBAR RELATIONSHIP	31
6.4.	IMPORTANT CONSIDERATIONS	32
7.	PROJECT IMPLEMENTATION	34
7.1.	FIRST ITERATION	34
7.2.	SECOND ITERATION	35
7.3.	THIRD ITERATION	36
7.4.	PROJECT DEVIATION	37
7.5.	FINAL STAGE OF IMPLEMENTATION	37
8.	FINAL TESTING	37
9.	CONCLUSIONS AND RECOMMENDATIONS.....	38
9.1.	ANALYSIS OF EVENTUAL RESULTS	38
10.	BIBLIOGRAPHY	40
11.	APPENDIX 1: ABBREVIATIONS.....	41
11.1.	41	
12.	APPENDIX 2: QUICKSTART PATTERN FORMS	42
12.1.	ADMINISTRATIONAL DEPLOYMENT	42
12.2.	ADMINISTRATIONAL TASKS	44
12.3.	BUSINESS APPLICATION DEVELOPMENT	46
12.4.	BUSINESS APPLICATION INTEGRATION.....	48
12.5.	PORTAL DEVELOPMENT	50
12.6.	COLLABORATION ENVIRONMENT DEVELOPMENT	52
12.7.	STUDIO MODELING.....	54
13.	APPENDIX 3: QUICKSTART PROCESS MODELS.....	56
13.1.	ADMINISTRATIONAL DEPLOYMENT	56
13.2.	ADMINISTRATIONAL TASKS	57
13.3.	BUSINESS APPLICATION DEVELOPMENT	58
13.4.	BUSINESS APPLICATION INTEGRATION.....	59
13.5.	PORTAL DEVELOPMENT	60
13.6.	COLLABORATION ENVIRONMENT DEVELOPMENT	61

13.7. STUDIO MODELING.....	62
----------------------------	----

FIGURE INDEX

FIGURE 1: CORDYS ORGANIZATION	6
FIGURE 2: CORDYS BUSINESS FRAMEWORK.....	9
FIGURE 3: CORDYS BUSINESS COLLABORATION PLATFORM ARCHITECTURE.....	10
FIGURE 4: CORDYS INTEGRATOR.....	12
FIGURE 5: CORDYS EXPLORER FRAMES	14
FIGURE 6: CORDYS EXPLORER UI	14
FIGURE 7: RESULTING CORDYS EXPLORER UI	38

TABLE INDEX

TABLE 1: PROJECT ORGANIZATION	20
TABLE 2: PROJECT INTERVIEWEES	20
TABLE 3: RISK MANAGEMENT	24
TABLE 4: PATTERN SUMMARY TABLE	29
TABLE 5: GENERIC PATTERN TEMPLATE	30

1. GENERAL

1.1. Introduction

The purpose of this document is to demonstrate the accountability and transparency of the execution of the project assigned. It is also intended for presentation to the academic tutor and eventually to the graduation committee in order to fulfil the requirements of the Hogenschool van Utrecht for graduation.

1.2. Cordys Company background

About Vanenburg Group B.V.

Headquartered in The Netherlands, Vanenburg Group is a federation of IT businesses that are focused on technology applications and services that enable enterprise collaboration. Vanenburg Group has a history of investment in Enterprise Applications that dates back to 1978. The Group started Baan company which was the leading Enterprise Resource Planning vendor competing with SAP and Oracle in 2000.

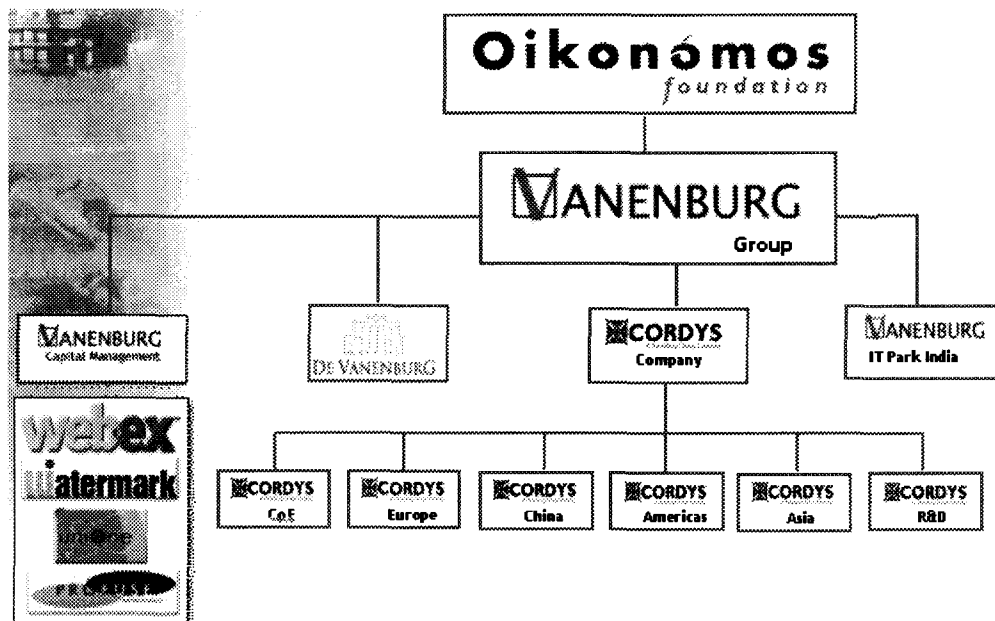


FIGURE 1: CORDYS ORGANIZATION

Vanenburg Group BV is the holding company and its software products are marketed under the **CORDYS** brand name.

1.3. Description of the Cordys Company

Founded in 2000 in the Netherlands, Cordys is a global software company that develops inventive collaborative solutions to deliver superior levels of adaptability, efficiency and responsiveness to companies and their networks.

Its a provider of real-time enabling, and application solutions based on Web Services that allow enterprises to easily define, deploy and manage business processes that are highly flexible, collaborative and involve seamless interaction among Organizations, People, Applications and Processes.

Cordys scalable, adaptive solutions fully leverage existing enterprise applications and extend their life cycle by adding new functionality and making them able to respond more quickly to ever-changing business requirements.

As the key underlying enabler for the real-time Enterprises, service-oriented Architecture will affect application design, deployment and the overall IT infrastructure for value-added networks: Modularity, Encapsulation, Interoperability.

1.3.1. Cordys Vision

From a business perspective, Cordys sees that people are gaining more power by being smarter, more autonomous and more demanding. Global competition compels companies and their networks to outsmart each other. But only a few of them have the ability to provide the next level of adaptability that really meets their clients' evolving expectations.

From a technology perspective, Cordys see that the commonly applied technology is failing to elevate the organizational performance to the desired level. However, more recently developed technologies -internet related and interactive by nature- offer incomparable opportunities.

By combining new, groundbreaking, Internet associated technologies with existing, commonly applied technology, they can offer companies unprecedented opportunities to reach new levels of business performance and truly satisfy stakeholders.

1.3.2. Cordys Mission

To energize companies with the 'Art of Opportunities' by delivering superior levels of adaptability, efficiency and responsiveness within their networks.

Cordys Real-Time business solutions are collectively known as the ***Cordys Business Framework***. Cordys Business Framework consists of:

- **Technology:** Cordys BCP (Business Collaboration Platform) provides a Service-Oriented Architectural framework based on Web Services that enables enterprises to integrate, orchestrate, monitor and improve key business processes and mission critical applications.
- **Applications:** Cordys B-Apps (Business Applications) is a suite of Web Services based solutions that allow enterprises to easily manage quality, production and services processes for their extended enterprise.

The diagram below describes the Cordys Business Framework:

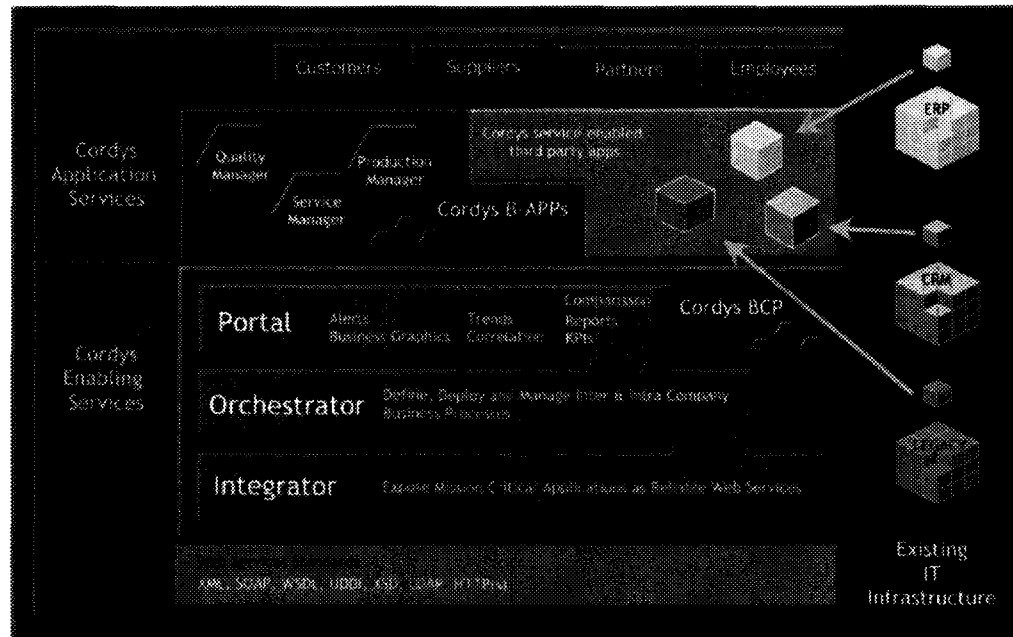


FIGURE 2: CORDYS BUSINESS FRAMEWORK

As it will be noticed while going through this document, the main focus of this project will be in the lower part of this framework, the Cordys Enabling Services, more specifically in the Cordys BCP.

1.4. Cordys Business Collaboration Platform (BCP)

Cordys BCP (Business Collaboration Platform) provides a state-of-the-art ESB (Enterprise Service Bus) that is able to adequately support mission critical applications through the extended Enterprise. Cordys Business Collaboration Platform has been completely architected using Web Services standards (no legacy).

Cordys Business Collaboration Platform offers real value by enabling enterprises to:

- Maintain visibility of transactional and business flows in real-time across the organization's extended network
- Easily integrate key legacy systems beyond original scope of your current IT infrastructure
- Enable the adoption of new technologies as they arrive without re-engineering

- Ensure data and transaction integrity among combined applications (strong support for exposing legacy applications as key services for the enterprise).
- Assist customers in the development of new software components that are Service-oriented Architecture (SOA) compliant.
- Provide for adequate process management, including exception handling.

The figure below shows the architecture of the Business Collaboration Platform of Cordys.

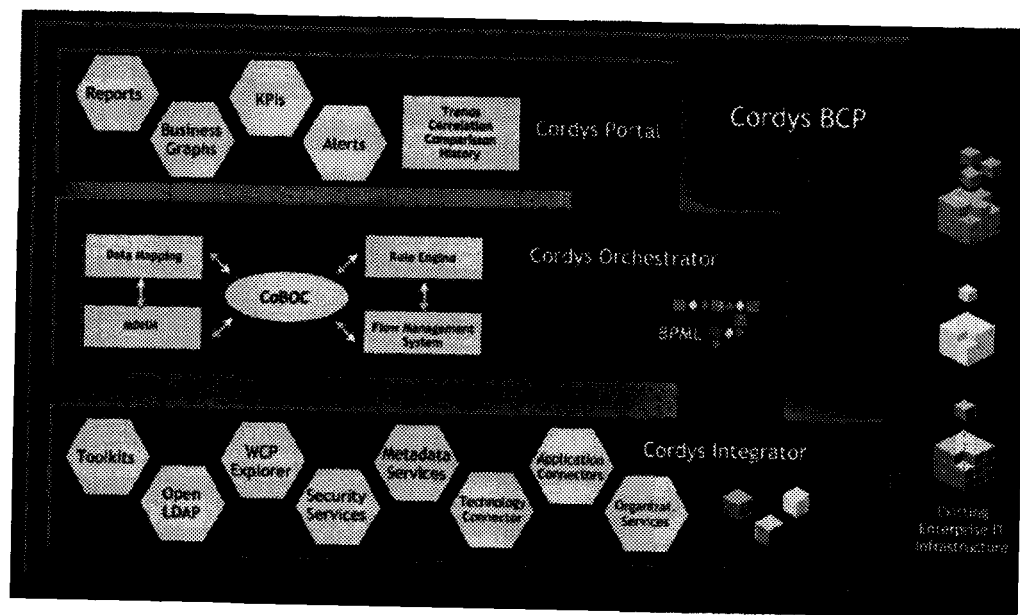


FIGURE 3: CORDYS BUSINESS COLLABORATION PLATFORM ARCHITECTURE

From this figure, it is important to emphasize the 'WCP Explorer' in the Cordys Integrator, the entry-point and UI to achieve any activity or interaction with the Cordys BCP. Moreover it is an important point of interest for this project since it is there where its final product will be reflected.

Cordys Business Collaboration Platform is comprised of the following components:

- Cordys Integrator
- Cordys Orchestrator

- Cordys Portal
- Cordys Business Application Connectors

1.4.1. Cordys Integrator

Cordys Integrator enables all intra / inter enterprise applications to become exposed as Web Services using industry standards (e.g. SOAP, XML, LDAP, WSDL). This includes both new and existing (i.e., legacy) applications, as well as systems that have been architected to comply with either a ".NET" or "J2EE" framework.

Cordys Integrator provides a platform for Rapid Application Development and Deployment, and can be used to build Web, Mobile and Java-based application. An integrated application built using Cordys Integrator can have both front end and back end components. Integrator provides a framework for developers to develop their applications within, and tools that help the developers to develop their applications. It leverages on Web service technology to expose required business logic, meaning that all functions, or services, are defined using a description language and have invokable interfaces that are called to perform business processes. Web services use SOAP as the standard web protocol for communication and Cordys Integrator can be used to develop web services, which use XML as a medium of information. They can be used to develop applications that work together regardless of where they reside and the protocol they use to communicate. Once applications become services, Cordys Business Integrator can also be leveraged to provide integration, establish a corporate-wide framework for collaboration, and through the use of Cordys Orchestrator these services-based systems can then become part of company-wide business processes, involving: people, applications, departmental units, business partners, other business processes.

The next figure further details Cordys Integrator's components and their relation.

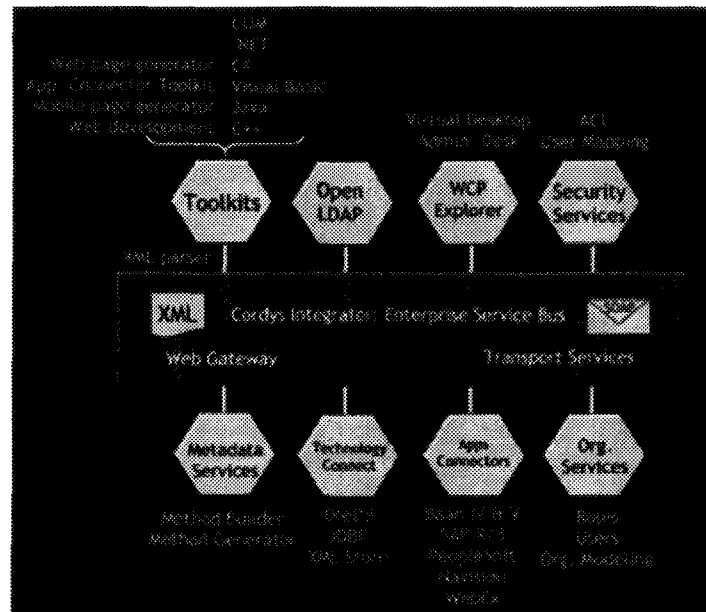


FIGURE 4: CORDYS INTEGRATOR

1.4.2. Cordys Orchestrator

Cordys Orchestrator enables modeling, deployment, monitoring and management of Real-Time Collaborative Business Processes. Based on Service-Oriented Architecture, Cordys Orchestrator seamlessly handles business processes spread across data sources, applications and enterprises. Cordys Orchestrator offers Web Services-based business process collaboration infrastructure and solutions leading to agile, responsive, process driven enterprise.

1.4.3. Cordys Portal

Cordys Portal provides enterprises with the ability to measure, monitor and improve business processes. Cordys Portal can reuse the services defined by the Cordys Integrator and provide the necessary building blocks for an enterprise-wide information portal. With Portal, users can have an access to performance indicators, analyze business intelligence, and use collaborative communication tools.

Cordys Portal includes a toolkit to easily build key performance indicators (KPI), alerts, business graphics and business analytics. It also incorporates embedded reporting capabilities and easily integrates into other enterprise reporting tools.

1.4.4. Cordys Business Application Connectors

Real-Time enterprises not only require new architectural components to implement agile, adaptive business processes throughout the extended value-chain, but also need to effectively leverage the existing IT infrastructure and maximize their value.

Application Connectors are translators that convert outgoing SOAP messages in Cordys Integrator to appropriate application/service APIs. The Application Connectors also convert information received from the APIs to incoming SOAP messages. Thus, in Cordys Integrator, an Application Connector is used to talk to a back end.

Cordys Business Application Connectors provide Web Services based access to a wide array of leading enterprise applications such as Baan, SAP, PeopleSoft etc and enable their business logic, data and information to become part of overall real-time framework deployed on Cordys BCP.

1.5. Position of the student in the company

The intern student is located in the Research & Development department fulfilling the role of planner and executer of the project, working on one of the key products of the company, the BCP Platform. He is under direct supervision of the Project Manager of the company.

1.6. Starting situation

The Cordys platform provides a very wide variety of applications and tools for many different purposes. The starting point for any activity in the Cordys environment is the **WCP Explorer** (or Cordys Explorer), a browser-based User Interface comprised mainly of eight frames, where invoked htm-applications can be docked for interaction with the user. The distribution of these frames across the browser is depicted in the following figure:

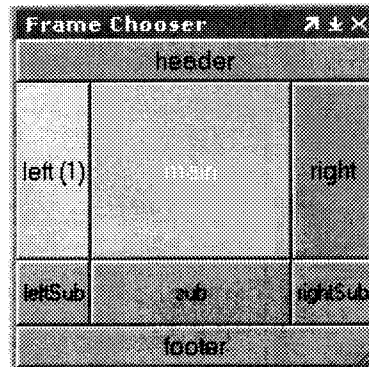


FIGURE 5: CORDYS EXPLORER FRAMES

In the Cordys Explorer, the main navigation interface is the **Menu tree**, docked in the left frame by default. From this interface all other Cordys Tools and Applications are prompted, usually, into the main frame. The following is a screen capture of the Cordys Explorer UI:

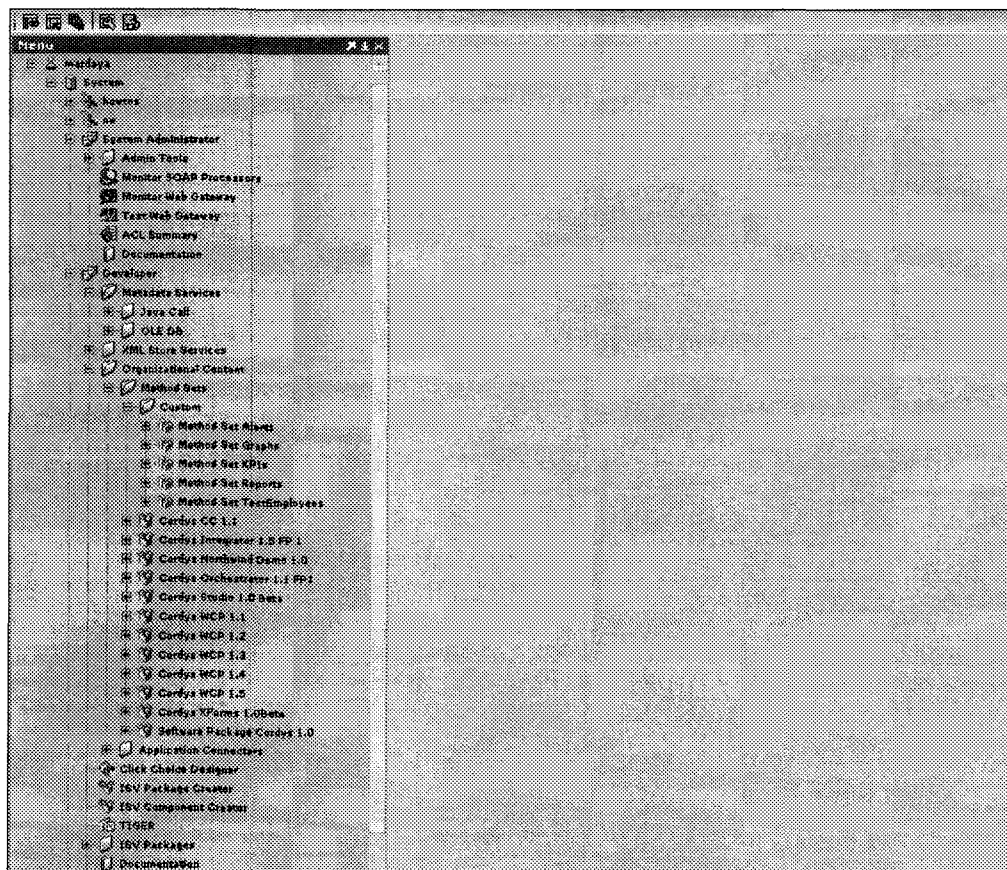


FIGURE 6: CORDYS EXPLORER UI

The default menu of the user has the User's name as the root of the tree, followed by the Organizations/Organizational Units that the user belongs to (only System in this case). The default organization/organizational unit of the user opens up to display the roles and menus that the user has access to in that organization/organizational unit. These roles and menus are assigned to a user by the administrator in that Organization/Organizational Unit. **System** is an organization primarily for maintaining Cordys Integrator and for grouping system objects.

Very generally, the current way to proceed is to browse through the Menu tree across the different roles and folders as needed, select the desired application, perform the necessary actions there and proceed with the next tree item in order to accomplish the task you need.

2. PROJECT DEFINITION

2.1. Definition of the problem

Cordys provides an extensive toolset to develop and configure web-based applications and web services. With the overwhelming possibilities and complex UI structure the risk exists that a developer new to Cordys drowns quickly in all provisions and loses oversight. At a simple first glance, the following problems can be detected:

- The starting point for any desired development task is not clear at all.
- The procedural sequence across the different tree items necessary along the developing process may not be obvious to the beginner.
- The whole tree-browsing process can be highly time consuming.
- Even for developing the simplest application there is a need for preliminary training/study.
- The available Cordys Documentation is not very objective and browsing through it can be as overwhelming as the Menu tree in the Explorer.

This creates the need to define an easy entry approach into the development environment supported by a simple task-oriented user interface.

2.2. Project Assignment

To provide the inexperienced Cordys user with a set of role-based and task-oriented toolbars customizable to his own needs. Each toolbar, consisting of a set of buttons, is associated with a common development task, enabling the developer to quickly gain access to the interfaces required along the developing process in order to intuitively accomplish this task.

Additionally, the user should be able to access help files describing how to proceed to accomplish the task, while taking full advantage of the buttons dispensed. Also included should be the necessary information in order to decide which toolbar suits better his specific situation (given the tasks he performs the most).

2.3. Objectives

2.3.1. Project General Objective:

Create a friendly and low-entry-level Quickstart development environment that supports simple task-oriented UI for the inexperienced Cordys user.

2.3.2. Specific Objectives

- Perform well-structured interviews to establish developer's requirements and to capture their expertise on a specific task in a way that can be transmitted to the beginner.
- Embody the information and results obtained in the interviews into well-defined Pattern templates
- Identify and rank the most common tasks, including the respective patterns that describe them.
- Develop Process models flows that describe the best way of executing this tasks as described by the interviewees
- Define a toolbar specification for each task with the necessary button-events that support the manual execution of the respective task.
- Write clear, beginner-oriented explanatory documentation aiding the appropriate usage of each toolbar for interactive learning. Also provide links for the minimum basic knowledge required to accomplish the task.
- Deliver resulting components in an ISV Package for easy deployment in the Cordys environment.

2.4. Scope of the project

- The number of toolbars to be defined is fixed and it will be defined after the interviewing phase in accordance with Project manager and Program manager.
- Initially it is expected that no additional functionality will be incorporated, but it is an aspect to be reconsidered if the circumstances demand so.

Not within the scope:

- The button events of the resulting toolbars will only support activities already available in the Cordys platform
- The toolbar generation, when transforming from Process model to Toolbar definition, will be performed manually.
- Advanced features of the Cordys platform will not be taken into consideration for support in the toolbar.
- Both the toolbar and the help files are expected to aid the user new to the Cordys environment, however it is expected some basic understanding of Web Services, XML, SQL and related industry standards.
- Generate Wizards to support the task is not considered as an option.

2.5. Knowledge transfer

All the knowledge acquired will be transferred in an explicit way through the explanatory help files. But also, each toolbar definition itself will indirectly reflect the knowledge captured from the developers interviewed.

2.6. Products to be delivered

- A deployable ISV Package containing the following components
 - Set of new defined roles
 - Set of toolbars definition (a XML-based file)
 - Process-oriented help documentation
- Final thesis report

3. ORGANIZATIONAL ASPECTS

3.1. Resources

3.1.1. Information resources

- Cordys Product Documentation
- Cordys web site
- Internet information
- Cordys Employees

3.1.2. Software resources

- Cordys BCP 4.1 FP 1
- Cordys XForms ISV Package
- Cordys Studio ISV Package
- Microsoft Visual InterDev 6.0
- Microsoft Internet Explorer 6.0

3.1.3. Hardware Resources

- DELL Latitude D600

3.1.4. Human resources

The student is responsible for the planning and execution of the project, including analysis, design, and implementation. But for supervision and supporting purposes there are also additional people involved.

Next table offer a view about the persons related to the project, and the functions they perform:

Name	Role in project	Function	Organization
Nicolas Bustillos	Planner and executer	Student Project	Development (Cordys Studio Team)
Wilfried Rijsemus	Supervisor	Program Manager	Program Management (Cordys)
Gerrit Spronk	Student Supervisor	Responsible of Students	Hogeschool van Utrecht
Arjan Klein	Technical Support	Software Architect	Development (Cordys Studio Team)
Joost Geuze	Technical Support	Software Architect	Development (Cordys Studio Team)
Henk ten Voorde	Project Owner	Program Manager	Program Management (Cordys)

Table 1: Project Organization

Additionally, there is also some personal indirectly involved for the interviewing phase. An initial list of potential interviewees is given in the following. However it is expected to add more names as deemed necessary.

Name	Title	Department
Freerk Wieringa	Connectivity team Mngr	Management
Phillip Gussow	Software Engineer	Development
Willem Jan Gerritsen	Software Engineer	Development
Rene Prins	Software Engineer	Development
Kees van Ravenhorst	Software Engineer	Development
Winfried van Holland	Product Mngr.	Development
Thijs Petter	Product owner	Management
Henk Rietveld	Developing Mngr.	Development
Jochem Scholenklopper	Software Engineer	Development
Jan Hoogendoorn	Project Manager	Consulting
Eddie Baron	Consultant trainer	Consulting

Table 2: Project Interviewees

3.2. Methods, techniques, standards and development platform

For this point it each item will only be listed and further in the document, at the respective moment, it will be explained how they are being used.

The *research methods* and sources that are going to be used are the following:

- ♦ Interviewing techniques (with members of the company for input and feedback)

- ♦ Information available on Internet.
- ♦ Cordys documentation.
- Pattern-oriented software *analysis* techniques
 - ♦ Pattern documenting
- *Design techniques* and tools to be used:
 - ♦ Business Process Modeling
 - ♦ Cordys Studio
- *Implementation languages to be used:*
 - ♦ XML
 - ♦ JavaScript
- *Standards* to be used:
 - ♦ BPMN
 - ♦ BPML
 - ♦ Progress report standards pre defined by the company.
 - ♦ Final paper standard pre define by the Hogeschool van Utrecht.
- Development *platform* to be used:
 - ♦ Cordys BCP 4.1

4. PROJECT MANAGEMENT

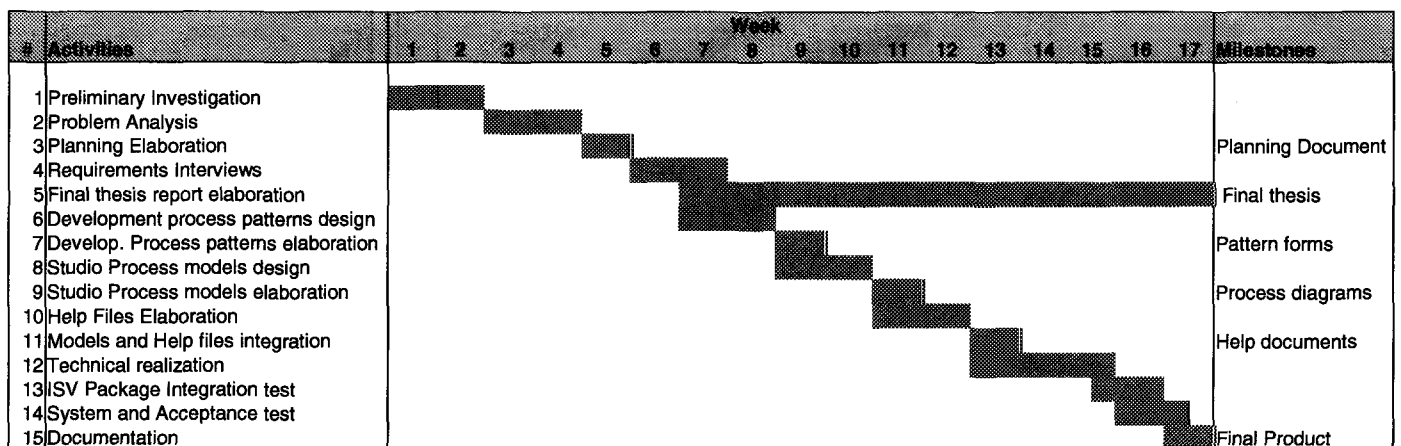
4.1. Approach

Considering that the essence of the project lies in providing the beginner with the required skills in an inferable way, the big challenge will be to acquire the skills in the different areas to a level high enough to be able of transferring them as accurately as possible to the beginner. For this purpose interviews will play an important role from the very beginning, in understanding the assignment, to the latest stage when performing the acceptance test. Being fully capable of performing the specific task will be the first step towards defining a toolbar that supports this task.

Also, to be able to decide which tasks better describe an experienced developer tendency, special importance will be paid to developing pattern definition, a newly emerging technique, that has not been thoroughly developed, so there is no specific standard to be followed in this aspect.

4.2. Activities Schedule

Starting date: 02/02/2004



■ = Milestone with deliverable

4.3. Assumptions, Dependencies and Constraints

4.3.1. The project will be executed assuming the following:

- The availability of the Cordys BCP and Studio all the time during the trainee period.

4.3.2. Dependencies of the project:

- Design details of parallel Toolbar generator project (further explained in 4.6)

4.3.3. Constraints of the project:

- Any function for the toolbar buttons is limited by the technology currently available in the Cordys BCP platform.
- The main constraint is time; the project is to be delivered by the end of May.

4.4. Success Factor

- Constant communication with all interviewees will assure the functionality, usability and reliability of each toolbar case.
- Along the whole process, constantly keeping in mind the project keywords (simple, quick and beginner-oriented) will assure that the project objective is met.
- Permanent overview of the technical feasibility will assure the completion of the project.

4.5. Risk Management

Risk	Action	
Conflicts arising from contradictory requirements	Program meeting involving individuals	
Oft Declination of Meeting requests	Consider alternate channels of communication (chat, phone, etc.)	
Technical constraints on a specific activity performed by a button	In the worse case scenario the functionality of a button will be reduced to simple text display	
Time limitations due to excessive number of toolbars	Consider cutting out some of them based on the ranking previously created.	

Table 3: Risk management

4.6. Relationship with other projects

This project is closely related as a part of the Content Package project, a feature to be included in the Cordys BCP 4.1 FP2, which is planned to be released by the end of May.

Additionally, parallel to this project, there is another related project, regarding dynamic generation of toolbars based on Studio Process models. Although it has been considered to make use of this product to generate the toolbars (instead of doing them manually), this is an option that has been discarded due to the fact that both projects are to be delivered by the end of may and included in the Cordys BCP 4.1 FP2. However, special attention will be paid to this project during the whole process in order to avoid inconsistencies in the final products. Design details will be taken into consideration in order to converge in the same Toolbar format and coincide in the conversion logic between the Process model and the Toolbar definition.

4.7. Project Communication Plan

- a. Bi-weekly reports to the Program manager in Cordys and the tutor in Hogeschool van Utrecht

- b. Meetings with the interviewees will be programmed on the run to track progress, as they are needed.
- c. Regarding the related 'Toolbar generator' project, a regular communication channel will be established with Thijs Petter to get information about any change or addition in the design of this project.
- d. Bi-weekly meetings will be programmed with the academical mentor from the Hogeschool van Utrecht

4.8. Quality Plan

The quality control methodology will be more process-oriented rather than product-oriented, meaning that along the process the quality of every intermediate product will be tested. The advantage of this methodology is that the workload at the end of the project is reduced and all flaws are corrected on the run before pulling them to the next phase, thus reducing error propagation.

The acceptance criteria for each toolbar will be the approval of the developer who was interviewed for its conception. The intermediate products for each toolbar will be the respective pattern description, the process model, the help document and finally the toolbar definition itself.

5. PROJECT REQUIREMENTS ANALYSIS

For the resulting Toolbars to be really useful, it is important to establish what is truly expected from them. A good requirement analysis will assure that the product delivered has a positive impact and that it proves to come handy at the moment of developing. This phase of the project will be mainly divided in two parts: collection of input and its interpretation. As for the input, it will be composed of available documentation (obtained by simple reading) and, more importantly, from the people interviewed (explained in further detail in the following point). As mentioned before, for processing and interpretation of the input, pattern-oriented techniques will be applied(also explained in 5.2).

5.1. Interviewing sessions

Considering that this will be the main input for the whole project, special attention will be paid to this phase and the interviewed individual/developer (henceforth referred to as the *interviewee*) will keep playing a role even beyond this phase of the project.

In general, there are two main objectives for this part of the project:

- Clearly establish the requirements as for the number of toolbars to be developed and as for what is expected to accomplish with each one of them.
- In further detail, determine how can a toolbar accomplish what is required: the number of buttons and their respective functionality.

Given the ambiguity of the two objectives, every interviewee will be playing a two-sided role and the interviewing sessions will be divided in two parts:

- The first part will be more abstract and the interviewee will play somehow the role of a customer, hence, this part will be addressed to the developer's personal opinion and criteria regarding what are the most common tasks where a toolbar could support. The answers to this part will provide the parameters to best meet the first objective.

- Whereas the second part will consist of punctual technical questions appealing to the developer's technical knowledge rather than his common sense. The result here is expected to capture their best practices in doing a specific task. These answers will be the base for the Process models that will be directly included in the help files.

To best fit the scenario described above, the interviewing methodology to be applied will adopt an iterative nature; meaning that each interviewee will be met more than once over the duration of the project life cycle, not only to complete the two parts mentioned above, but also to accomplish a repetitive corrective process of the project progress. This methodology will allow taking corrective measures as each iteration takes place, which is a desired outcome in accordance with the quality plan.

5.2. Pattern-oriented documents elaboration

After the interviews had taken place, the results of the requirement analysis interviews were somehow rough and blurry, which is the reason why from the very beginning it was established the use of Pattern-oriented techniques to better organize and expose this knowledge. This way, the dispersed information obtained has been methodically embodied into pattern forms definitions. These forms follow a generic pattern template, also described in this document.

5.2.1. Introducing the pattern concept

Patterns for software development are one of the latest "hot topics" to emerge from the object-oriented community. They are a literary form of software engineering problem-solving discipline that has its roots in a design movement of the same name in contemporary architecture, literate programming, and the documentation of best practices and lessons learned in all vocations.

By definition, "a pattern is the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts", in other words, a solution to a recurring problem in a specific context. In generic terms, each

pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution.

The purpose of the pattern forms is to capture the knowledge and best practices of the experienced developer in a way that can be distributed and reused by beginners.

5.2.2. Important considerations for the patterns

There are some important issues that were taken into consideration at the moment of conceiving the patterns, and that also need to be kept in mind while reading them:

- As the whole project itself, these patterns are beginner-oriented, so that many of the most advanced features of Cordys BCP are not covered. However, they are mentioned as further advanced reference.
- Since the assignment required the toolbars to be common task oriented, these patterns address the most frequent situations and most used items from the Cordys Menu.
- Contradictory to the last point, while analyzing every pattern requirement and considering they guide in the process starting from scratch, there are some preliminary mandatory tasks that in the long term cannot be considered as common/frequent. However these tasks are shared among the different patterns, which lead to the creation of an additional pattern to support initial tasks and to be reused by the other patterns.
- As it may be appreciated from the name of the project, 'Quickstart' implies that following the solutions of these patterns will lead to a quick and simple way of accomplishing the intent of each pattern
- Every assumption/consideration made for these patterns that may be relevant to the toolbar user will be explained thorough the help files.

5.2.3. Patterns Summary table

As a result of the interviews, it was decided that the following 7 patterns best cover all the requirements for the toolbar functionalities, while comprising the main features of Cordys BCP.

Name	Role assigned	Intent
Administrational Deployment	System administrator	All necessary settings for a newly installed BCP.
Administrational Tasks	System/ Organization administrator	Establish, organize and maintain the Cordys development environment for the everyday operation at the organization level <i>(Recommended before the patterns below)</i>
Business application Development	XForm Designer/ Developer	Design and build basic Web/Mobile based Applications and deploy them in the Cordys environment
Business application Integration	Developer	Establish back end connection between two existing applications.
Portal Development	Center Court Developer	Create a Management dashboard with all the desired Business Intelligence Components, providing real time business performance measurement
Studio Modeling	Business Analyst	Design, maintain, store, version and publish business models to runtime
Collaborative Website	Center Court Developer/Ad ministrator	Provide the means for a collaboration-enabled scenario

Table 4: Pattern Summary table

5.2.4. Generic Pattern template

In order to better identify, describe and document each pattern, in a way that all the knowledge can be clearly shaped and organized, this template will be the outline to capture the essence of each pattern.

Element	Description
Name	<i>Name of the development process pattern</i>
Role	<i>Roles(s) required to be able to use the toolbar supported by this pattern</i>
Intent	<i>Concise summary of the pattern's intention and rationale</i>

Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	Activities involved: <ul style="list-style-type: none"> ▪ <i>(List of the activities that are further explained in the Process Models to be presented in a separate document soon after this)</i>
Consequences	<i>Benefits provided, potential liabilities</i>
	Benefits: <ul style="list-style-type: none"> ▪ <i>(Here are mentioned both the benefits of using Cordys Tools & Applications and, additionally, the benefits provided of using this Quickstart Toolbar)</i> Liabilities & Limitations:
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	Related to this pattern: <ul style="list-style-type: none"> • <i>(Recommended before or while going through the respective toolbar)</i> For further advanced features: <ul style="list-style-type: none"> ▪ <i>(Only advised after getting entirely familiar with the features offered by the toolbar supporting this pattern)</i>

Table 5: Generic pattern template

The resulting seven patterns forms have been included as appendix at the end of this document. These patterns were the foundation for the respective design of each Toolbar, thus assuring that each resulting Toolbar will fit the scenario described by each pattern.

6. PROJECT DESIGN

This part is closely related to the Patterns Documentation (included in the appendix), and should be referenced as needed while going through the different Process models, since each of them has been designed to fit the corresponding pattern and fulfill its intent. In the context of the project, the purpose of each of these models is to serve as the base for constructing seven Toolbar definitions as XML files, which are the main deliverable of the project.

Same as the patterns, there are seven models defined in this document. Furthermore the final versions of these process models are also intended to be included as reference in the Help files, which are also a deliverable of this project.

6.1. The Process model concept

In general, a process model depicts a course of action from a starting situation to a specific end. It comprises a series of activities and the concerning decisions to take along the process.

Each of the models presented in this document as appendix has been drawn based in BPMN, and making use of Cordys Studio as drawing tool.

6.2. The Pattern-Model relationship

Considering that the patterns were the requirement analysis tool, the models, equivalently, are the designing tool. The main input from the Pattern form to the Process model is of course the *Intent* for which it has to be designed. Furthermore, the process model can be considered as an extension of the *Solution*, since it involves the same activities listed in each Pattern form. Additionally, each model has been conceived also considering the other attributes in the pattern form (*Problem*, *Context*, *Consequences*, etc)

6.3. The Model-Toolbar relationship

In accordance with a closely related project within Cordys, it has been established that the conversion logic will be as simple as one button in the

Toolbar per activity in the Process model. As a result, the Toolbar will provide a fluent mean to go through the Process quickly and avoiding the complexity of the Menu tree. In addition, an extra button will be available in each Toolbar, which will link to a help file containing the respective Process model and explanatory text clarifying how to proceed with each button.

6.4. Important Considerations

There are some important issues that were taken into consideration at the moment of designing these models, and that also need to be kept in mind while following them:

- Continuing the pursuit of the project's keywords ('Quick' and 'beginner'), each model's outline of activities has been distributed in such a manner, that following it in a vertical way leads to the simplest and shortest way of accomplishing the pattern's intent.
- Even though the Models are Pattern-based, it is important to remember that they are also Toolbar-oriented. And the latter one will be mandatory at the moment of defining the number of activities presented in the model, considering that each activity will be associated to a button, which at the same time is linked to a single application interface. Therefore, the model by itself represents the necessary interfaces prompted to accomplish the pattern's intent, but it does not reflect a clear explanation for a beginner. However, for explanatory purposes in the help files, further detailed sub-processes will be available for those activities that comprise more actions within one single application interface.
- These Process models are not intended to be executable; instead they are a visual reference for a suggested course of action. The advantages for this scenario are listed below:
 - ✓ Since a developing process mostly resembles an iterative process, the developer should not be forced to follow a sequential course.

- ✓ In the long term, as the user gets to understand the function of each button, he can slowly detach himself of the process model and use the buttons independently for more specific smaller tasks.
- ✓ Once the user gets fully acquainted with the Toolbar and moreover with the Cordys BCP environment, he has the freedom to further customize his Toolbar for a purpose other than the one originally intended by the model.
- The decisions points (◇) encountered in any model will not be reflected in the resulting Toolbar. However, by means of the help files, it will be stated what the criteria parameters are and the corresponding consequences of either decision. Hence, the impact of decisions will only be seen in the actions of the user: whether he clicks on the next Toolbar button or not.
- Every assumption/consideration made for these models that may be relevant to the Toolbar user will be explained thorough the help files.

The corresponding Process Model can also be found attached to this document as appendix (APPENDIX 3).

7. PROJECT IMPLEMENTATION

Given the nature of the project, for this part a combined methodology was implemented. In the wide framework, considering that there were 7 Toolbars to be developed, a prototyping methodology was adopted; meaning that any progress attempt was first considered for one Toolbar and, if successful, then applied to the others one by one. But when focusing on each Toolbar individually, the process clearly reflected an iterative development, and for this document this will be the base since it depicts better the steps and obstacles during the implementation.

7.1. First Iteration

As a first step, it was desired to establish a clear framework for the coming iterations. For this purpose, based on the Process models, every toolbar was defined first in terms of *name*, number of *Items* (buttons) and respectively each button's *id*, *description* and in some cases *url* associated; in other words, the minimum definition parameters for a toolbar. The outcome of this procedure was a complete Toolbar, but with very limited functionality and many dummy buttons. This first iteration was accomplished making use of the Cordys Toolbar creator; hardly any manual coding was involved. By simply right clicking on the header and selecting **New > Toolbar**, an empty space becomes available in the left part of the Cordys Explorer (this is not a frame, since no application can be loaded here; it is specially dedicated for toolbars). Again, by right clicking on this blank space and selecting **New > Item**, a Properties window pops-up to define the button. Both, the properties window and the blank space can be appreciated in the following figure:

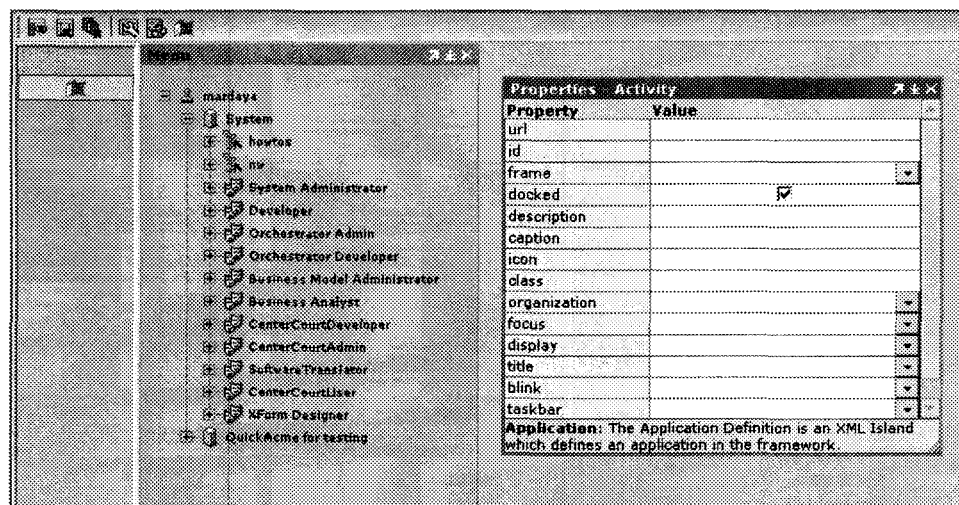


FIGURE 7: FIRST ITERATION

Each of these properties is reflected as a tag in the corresponding XML file of the Toolbar, and each button is reflected as an **<Application>** tag with all its properties defined within.

So far after this iteration, from the coding point of view, 7 XML files with the main root tags were already defined, and they were accessible from the XML Editor, a simple text pad available in the Cordys BCP to edit and manually change the XML files.

Up to this point, use of the toolbars provided the guidelines to complete a task, but the complexity of the Menu tree was still there.

7.2. Second Iteration

At this point it was clear that, with the available technology, it was not going to be possible to hide the Menu tree complexity. Because of this it was needed to alter the source code of the Menu tree, but this of course is not allowed. Therefore, an exact copy was made, altered and saved with a different name. Some buttons could then call this application, so that they can be able to show only the desired part of the Menu tree.

In the code of each Toolbar, making use of the XML Editor, the respective buttons were manually added a **<data>** tag (as an additional property) and the **<url>** tag was referenced to the newly created application. Within the

data tag, another Application would be defined, with the class of objects to be shown and the corresponding url.

The modified application, when called, should prompt for the contents of this tag and generate a different tree based on this information. For this purpose the original Menu application was edited using MS Visual InterDev. Within the JavaScript function '*CreateMenu()*', a new variable was created and assigned a value by calling the *selectSingleNode("data")* function. So, the *<data>* tag content is used to be the root of the new tree to be created.

For this part of the implementation, the 'Debug' feature of MS Visual InterDev was extremely useful. This feature allowed to attach the Cordys Explorer to InterDev and follow step by step the code sequence resulting from any action performed in the Cordys Explorer. This was necessary to compare the procedures followed by the old tree with the new one and make necessary adjustments to obtain the desired results.

7.3. Third Iteration

In order to achieve quick results when using the Toolbars, it was also required that one single button be able to open two simultaneous applications in two different frames, which is also not possible with the available technology. For this purpose an entirely new application was to be developed; this application (called the *launcher*) should execute hidden in the background and from there launch the desired applications. Buttons calling this launcher should also include the *<data>* tag with the corresponding urls and frames to be used.

This time, in the XML editor, within the *<data>* tag, two different Applications (thus two different urls) would be defined, identified by two different tags that the launcher could recognize. Additionally the main url was also referenced to the launcher.

In MS Visual InterDev, a new application was created with an empty *body* and a single JavaScript function to be prompted when the application was called. In this function two different variables are defined based on the two

tags within the <data> and they are launched consecutively also from this function.

7.4. Project Deviation

At this point the project took an unexpected turn, when a new set of requirements raised as a result of a progress meeting, where the current capabilities of the Toolbars so far were shown. Among some of these requirements, it was desired that also objects (e.g. a Table, a User, a XML file, etc.) should be accessible from the Toolbars and, if possible, Drag' n' drop enabled from the Menu tree.

As of the printing of this document, these new requirements are still under discussion and yet to be exactly determined. Therefore, the implementation of the project is still in progress.

For this reason, in coordination with the Program manger and Project Manager it was agreed to deviate from the established schedule in order to incorporate the new functionality. The new delivery date is now set to mid Juni.

7.5. Final stage of implementation

It is expected, that the final step will comprise the packaging of all the Toolbars definitions, applications and documents, making use of the Cordys ISV Package creator. This way, all the facilities provided by this project can be easily deployed into the Cordys environment for any user to be exploited.

8. FINAL TESTING

Due to the deviation from the planning, this phase of the project has yet not taken place as of to the date of publication of this document.

However, a lot of testing is involved during the implementation in each iteration, this assures that the final testing will be minimum. Additionally, regular meetings with the supervisors and live demos preformed in their presence validate the progress in the Toolbars.

9. CONCLUSIONS AND RECOMMENDATIONS

Even though the project has not yet been concluded, there are already many indicators that suggest the good progress and some of the results can already be analyzed.

Regardless of the new requirements, so far the general objective has clearly been covered and a new, lighter and friendlier UI can already be appreciated.

9.1. Analysis of eventual results

The positive result of this project, so far, can be appreciated at a simple glance, but there is a lot more additional value embedded as knowledge, that may not be visible by the user.

The following screen capture is an example of the resulting UI, and in the left part the Toolbar corresponding to Portal Development can be appreciated.

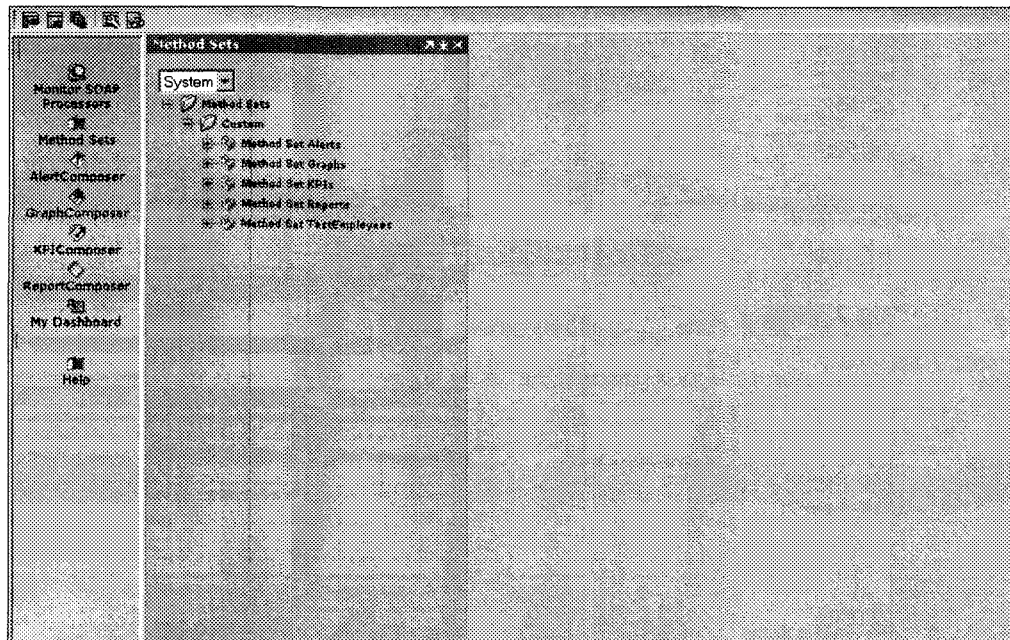


FIGURE 8: RESULTING CORDYS EXPLORER UI

This screen can be directly compared with figure 6 (page 14); in both cases the Method Sets folder is shown in the Menu tree. Some of the final advantages of the new UI regarding Portal Development are listed below:

- ✓ The top button in the Toolbar automatically explains to the user where to begin if he wants to develop a Portal.
- ✓ The order of the buttons explains the order to be followed in order to complete the task.
- ✓ By clicking in the Method sets button, not only can the user easily find the Method set button, but also saves some valuable time.
- ✓ The resulting Method set folder exposes only the objects that are relevant to complete the task (it hides the non-custom methods from the tree).
- ✓ At any point the user may press the Help button that will guide him in the process with the help of a process model and explanatory text.

Unfortunately, the two figures comparison provided in this document may not entirely illustrate the fast-paced dynamics provided to the new developing process when compared with the old way of doing it. It is expected that a live demo will much better reflect the value provided by this project.

The acceptance that the Program management has given to the project, is also signal of success of the project. It is clear that they are interested in the project results and they are willing to further invest in it. As of the publication of this document, a videoconference is planned with the counter part in India to further disseminate the core concept of this project and obtain additional feedback. Additionally, an internal pre-release has been recommended in order to be studied and revised by other developers.

10. BIBLIOGRAPHY

- Cordys web site
(www.cordys.com)
- Cordys Product Documentation
- "Understanding and Using Patterns in software development" Dirk Riehle and Heinz Zullighoven
(<http://www.riehle.org/1996/TAPOS-96-Survey.html>)
- Resources for learning Patterns
(<http://csis.pace.edu/~bergin/patterns/patternresources.html>)
- Patterns and Software: Essential Concepts and Terminology
(<http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>)
- Business Process Modeling Notation documents
(www.bpmn.org)

11. APPENDIX 1: Abbreviations

11.1.

Abbreviation	Description
BCM	Business Context Model
BCP	Cordys Business Collaboration Platform
BPM	Business Process Model
ERP	Enterprise Resource Planning
ISV	Independent Software Vendor
LDAP	Lightweight Directory Access Protocol
MRP	Manufacturing Resources Planning
SOA	Service-oriented Architecture
VCM	Value Chain Model

12. APPENDIX 2: QUICKSTART PATTERN FORMS

12.1. Administrative Deployment

Element	Description
Name	<i>Roles required to be able to use the toolbar supported by this pattern</i>
	Administration Deployment
Role	<i>Roles(s) required to be able to use the toolbar supported by this pattern</i>
	System Administrator
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Setting up the basic Deployment environment from a blank BCP at the system level.
Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none">▪ As a beginner, it might be desired to establish the basic and minimum settings for a quick start into the Cordys environment.▪ At first sight, the relevance of each of the System administration tools may not be visible.▪ In Cordys Integrator the administrative activities are performed at two levels, and the line dividing the responsibilities of the System administrator and the Organizational administrator may not be clear enough.▪ To maintain a better organizational management of a new project it is often required that a new Organization be entirely dedicated to it.
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
	<ul style="list-style-type: none">• The role of System administrator is a requisite for this pattern to be applicable.• Initial state establishing all appropriate settings in a recently installed Cordys BCP situation• New acquisition of an ISV Package required for deployment and configuration.• The beginning of a new project requiring to be assigned a new Organization.

Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	<p>Activities involved:</p> <ul style="list-style-type: none"> • Organization creation • Authenticated User generation • Load balancing using SOAP Node Configuration Techniques • Register ISV Packages • Manage License • Handle Failover • Customizations (Layout, Styles, etc.) <p><i>See process model for more detailed explanation</i></p>
Consequences	<i>Benefits provided, potential liabilities</i>
	<p>Benefits:</p> <ul style="list-style-type: none"> • Quickly enable and maintain an appropriate developing environment in the Cordys platform. • Exploit the advantages of grouping system objects as Organizations • Maintain visibility of your duties as a System administrator. <p>Liabilities & Limitations:</p> <ul style="list-style-type: none"> ▪ Since this pattern is beginner oriented, some advanced features are not included (Load balancing using Cats Clouder Clusters) ▪ Some advanced activities/configurations that are done outside the Cordys environment, and thus cannot be supported by a toolbar, have been left out of the scope of this pattern (hardware selection, IIS configuration, settings during installation, <code>wcp.properties</code> file manipulation, etc.)
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	<p>Related to this pattern:</p> <ul style="list-style-type: none"> • Administrative Tasks Pattern • Managing Authenticated Users • Managing ISV Packages • Handling Failover <p>For further advanced features:</p> <ul style="list-style-type: none"> ▪ Load balancing using Cats Clouder Clusters ▪ Deployment Guidelines: Nov 2003 ▪ Development setup Guidelines: Nov 2003

Table 5: Administrative Deployment form

12.2. Administrative Tasks

Element	Description
Name	<i>Name of the development process pattern</i>
	Administrational Tasks
Role	<i>Role(s) required to be able to use the toolbar supported by this pattern</i>
	Organizational Administrator
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Establish, organize and maintain the basic Cordys development environment for the habitual operation at the organization level.
Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none"> ▪ At first sight it is not clear exactly what entries define entirely an organization to be ready for developing activities. ▪ Regardless of the nature of the developing project in Cordys environment, there is a basic organization setup that needs to be configured in a similar way every time, thus investing considerable time in a repetitive procedure. ▪ Organizations may need to be further decomposed into departments with different Roles, Menus and SOAP Node specific to them.
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
	<ul style="list-style-type: none"> • Early stage of any developing project in general. • Necessary to establish the organizational outline, the data sources and Application Connectors, that are going to be required along the implementation of a developing process. • Before any developing attempt using any other Cordys Quickstart Toolbar there is a set of basic pre-requisites that can be quickly configured making use of this Toolbar.

Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	<p>Activities involved:</p> <ul style="list-style-type: none"> • Organizational Unit creation • User generation and role assignation • Database creation (manually) • Application Connector configuration (SOAP Node and Processor) • Create Menus and Roles assigned to them • SOAP Processor Monitoring <p><i>See Process Model for more detailed explanation</i></p>
Consequences	<i>Benefits provided, potential liabilities</i>
	<p>Benefits:</p> <ul style="list-style-type: none"> • Effectively reuse the organizational establishment procedure for any developing project regardless of its nature. • Gain quick access to the every-day administrative activities. • Manage the responsibilities as Organizational administrator in an efficient manner. <p>Liabilities & Limitations:</p> <ul style="list-style-type: none"> ▪ Data base creation is not supported in the Cordys Environment. ▪ Since this pattern is also for Organization administrators, System administration exclusive tasks are not included (Authenticated Users, Anonymous Users, Organization creation, Cats Clouder Clusters, Licensing, ISV Package Registry and (Un) Loading
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	<p>Related to this pattern:</p> <ul style="list-style-type: none"> • Cordys Integrator LDAP Schema • Overview of Application Connectors • Managing SOAP Nodes • Configuring Application Connectors <p>For further advanced features:</p> <ul style="list-style-type: none"> ▪ Administration: How do I... ▪ Auditing Applications ▪ Cordys Management Console

Table 6: Administrative Tasks form

12.3. Business Application Development

Element	Description
Name	<i>Name of the development process pattern</i>
	Business Application Development
Role	<i>Role(s) required to be able to use the toolbar supported by this pattern</i>
	Developer
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Design and build basic Web/Mobile based Applications and deploy them in the Cordys environment
Problem	<i>Issue(s) addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none"> ▪ No clear starting point for the developing process of a Web application. ▪ Usual required steps are scattered all over the Cordys platform and the sequence may not be obvious to the beginner. ▪ Limited functionality of the Cordys Web Generator ▪ Large complexity and long time invested in using the HTML Components (HTCs).
Context	<i>The situation or state of a development project in which the process pattern may be applicable and the basic Pre-requisites</i>
	<ul style="list-style-type: none"> • Starting implementation phase of an Application Development project • An installed XForm ISV Package is required and the respective Roles added to the User. • Basic organization setup completed (previous use of the Development Task Pattern is recommended) • OLE DB/JDBC SOAP Processor configured and running • Java Call SOAP Processor configured and running

Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	<p>Activities involved:</p> <ul style="list-style-type: none"> ▪ Check Status of required SOAP Processor ▪ Define and build all required methods ▪ Optionally use TIGER ▪ Design the XForm ▪ Optionally use Click Choice Designer ▪ Create ISV Components and Package <p><i>See corresponding Process Model for more detailed explanation</i></p>
Consequences	<i>Benefits provided, potential liabilities and limitations</i>
	<p>Benefits:</p> <ul style="list-style-type: none"> • Customize generated applications to include further enhancements without editing the code. • XForm technology fully exploited. • Decoupled data, logic and presentation • Minimizes the DHTML Programming • The pattern excludes the advanced features of Cordys, thus simplifying the developing process. <p>Liabilities & Limitations:</p> <ul style="list-style-type: none"> ▪ The pattern excludes the advanced features of Cordys, thus restricting the functionality of the resulting application. ▪ Manual method creation is not supported. ▪ The HTC's not supported in XForms are not included by this pattern
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	<p>Related to this pattern:</p> <ul style="list-style-type: none"> • Administrative Tasks Pattern • Development Methodology • Cordys XForms Documentation <p>For further advanced features:</p> <ul style="list-style-type: none"> ▪ Editing methods ▪ Programming Cordys XForms ▪ Application Development: How do I... ▪ Developer Best Practices

Table 7: Business Application Development form

12.4. Business Application Integration

Element	Description
Name	<i>Name of the development process pattern</i>
	Business Application Integration
Role	<i>Role(s) required to be able to use the toolbar supported by this pattern</i>
	Developer
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Allow business processes to connect to specific back ends and exchange messages making use of current industry standards.
Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none"> • No interaction between two stand-alone applications. • Need of going beyond simple information exchange and accomplish integration of business processes • Requirement to connect to a back end that is not supported by any of the available Application Connectors in Cordys BCP. • Custom JavaCall integration solutions are required to connect the Cordys Applications with a third-party software/back end
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
	<ul style="list-style-type: none"> ▪ The pattern covers only the connection; so two existing back ends (e.g. two applications) are required. ▪ Basic organization setup completed (previous use of the Development task Pattern is recommended) • The developer should be familiar with the Application Connectors in the Integrator architecture (See Overview of Application Connectors[.8]) • The developer should be familiar with Java libraries (JDK Docs) in case of JavaCall integration. • Java Call SOAP Processor configured and running • CoBOC Service SOAP Processor configured and running

Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	<p>Activities involved:</p> <ul style="list-style-type: none"> • Check Status of required SOAP Processor • Configure and run Application Connectors • Define and build all required methods • Optionally use TIGER • Use Workbench (Rules, Notifications, Business Channels, etc.) <p><i>See corresponding Process Model for more detailed explanation</i></p>
Consequences	<i>Benefits provided, potential liabilities</i>
	<p>Benefits:</p> <ul style="list-style-type: none"> • Take full advantage of the types of connectivity provided by Cordys Integrator as pre-defined (BDP, COMCall, JavaCall, etc.) • Automatically and safely generate code for other programming languages (C++, VBA and C#) • Achieve synergy in operations by integrating business processes. <p>Liabilities & Limitations:</p> <ul style="list-style-type: none"> • The pattern excludes the advanced features of Cordys. • Building custom non-JavaCall Application Connectors are not covered by this pattern.
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	<p>Related to this pattern:</p> <ul style="list-style-type: none"> • Administrative Tasks Pattern • Protocol for Application Connectors • Providing Integration Solutions using Cordys Integrator • Cordys Orchestrator Documentation <p>For further advanced features:</p> <ul style="list-style-type: none"> ▪ Working of a SOAP Processor ▪ Cordys integrator HowTos ▪ Developer Best Practices

Table 8: Business Application Integration form

12.5. Portal Development

Element	Description
Name	<i>Name of the development process pattern</i>
	Portal Development
Role	<i>Role(s) required to be able to use the toolbar supported by this pattern</i>
	Center Court Developer
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Create a Management dashboard with all the desired Business Intelligence Components, providing real time business performance measurement
Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none"> ▪ The Dashboard Composer is not accessible from the Menu tree. ▪ Center Court's Menu Composer provides only visibility of individual BI Components separately. ▪ Need to measure the performance and implement continuous improvement is imminent. ▪ Executive decision makers lack a wide and deep visibility of parameters across periods, impeding them to respond in real time.
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
	<ul style="list-style-type: none"> • Starting implementation phase of a Portal Development project • Basic organization setup completed (previous use of the Development Task Pattern is recommended) • CC Application Connector configured and running • OLE DB/JDBC SOAP Processor configured and running

Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	<p>Activities involved:</p> <ul style="list-style-type: none"> • Check Status of required SOAP Processor • Define and build all required methods • Compose Alerts • Compose Graphics • Compose KPIs • Compose Reports • Create Dashboard <p><i>See corresponding Process Model for more detailed explanation</i></p>
Consequences	<i>Benefits provided, potential liabilities</i>
	<p>Benefits:</p> <ul style="list-style-type: none"> • Exploit Center Court's information analysis capabilities. • Deeper & wider analytical insight into business processes • Converge related BI Components into one single portal <p>Liabilities & Limitations:</p> <ul style="list-style-type: none"> ▪ Because it is not supported in the Cordys environment, composing and invoking Crystal Reports have been taken out of the scope of this pattern. ▪ Since this pattern is beginner oriented, Click Choice on Content excluded of the suggested development process.
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	<p>Related to this pattern:</p> <ul style="list-style-type: none"> • Before you Begin (Center Court - Developer's guide) • Alerts, Graphs, KPIs and Report Composer • Collaborative Web Site pattern <p>For further advanced features:</p> <ul style="list-style-type: none"> ▪ Click Choice on Content ▪ Customizing Crystal Reports

Table 9: Portal Development

12.6. Collaboration Environment Development

Element	Description
Name	<i>Name of the development process pattern</i>
	Collaboration Environment Development
Role	<i>Role(s) required to be able to use the toolbar supported by this pattern</i>
	CenterCourt Developer/Admin
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Provides an organization-wide informal structure of communication and collaboration
Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none"> ▪ Lack of informal channels of communication abroad the organization as a whole. ▪ No availability of a mean to broadcast unstructured information proceeding from the people
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
	<ul style="list-style-type: none"> • Basic organization setup completed (previous use of the Development Task Pattern is recommended) • CC Application Connector configured and running

Solution	<i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i>
	Activities involved: <ul style="list-style-type: none"> • Discussion Boards • Surveys • Polls • Home Pages • User Profiles • Virtual Work Groups
	<i>See corresponding process Model for more detailed explanation</i>
Consequences	<i>Benefits provided, potential liabilities</i>
	Benefits: <ul style="list-style-type: none"> • Provides the broadcasting and network to enable a human collaboration scenario • Communication and portal horizontal capabilities across the entire spectrum of enterprise users.
See Also	<i>Reference to similar or related issues. Non pattern based also included</i>
	<ul style="list-style-type: none"> • Cordys Studio Center Court Documentation

Table 10: Collaboration Environment Development

12.7. Studio Modeling

Element	Description
Name	<i>Name of the development process pattern</i>
	Studio Modeling
Role	<i>Role(s) required to be able to use the toolbar supported by this pattern</i>
	Business Model Analyst/Administrator
Intent	<i>Concise summary of the pattern's intention and rationale</i>
	Design, maintain, store and publish collaborative Business Models in bottom-up fashion.
Problem	<i>Issue addressed, scenario description and its associated forces demonstrating the existence of the problem</i>
	<ul style="list-style-type: none"> ▪ The complexity of the system in the designing phase makes it difficult to maintain a clear overview of it as a whole. ▪ Uncertainty when introducing improvements into a complex system and still be able to predict the repercussion to assure the desired outcome. ▪ Need for model-based execution of a real life system. ▪ In the top-down modeling process, the Business Context Models are not available for drag 'n dropping while creating the Value Chain Model.
Context	<i>The situation or state of a development project in which the process pattern may be applicable</i>
	<p>The use of this pattern may be applicable in any of these situations:</p> <ul style="list-style-type: none"> • Early stage in the development of a system to simplify its analysis • Introduction of changes in an already existing system • Just create a simplified representation of something • Early modeling in process automation and system integration projects • An installed XForm ISV Package is required and the respective Roles added to the User. • It is expected a basic understanding of Business Modeling fundamentals in order to complete this pattern

Solution	<p><i>The suggested development process and activities. Using textual as well as graphical description techniques (BPM)</i></p> <p>Activities involved:</p> <ul style="list-style-type: none"> • Check Status of required SOAP Processor • Define and build required methods • Set user preferences (Design time and Run time) • Working with Business Model Components • Create the Value Chain Model • Create Business Context Model • Import services (Content Transfer Utility) • Create Business Process Model, publish and Instantiate <p><i>See respective process model for more detailed explanation</i></p>
Consequences	<p><i>Benefits provided, potential liabilities</i></p> <p>Benefits:</p> <ul style="list-style-type: none"> • In this pattern every model is optional, allowing the user to decide based on his necessities and objectives • Executable flows are also optional, making the developing process shorter and simpler. • Improved abstraction, analysis and communication in development projects <p>Liabilities & Limitations:</p> <ul style="list-style-type: none"> ▪ Following the solution of this pattern enforces the user to bottom-up modeling. It is also suggested to consider a top-down modeling, depending on the objectives of the modeling projects ▪ This pattern skips some non-common or non-beginner Studio Settings (Parameters, Versions, File types and Report Definitions)
See Also	<p><i>Reference to similar or related issues. Non pattern based also included</i></p> <p>Related to this pattern:</p> <ul style="list-style-type: none"> • Getting Started • Business Modeling Guide • BPMN Specification (www.bpmi.org) <p>For further advanced features:</p> <ul style="list-style-type: none"> ▪ Business Model Admin Guide ▪ Clustering Model ▪ Managing Reports

Table 11: Studio Developing

13. APPENDIX 3: QUICKSTART PROCESS MODELS

13.1. Administrative Deployment

Name	Administration Deployment
Role	System Administrator
Intent	Setting up the basic Deployment environment from a blank BCP at the system level.

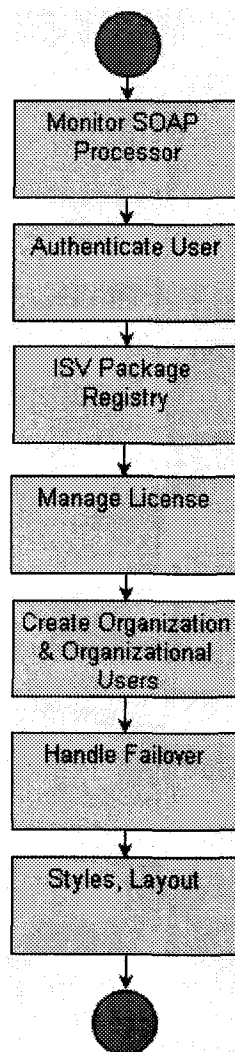


Figure 1: Administrative Deployment model

13.2. Administrative Tasks

Name	Administrational Tasks
Role	Organizational Administrator
Intent	Establish, organize and maintain the basic Cordys development environment for the habitual operation at the organization level.

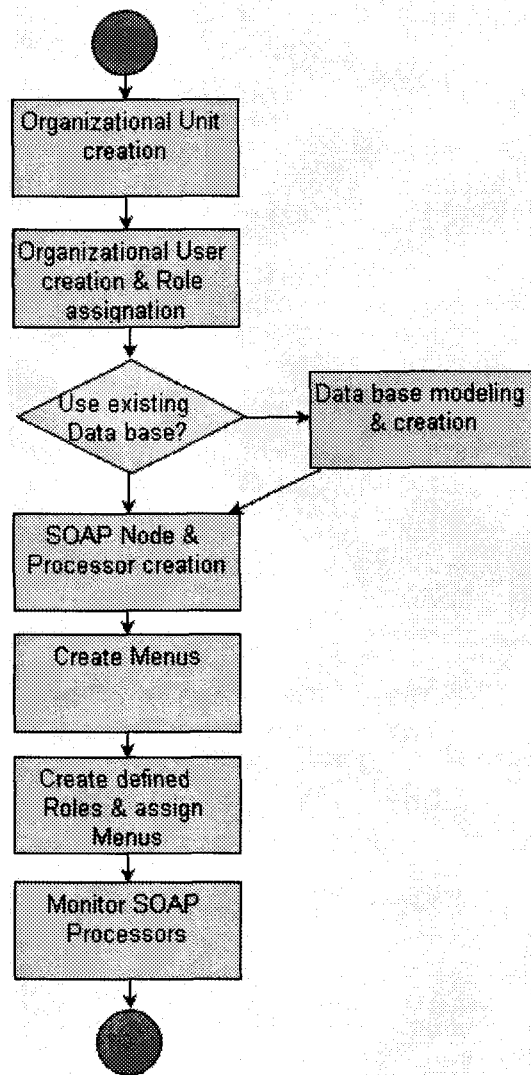


Figure 2: Administrative Tasks model

13.3. Business Application Development

Name	Business Application Development
Role	Developer
Intent	Design and build basic Web/Mobile based Applications and deploy them in the Cordys environment

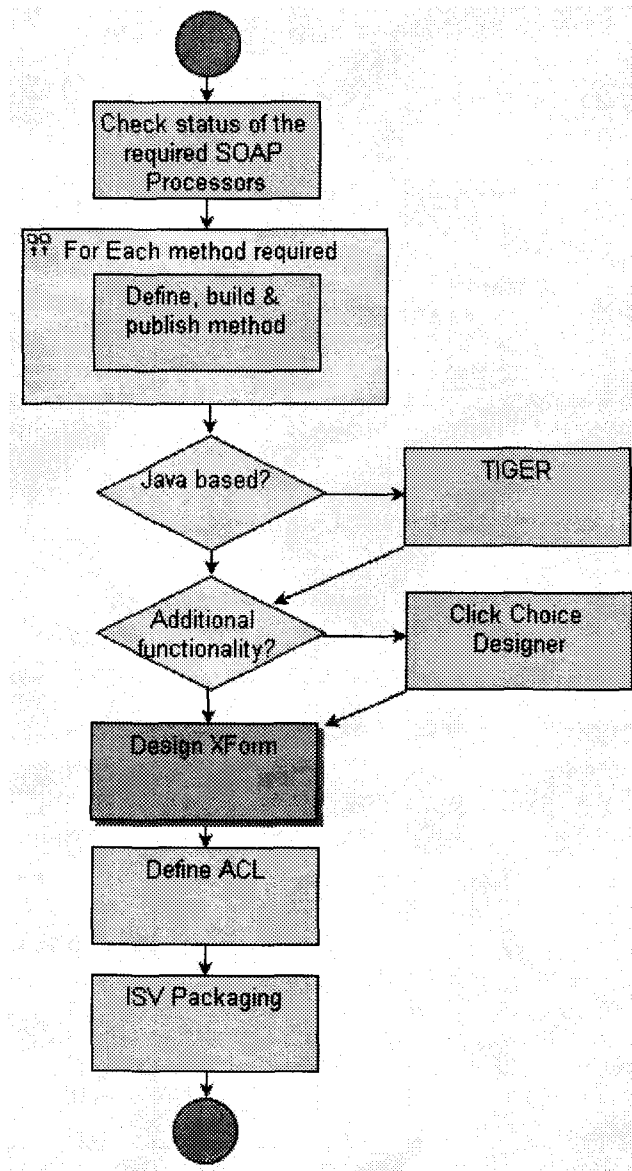


Figure 3: Business Application Development model

13.4. Business Application Integration

Name	Business Application Integration
Role	Developer
Intent	Allow business processes to connect to specific back ends and exchange messages making use of current industry standards.

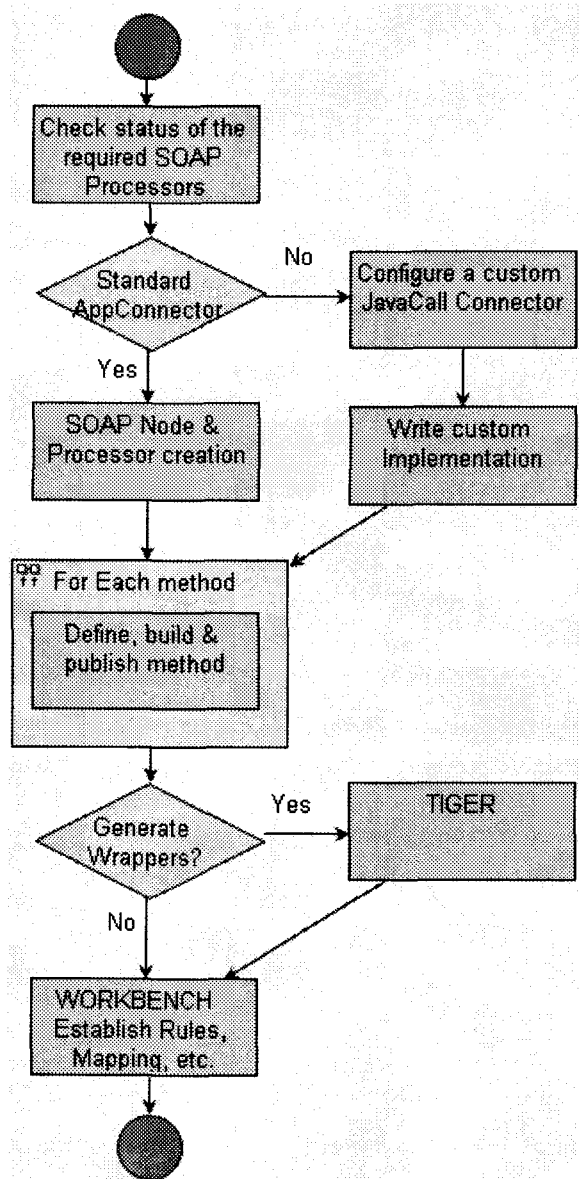


Figure 4: Business Application Integration model

13.5. Portal Development

Name	Portal Development
Role	Center Court Developer
Intent	Create a Management dashboard with all the desired Business Intelligence Components, providing real time business performance measurement

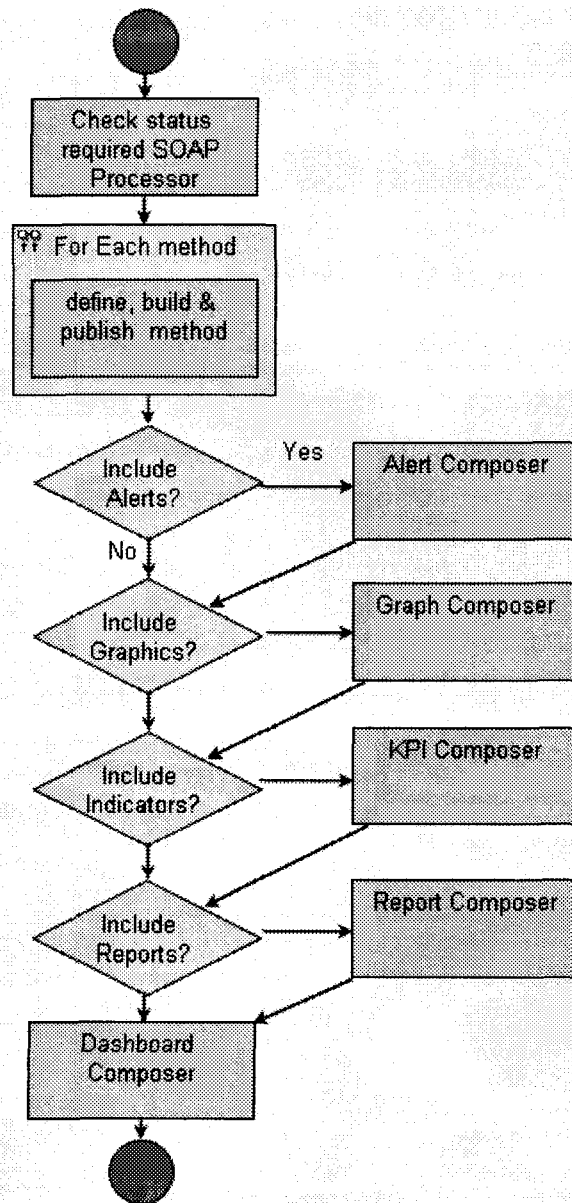


Figure 5: Portal Development model

13.6. Collaboration Environment Development

Name	Collaboration Environment Development
Role	CenterCourt Developer/Admin
Intent	Provides an organization-wide informal structure of communication and collaboration

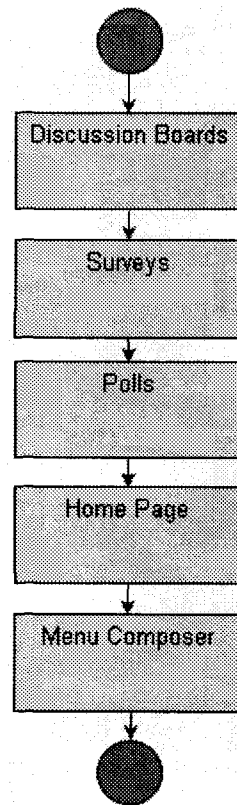


Figure 6: Collaboration Environment Development model

13.7. Studio Modeling

Name	Studio Modeling
Role	Business Model Analyst/Administrator
Intent	Design, maintain, store and publish collaborative Business Models in bottom-up fashion.

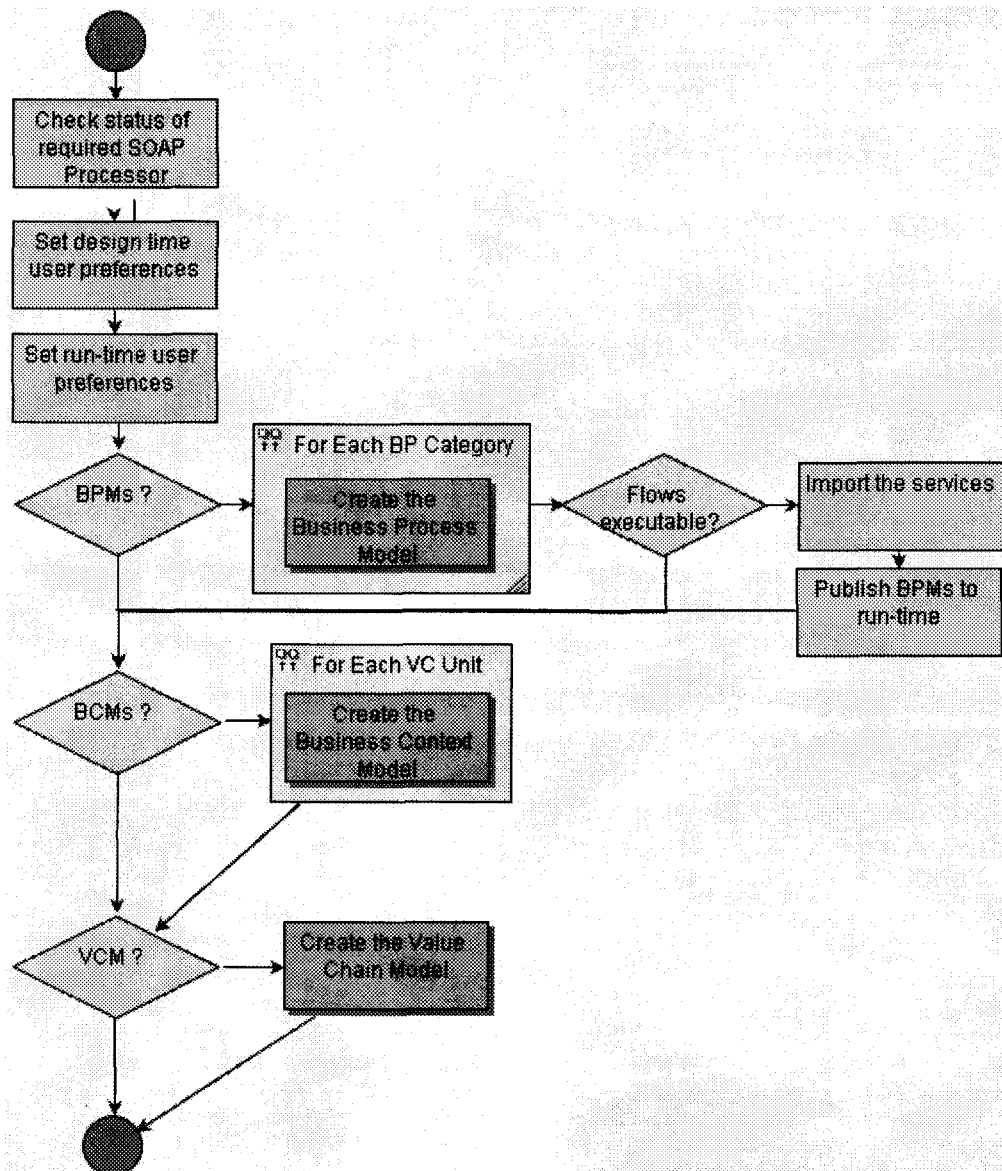


Figure 7: Studio Modeling model