

**Integration of pre-existing
heterogeneous information sources
18.08.2003**

IST-2001-32429

ICONS

Intelligent Content Management System

www.icons.rodan.pl

Project Partners

Rodan Systems (PL)

The Polish Academy of Sciences (PL)

Centro di Ingegneria Economica e Sociale (IT)

InfoVide (PL)

SchlumbergerSema (BE)

University Paris 9 Dauphine (FR)

University of Ulster (UK)



Integration of pre-existing heterogeneous information sources

Project name	Intelligent Content Management System
Acronym	ICONS
Workpackage	WP5
Task	T5.3
Document type	report
Title	Integration of pre-existing heterogeneous information sources
Subtitle	
Document acronym	D20
Author(s)	D.A. Bell, G. L. Zalewska, M. Momotko, N. Leone, P. Rullo, W. Piszczewiat, W. Staniszkis
Reviewer(s)	Nicola Leone, Witold Litwin
Accepting	Witold Stanisykis
Location	C:\bart\ICONS WP5 T3 D20 0102.doc
Version	1.11
Date	18.08.2003
Status	final version
Distribution	public

18.08.2003

History of changes

Date	Version	Author	Change description
18.8.03	1.10	Witold.Staniszkis	Final integration
05.07.03	1.05	D.A. Bell, G. L. Zalewska, M. Momotko, N. Leone, P. Rullo, W. Piszczewiat, W. Staniszkis	document expansion
15.04.20 03	1.00	Witold.Staniszkis	document creation

Executive summary

Information integration constitutes an important part of most knowledge management applications. We have set out information integration requirements in the initial project report the ICONS Project Description [ICONS D01] and this reports presents our research and development results matching those requirements.

The information integration features constitute an integral part of the ICONS platform architecture [ICONS D16] and their functionality is verified by the prototype knowledge management application, i.e. the Structural Fund Projects Knowledge Portal [ICONS D35].

The external information artefacts to be integrated in the ICONS repository include binary files of assorted formats, report files, XML annotated text files, extracted relational tables, and information directly derived from external information systems. The extracted relational tables, called materialised tables, are integrated from pre-existing relational databases or from Web pages. In the latter case, we use Lixto developed at the Technical University of Vienna [Baumgartner2001]. All information integration modules are tightly integrated within the ICONS platform.

Workflow platforms are universally considered as an important technology enabling integration of pre-existing, heterogeneous information systems. We have presented the integration features of the ICONS Intelligent Workflow module from two perspectives; (a) the information system integration, and (b) the distributed workflow process integration. Both capabilities are supported in the ICONS platform.

Table of contents

History of changes.....	3
Executive summary	4
Table of contents	5
List of figures	6
List of tables	6
1. Introduction	7
1.1 Objectives	7
1.2 Scope	7
1.3 Relations to Other Documents.....	7
1.4 Intended Audience.....	7
1.5 Usage Guidelines.....	7
1.6 Notation Conventions	7
2. The ICONS Information Integration Requirements	8
3. Overview of the ICONS Information Integration Features	9
4. ICONS Integration Control Structures	11
4.1 ICONS External Data Integrator Control Structures	11
4.2 ICONS Report Scheduler Control Structures	12
5. ICONS Data Extractor.....	14
6. LIXTO Integration.....	16
7. ICONS Materialised Table Semantic Integration.....	18
8. ICONS Intelligent Workflow based integration	22
8.1 The Intelligent Workflow core functionality	22
8.2 The ICONS Workflow process integration.....	23
8.3 The ICONS Workflow information system integration.....	24
Bibliography.....	25

List of figures

Figure 3.1. The ICONS Integration Architecture (IIA)	9
Figure 3.2. Content object classes and relationships pertaining to information integration	10
Figure 4.1. External Content Integrator – un-attributed class association diagram	12
Figure 4.2. Report Scheduler – un-attributed class diagram	13
Figure 5.1. Data Extractor Table Constructor window	14
Figure 5.2. Data Extractor Selection Criteria window	15
Figure 7.1. ICONS semantic materialised table integration use cases	18
Figure 7.2. Architecture of the Semantic Materialised Table Environment	19
Figure 7.3. Semantic materialised table integration class association diagram	21
Figure 8.1. Models of workflow co-operation	23

List of tables

1. Introduction

1.1 Objectives

Objectives of this report are to provide succinct information pertaining to the information integration features designed and developed within the ICONS project.

1.2 Scope

This report summarises the information integration features developed within the ICONS project.

1.3 Relations to Other Documents

This report is associated with the following ICONS research reports:

ICONS D01	The IST-2001-32429 ICONS Consortium, Intelligent Content Management System. Project Presentation, www.icons.rodan.pl , April 2002
ICONS D06	The IST-2001-32429 ICONS Consortium, Analysis and selection of the ICONS project research base, www.icons.rodan.pl , June 2002
ICONS D07	The IST-2001-32429 ICONS Consortium, Extracting Knowledge From Complex Content Objects into an Ontology Base with Logic Inference Capabilities, www.icons.rodan.pl , January 2003
ICONS D08	The IST-2001-32429 ICONS Consortium, Equivalence of UML semantic data model and the RDF content model, www.icons.rodan.pl , January 2003
ICONS D09	The IST-2001-32429 ICONS Consortium, Capturing procedural knowledge from process class definitions and from process instance execution measures, www.icons.rodan.pl , February 2003
ICONS D10	The IST-2001-32429 ICONS Consortium, a Multi-paradigm Integrated Knowledge Schema, www.icons.rodan.pl , February 2003.
ICONS D16	The IST-2001-32429 ICONS Consortium, Specification of the ICONS architecture, www.icons.rodan.pl , December 2002
ICONS D18	The IST-2001-32429 ICONS Consortium, Access algorithms and data structures underlying a distributed knowledge base, www.icons.rodan.pl , February 2003.
ICONS D25	The IST-2001-32429 ICONS Consortium, The Knowledge-based Content Management Application Design Methodology, www.icons.rodan.pl , to be completed.
ICONS D35	The IST-2001-32429 ICONS Consortium, The Structural Fund Project Knowledge Portal. Conceptual Design, www.icons.rodan.pl , December 2002

1.4 Intended Audience

The intended audience comprises all members of the ICONS project consortium as well as the representatives of the European Commission monitoring and evaluating the progress of the project research and development work.

1.5 Usage Guidelines

This report provides general information pertaining to ICONS external information integration features. It provides guidelines and constraints for the ensuing design and implementation work.

1.6 Notation Conventions

No special notation conventions are used in this report.

2. The ICONS Information Integration Requirements

Requirements pertaining to information integration features to be developed in the ICONS project are specified in the ICONS Project Description report [ICONS D01].

All entities, regardless of their character (structural, procedural), participating in the content integration process must be accessible via the knowledge map graph, or via other existing access path to the content repository. Any of the integrated content objects, constrained by the corresponding descriptions of the content repository schema, may either be physically stored in the repository as a content object (snapshot, re-freshable), or may be dynamically materialised at the reference time. Usage of the above integration modes should be entirely transparent to the KMS user.

Files

Files feature among candidates for content integration, due to the widely diffused usage of file systems as repositories of large, multimedia content objects. Little, or no, analysis of the multimedia objects content, apart from the automatic categorisation analysis, is performed during the integration process.

Data Bases

Heterogeneous databases are a typical source of data for content integration. Multi-database query and integration techniques, as well as the homogenization of heterogeneous data models, are the underlying technologies. The most straightforward cases entail querying a single database to materialise the required content to be further exploited in the KMS context, either as an element of a content object stored in the repository, as a virtual content object materialised on-the-fly.

Business Intelligence Systems

Data warehouses and OLAP system deliver relevant knowledge content, that should be integrated into the KMS environment. The BIS-generated content may be integrated into repositories as elements of content objects or may be delivered dynamically.

Legacy Information Systems

Similarly, the legacy information systems are the source of content that may be relevant to the KMS users. Selected legacy system reports may be accessible as content objects, or their elements, via the KMS content repository.

Intelligent Agents

Intelligent agent (IA) technology is a rapidly growing area of research and new application development. Applications of IA technologies in the KMS context are discussed in [Baek1999]. The definition of an intelligent agent proposed by IBM [IBM1995] states that an intelligent agent is "*a software entity that carries out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employs some knowledge or representation of the user's goals or desires*".

The IA technologies are clearly useful and applicable in the KMS context, meeting two broad functionalities, that of a **personal assistant** or that of a **communicating/collaborating agent**. In both roles the intelligent agents are relevant as knowledge-based support for the content integration features.

Document Management Systems

Document management systems are a particular class of legacy information systems providing a rich content infrastructure directly relevant to the KMS users. Electronic documents and image-based information typically integrated into the KMS content repositories as principal factual knowledge artefacts. In some KMS architectures the document management functionalities are subsumed by the KMS features.

Web Pages

Paradoxically, the genuine knowledge is perfectly hidden in the enormous amount of data volumes that is available on web pages. Therefore even more intelligent and flexible mechanism are to be developed in the area of external knowledge acquisition and, what is even more important, keeping it up-to-date. Interoperability of systems and ability to choose the best offered content are of the primary importance.

3. Overview of the ICONS Information Integration Features

The ICONS integration architecture (IIA) is presented in Figure 3.1. The focal point of the IIA is the ICONS Content Repository comprising content objects conforming to the multi-paradigm knowledge schema specification [ICONS D10]. Integrated data are stored as content object attributes of appropriate types. The respective fragment of the ICONS Multiparadigm Knowledge Schema is shown in Figure 3.2.

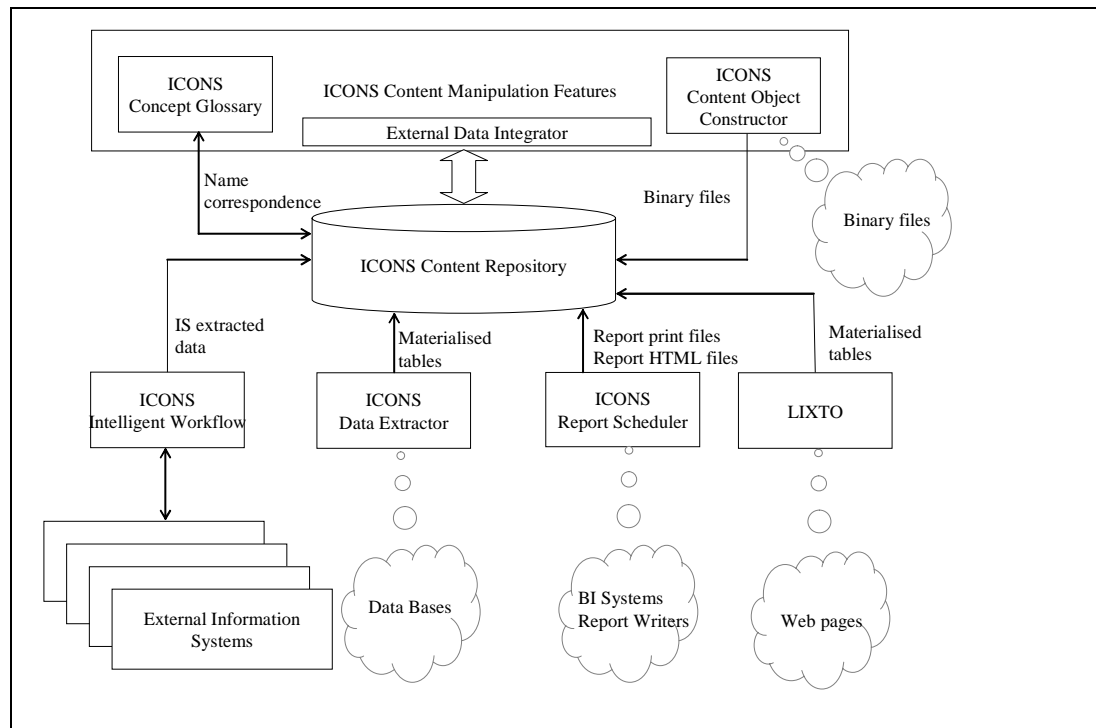


Figure 3.1. The ICONS Integration Architecture (IIA)

The ICONS Concept Glossary, based on the Topic Maps ISO standard [ICONS D10], provides a mechanism to enforce and maintain ontological consistency of integrated data objects. Consistency is maintained through the name correspondence constraint meaning that all MKS names must be defined in the Concept Glossary. The explanation feature is supported by meta-information comprised in the Concept Glossary data structure, namely the concept definitions and concept relationships.

The ICONS Content Object Constructor is integrated with the Content Presentation Manager and provides a facility for the system users to insert assorted binary files into content object occurrences. The binary files to be inserted are accessible on the users' workstation hard disc.

Although information integration has been the subject of several research projects [Arens1996, Calvanese1998, Hammer1995, Huhns1993, Hull1996, Jarke2000, Lattes1998, Levy1995, Ullman1997], in practice, an information integration system is usually realized in a pragmatic way. This is partially due to the fact that information integration is a highly complex problem, comprising aspects ranging from modelling to query processing, query optimization, inconsistency handling, etc. Comprehensive, formal design methodology for information integration and a set of coherent tools for the design and use of an information integration system are missing to date.

Our approach to ICONS materialised table integration entails two distinct integration steps; (1) data extraction step resulting in creation and storage in the ICONS repository of materialised tables, (2) integration of materialised tables extracted from different database views and/or web pages. Note, that databases need not be distinct. We have taken an application-oriented approach allowing the knowledge application developer to specify inferential methods of the Integration Object class to provide a mechanism of materialised table integration. The added value of our approach, although rather pragmatic, is to provision of declarative specification and execution of data integration rules.

The first issue in this context is the interpretation and merging of the data extracted from different sources. Interpreting data can be regarded as the task of casting them into a common representation. Moreover, the data returned by various sources need to be converted/reconciled/combined to provide the data integration system with the requested information. The complexity of this reconciliation step is due to several problems, such as possible mismatches between data referring to the same real world object, possible errors in the data stored in the sources, or possible inconsistencies between values representing the properties of the real world objects in different sources [Galhardas1999]. The above task is known in the literature as data cleaning and reconciliation [Bouzeghoub2001, Calvanese2001, Galhardas1999].

The ICONS Semantic Materialised Table Integration Environment (ISMTIF) has been developed and integrated with the ICONS platform to facilitate development of semantic data integration applications providing an additional layer of the ICONS information integration features.

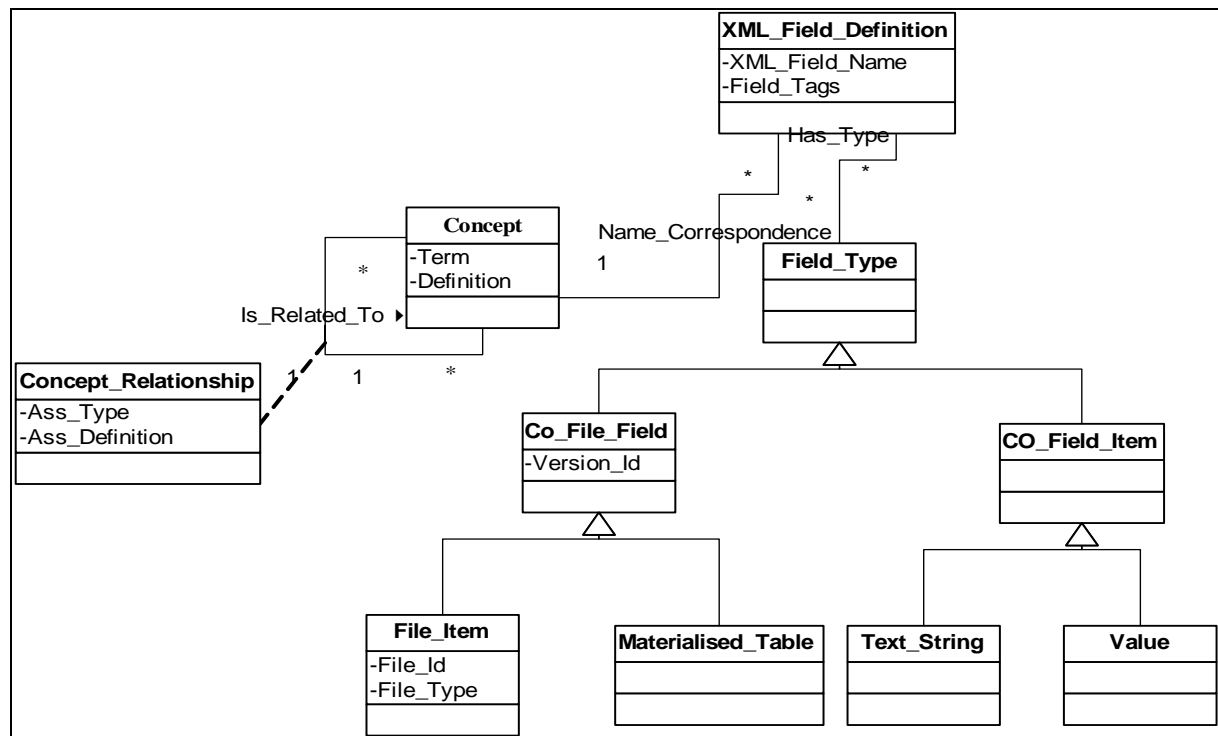


Figure 3.2. Content object classes and relationships pertaining to information integration

Materialised Table integration modes are controlled by the corresponding content object attribute persistency rules providing for the following materialised table types:

- **actual** materialised table representing a snapshot of an external data source valid at the report generation instance,
- **virtual** materialised table to be created at attribute reference time,
- **refreshable** materialised table is to be updated (recreated) at a specified time interval.

Semantics of the materialised tables are application dependent and as such may either be interpreted by the system user or by the corresponding content object methods.

Information integration with external information processing systems and databases may also be supported by third party data extraction and report generation tools. Such information may be stored in the ICONS repository to be subsequently accessible with the standard ICONS content object categorisation, selection and browsing features. A report scheduling facility OfficeObjects® Report Scheduler has been integrated with the ICONS platform to support required integration capabilities. This feature may be used to integrated information from external systems, such as business intelligence systems or transactional systems, to supplement content to be accessible as a knowledge management application.

4. ICONS Integration Control Structures

4.1 ICONS External Data Integrator Control Structures

The EDI control structures are presented in Figure 4.1 in the form of an un-attributed class association diagram. Below we describe semantics of the key object classes to illustrate the module integration capabilities

MaterialisationDefinition represents schema relevant information on materialisation procedure of any **materialised table** attribute type. There is one **materialisation definition** for one **materialised table** attribute.

Note: Actually there is no special type of ‘*materialisation table*’ in the content repository. The External Content Integrator assumes that if there is a *MaterialisationDefinition* defined for the ‘*binary file*’ attribute then this attribute should be treated as ‘*materialised table*’.

method defines type of an extraction method (currently there are three types available: *LIXTO Visual Wrapper*, *LIXTO Transformation Server* and *Data Extractor*).

persistency specifies whether materialisation is *actual* (materialised once), *virtual* (materialised on every field access) or *refreshable* (re-materialised on every time interval / periodical base).

description covers a detailed and instructive description of materialisation intention from the application point of view (this field is optional)

extractionParams string containing an XML document with formal extraction parameters (TODO: define format). The formal parameters during integration execution are replaced by actual parameters (values).

cron it is a string field containing cron definition for the scheduler (filled in only if refreshable extraction mode was chosen)

startDate is a scheduler start date (first extraction date) (filled in only if refreshable extraction mode was chosen)

MaterialisationExecution contains all additional information about an extraction execution process except for the result of extraction which is placed in the materialised table attribute. It stores all materialisation information on the base level. There is one and only one integration execution specified by a given integration definition and materialised table attribute belonging a given content object (the number of materialisation executions is stored in **matCount**).

lastMatDate date of the last materialisation. As process of materialisation can take a significant amount of time this date always points to a materialisation process beginning while materialisation duration time is stored in **lastMatDuration**.

The **result of the materialisation** which is a file is stored in an object.

External Data Source is the data source (database or web pages) used for data materialisation/extraction. It can be either a relational data base (extraction is performed by Data Extractor) or web page (extraction/integration is performed by LIXTO software). Integration definition may integrate from only one external data source while the same external data source may be invoked by many integration definitions.

Rel Database defines all necessary parameters allowing for connection to a given database via JDBC.

The precise description what is to be extracted from the relational database is stored in the parameterised **Extraction_Query**. The query can operate on tables or views (specified in **Table_Definition** that in turn is composed of **table columns**). Data is extracted only from **Extractable_Columns** while the query condition (the “where” clause) can use both **Extractable_Columns** and **Auxiliary_Columns**.

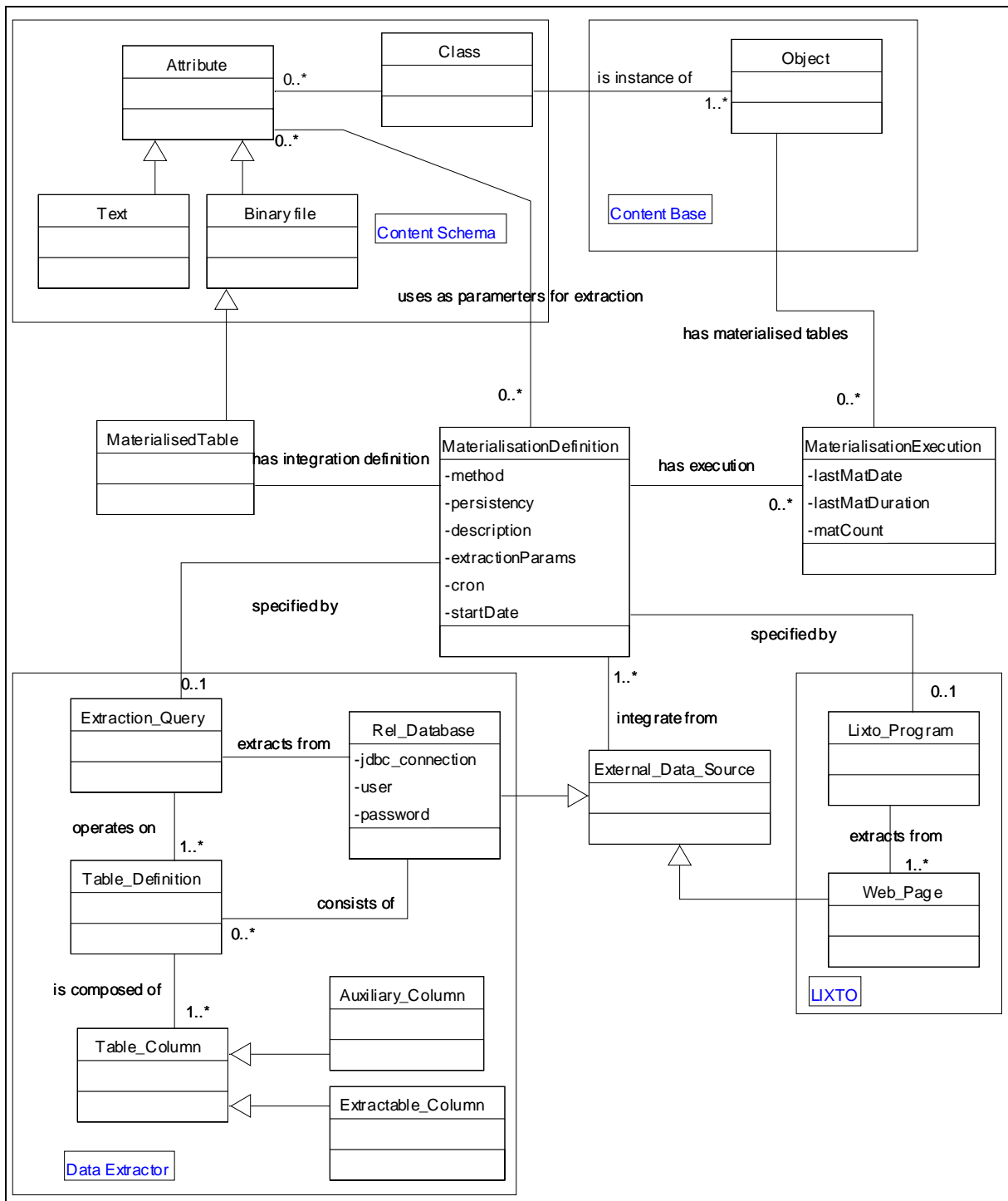


Figure 4.1. External Content Integrator – un-attributed class association diagram

4.2 ICONS Report Scheduler Control Structures

The EDI control structures are presented in Figure 4.2 in the form of an un-attributed class association diagram. Below we describe semantics of the key object classes to illustrate the module integration capabilities. Generated reports are stored in content objects, hence all ICONS features, notably data categorisation and indexing, may be used to support selection and browsing of the integrated reports.

The **External Source** may specify any pre-existing database, data warehouse, or data repository accessible via a third party report generation tool. Currently interfaced tools include Crystal Reports, Oracle Reporter, MapInfo

Professional, and the ICONS External Data Integrator. Appropriate access rights must be defined within the source data management systems for an ICONS user specifying a report.

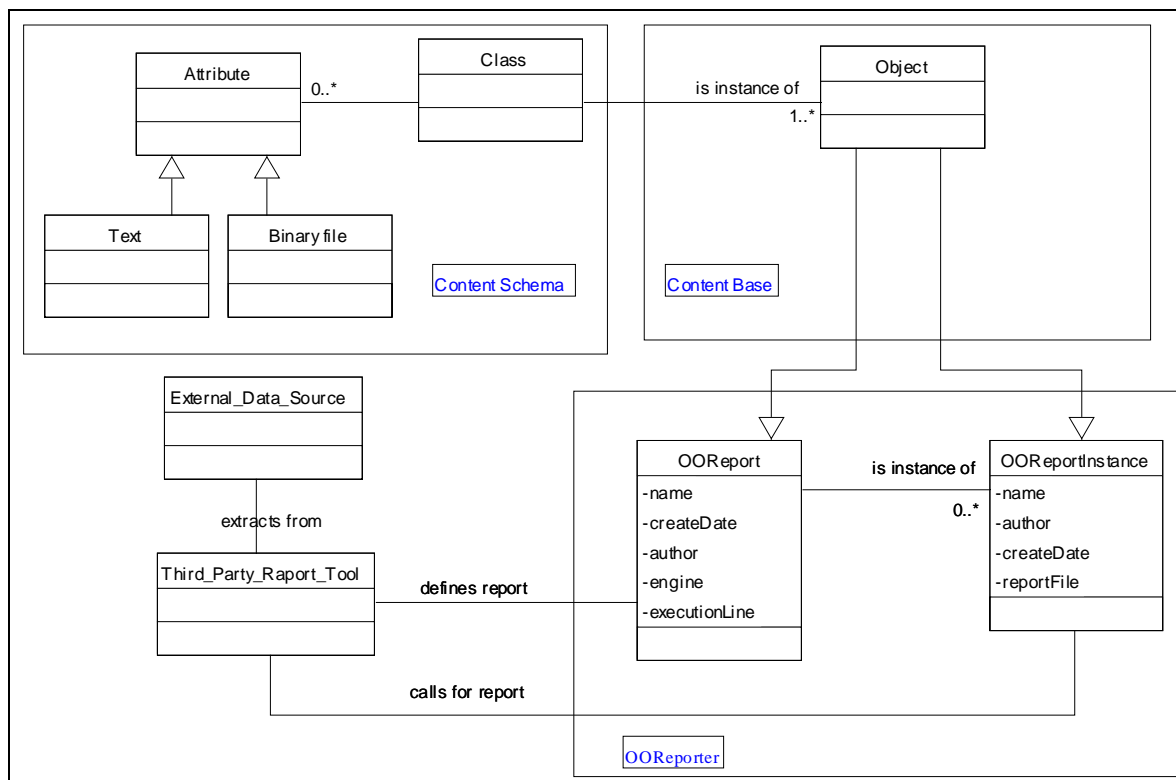


Figure 4.2. Report Scheduler – un-attributed class diagram.

The **OOReport** class represents a definition of a report type to be subsequently used by ICONS users to generate report files with the following key attribute semantics: **name** defines the report name or title, **createDate** is the date of report definition, **author** defines the report specifier, **engine** defines the third party report generation tool, and **executionLine** provides report parameters to be instantiated by the user at report execution time.

The **OOReportInstance** class objects represent instances of report files generated from External Sources with the following key attribute semantics: **name** defines the report name or title, **createDate** is the date of report generation, **reportFile** defines the report file address (usually the corresponding content object).

Auxiliary data structures required for the Report Scheduler operation are **report queues** providing asynchronous facilities to generate reports and to control the number of concurrent tasks for the third party report generation tools, as well as **report generation logs** storing required information pertaining to report generation activities.

5. ICONS Data Extractor

The ICONS Data Extractor (ICONS DE) provides the Query By Example (QBE) [Zloofxx] capability to integrate data from existing relational databases. Two principal screens of the ICONS DE module are presented in Figure 5.1 and Figure 5.2 representing respectively the table constructor functionality and the selection criteria specification window.

The **Table Constructor** window supports definition of the extraction target list (i.e. structure of the materialised table and auxiliary selection attributes), and specification of the source database and the relational (SQL) database view underlying the extraction query. The auxiliary attributes do not appear in the materialised table, they are only used as arguments of the selection criteria conditions corresponding to respective columns. The **Extraction** field provides the mapping of the materialised table to the content object attribute name.

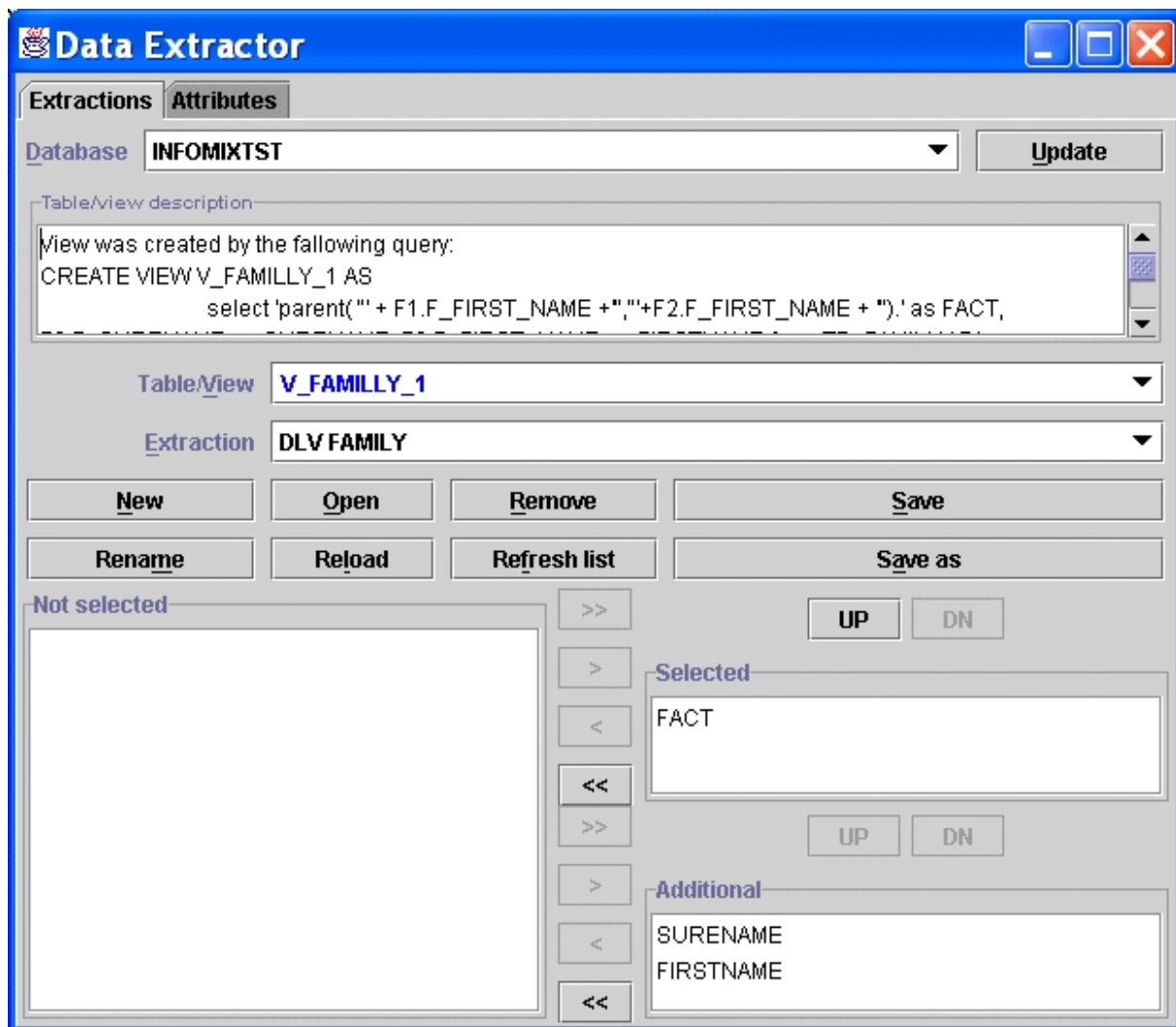


Figure 5.1. Data Extractor **Table Constructor** window.

The **Selection Criteria** window supports a classical QBE specification invoking a table specification defined with the use of the **Table Constructor** window by specifying the extraction name (usually corresponding the content object attribute name) and the selection criteria. The auxiliary materialised table columns are marked in pink background and they are dropped on completion of the table restriction process. Query parameters, i.e. names are preceded by the & sign, they are to be set by the system users at the extraction invocation instance with the use of an appropriate External Data Integrator feature.

The ICONS Data Extractor is to be invoked by the External Data Integrator or the Report Scheduler.

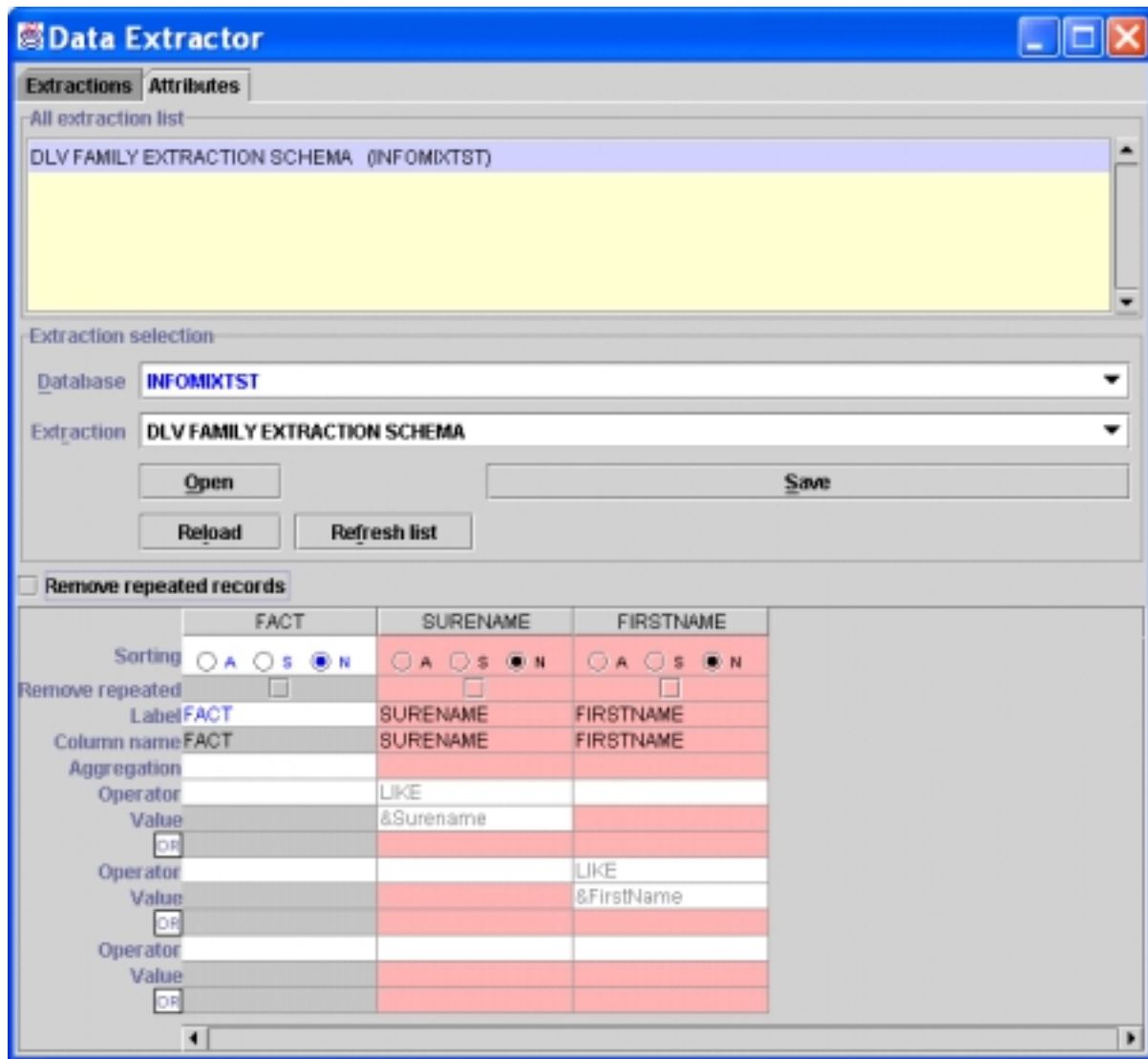


Figure 5.2. Data Extractor Selection Criteria window.

6. LIXTO Integration

The Lixto prototype [Baumgartner2001, Lixto2002] consists of two main blocks: The Wrapper Generator and the Program Evaluator. One module of the wrapper generator, the Interactive Pattern Builder, allows a wrapper designer to create and to store a wrapper in form of an extraction program (a program in the language Elog). Moreover, the wrapper generator contains the XML Translation Builder that allows a designer to specify how extracted data should be translated into XML format and to store such a specification in form of an XML translation scheme. The program evaluator automatically executes an extraction program (performed by the Extractor module) and a corresponding XML translation scheme (performed by the XML translator module) over Web pages by extracting data from them and translating the extracted data into XML format.

A wrapper is constructed by formalizing, collecting, and storing the knowledge about desired extraction patterns. Extraction patterns describe single data items or chunks of coherent data to be extracted from Web pages by their locations and by their characteristic internal or contextual properties. Extraction patterns are generated and refined interactively and semiautomatically with help of a human wrapper designer. They are constructed in a hierarchical fashion on sample pages by marking relevant items or regions via mouse clicks or similar actions, by menu selections, and/or by simple textual inputs to the user interface. A wrapper, in our approach, is thus a knowledge base consisting of a set of extraction patterns. While patterns are descriptions of data to be extracted, pattern instances are concrete data elements on Web pages that match such descriptions, and hence are extracted. Lixto distinguishes different types of patterns: Tree, string, and document patterns. Tree patterns serve to extract parts of documents corresponding to tree regions, i.e., to subtrees of their parse tree. String patterns serve to extract textual strings from visible and invisible parts of a document (an invisible part could be, e.g., an attribute value such as the name of an image). Document patterns are used for navigating to further Web pages. Logical Organization of Patterns. The logical organization of an extraction pattern is as follows: each extraction pattern has a name and contains one or more so-called filters. Each filter provides an alternative definition of data to be extracted and to be associated with the pattern. The set of filters of a pattern is interpreted disjunctively (i.e., connected by logical ORs). Each filter is associated to a parent pattern from which it extracts the desired information. Tree (string) patterns are specified via tree (string) filters. A tree filter contains a representation of a generalized parse tree path that matches a set of items on a Web page, and contains a set of conditions that these items must satisfy. All the conditions of a filter are interpreted conjunctively, i.e., an element of a Web page satisfies a filter if and only if it matches its generalized tree path and satisfies all the conditions of the filter. Similarly, a string filter specifies the characteristics of the text to be extracted (using a formal language), and possibly further conditions.

Lixto offers a wrapper designer the possibility to express various types of conditions restricting the intended pattern instances of a filter. The main types of conditions are inherent (internal) conditions, contextual (external) conditions, and range conditions. In addition to these three basic types of conditions, Lixto allows a designer to express auxiliary conditions like pattern reference conditions, concept conditions and comparison conditions. They are discussed as atoms of the Elog language in more detail in Section 3.

Extraction patterns are defined by the designer in a hierarchical manner. A pattern that describes an entire document is referred to as a document pattern. In particular, the document pattern corresponding to the starting Web page, the so-called "home document pattern", is available as a preexisting pattern. Other patterns are defined interactively. Filters or patterns are usually defined in the context of other patterns (so-called parent patterns). For example, a pattern <name> may be defined first, and then patterns <firstname> and <familyname>, etc., may be defined in the context of the source pattern <name>. For the majority of common extraction tasks, defining at patterns or a strict hierarchy of patterns will in practice be sufficient. However, Lixto does not limit the pattern definition to be strictly hierarchical (i.e. tree-like).

Moreover, pattern definitions are allowed to be recursive (similar to recursive type definitions in programming languages). While patterns are not required to form a strict hierarchy, pattern instances do always form one and can be arranged as a tree (or forest, in case they stem from different documents, which can be the case in recursive programs).

The visual and interactive pattern definition method allows a wrapper designer to define an extraction program and an associated XML translation scheme without any programming efforts. The Lixto Interactive Pattern Builder allows a wrapper designer to define filters and patterns with the help of one or more characteristic example pages, and to modify and store patterns. At various intermediate steps, the designer may test a partially or fully constructed filter or pattern, both on the example pages used to construct the pattern as well as on any

other Web page. The result of such a test is a set of pattern instances, which is displayed by a browser as a set of highlighted items.

The filter description procedure for tree-filters can be described as follows:

The designer marks an initial element on an example Web page (for example, a table). The system associates with this element a generalized tree path of the parse tree that (possibly) corresponds to several similar items (for example, several tables). The designer then tests the filter for the first time. If more than just the intended data items are extracted (and thus highlighted) as a result of the test, then the designer adds restrictive conditions to the filter and tests the filter again. This process is repeated as long as undesired data items are extracted. At the end of the process, the filter extracts only desired items. A similar procedure is used for designing string filters. However, for creating a string rule usually no example is selected, but some characterizations are visually composed, e.g. by relying on concept conditions. A pattern is designed by initially asserting one filter for the pattern, and, in case this is not sufficient (because testing shows that not all intended extraction items on the test pages are covered), by asserting successively more filters for the pattern under construction, until each intended extraction item is covered by at least one filter associated to that pattern.

Observe that the methods of filter construction and pattern construction correspond to methods of definition-narrowing and definition-broadening that match the conjunctive and disjunctive nature of filters and patterns, respectively. It is the responsibility of the wrapper designer to perform sufficient testing, and if required by the particular application-test filters and patterns also on Web pages different from the initially chosen example pages. Moreover, it is up to the wrapper designer to choose suitable conditions that will work not only on the test pages, but also on all other target Web pages. The visual and interactive support for pattern building offered by Lixto also includes specific support for the hierarchical organization of patterns and filters. A wrapper definition process according to Lixto (and consequently, a Lixto wrapper) is not limited to a single sample Web document, and not even to sample Web pages of the same type or structure. During wrapper definition, a designer may move to other sample Web pages (i.e., load them into the browser), continuing the wrapper definition there.

7. ICONS Materialised Table Semantic Integration

The ICONS Semantic Materialised Table Integration Framework (ISMTIF) is to be used by ICONS knowledge management application developers for development materialised data integration algorithms supported by declarative DLV content object methods. The data integration scenarios are schematically represented as a Use Case diagram shown in Figure 7.1.

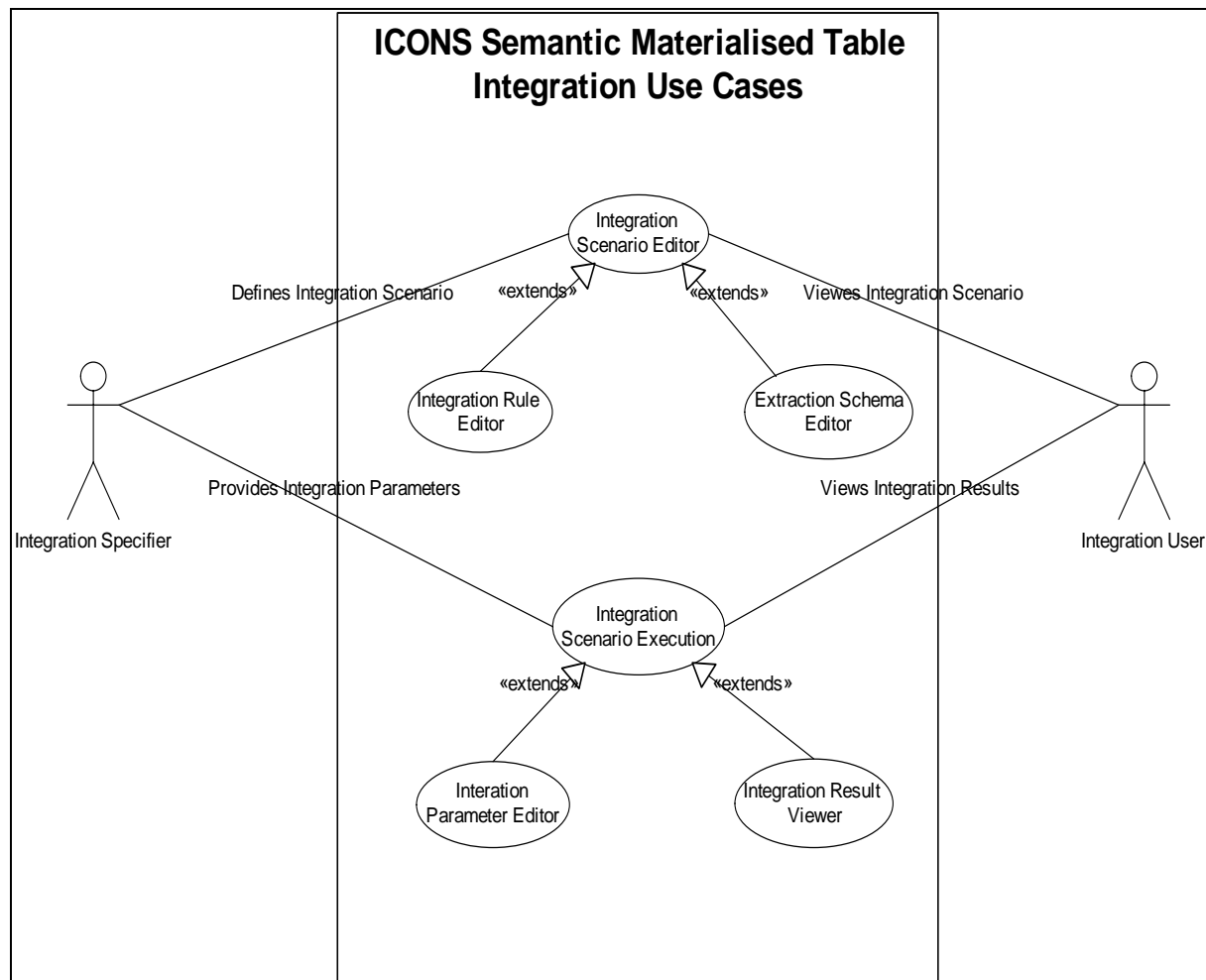


Figure 7.1. ICONS semantic materialised table integration use cases.

Naturally, in the ICONS platform environment the roles of an Integration Specifier and an Integration User may be performed by the same person. However, the skills and the corresponding system authorizations may be different.

The Integration Specification comprises definitions of the Extraction Schema, providing specification of the external data sources and the appropriate extraction rules (relational view, parametric QBE expressions), and of the Integration Rules specified in Disjunctive Datalog. The Extraction Schema controls materialization of relational tables that are to provide the “fact base” for the DLV integration algorithm (program).

Integration result is to be available for viewing in the form of a relational table to be stored in the corresponding content object occurrence and appropriately presented in the ISMTIF graphic user interface.

All integration content, namely the extracted data, the integration rules, the extraction rules and the corresponding QBE parameter values, as well as the integration result, are to be persistently stored in the ICONS repository in a content object occurrence. Standard version control may be applied to integration scenarios represented by the ICONS content objects.

A content object class, called the Integration Object class, may be defined to specify the required integration methods and the content object structure specified as the XML schema. Instances of the Integration Object class may represent different integration scenarios or to reduce the content structure complexity, specialization classes of the Integration Object class may be defined for distinct integration scenarios. The QBE data extraction parameters may be defined at the integration scenario run time by appropriately modifying values of the corresponding content object fields.

The ISMTIF has been developed as a collection of Java classes representing the Integration Object class methods. The ICONS Content Base Manager API as well as the required functional modules, such as the QBE Data Extractor, the Text Editor, the Content Object View Server, etc are used to develop the system architecture of the ISMTIF system is shown in Figure 7.2.

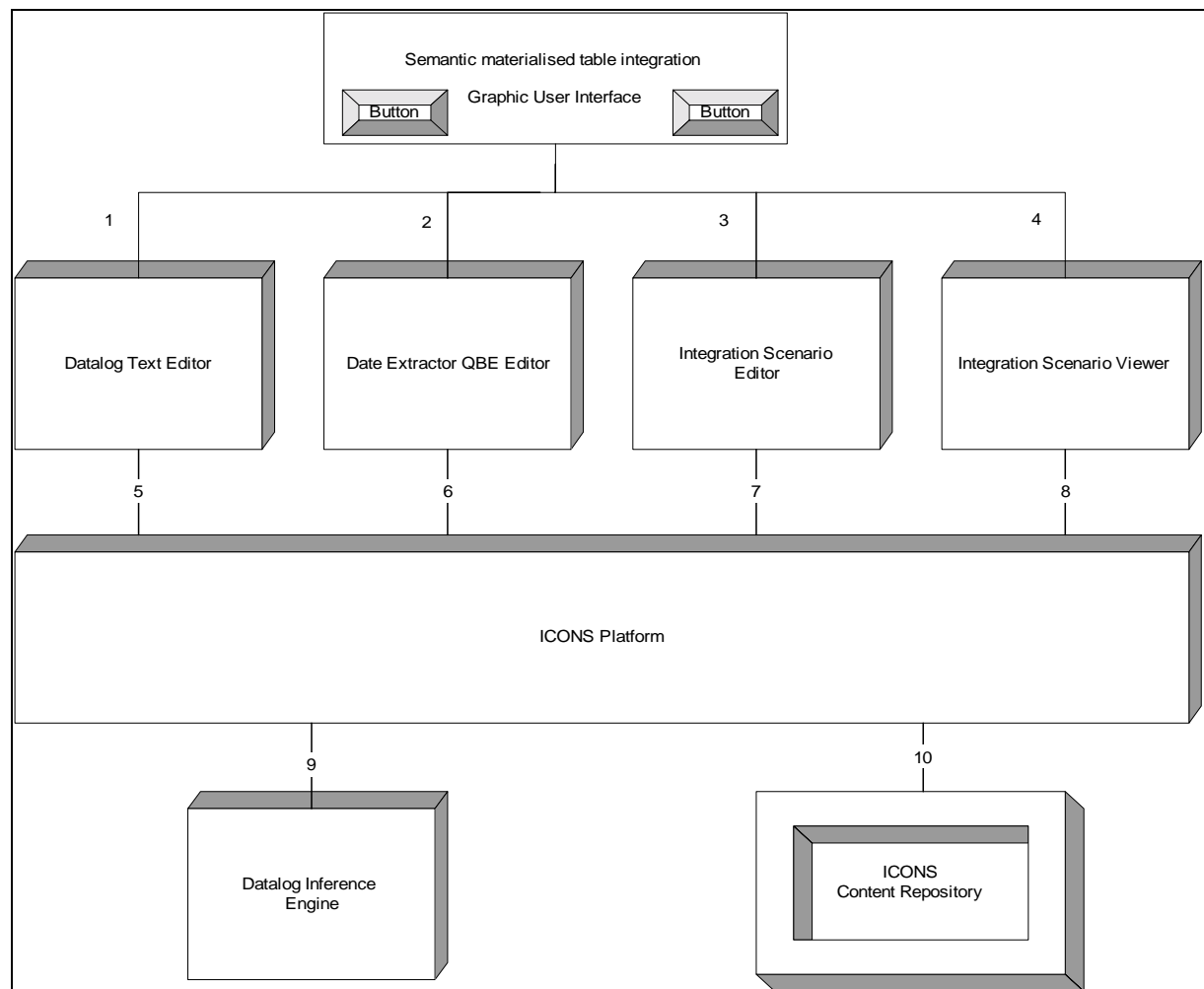


Figure 7.2. Architecture of the Semantic Materialised Table Environment

All ISMTIF features are accessible via the graphic user interface implemented with the use of the JSP (Java Server Pages) technology and executed in an Internet browser environment. The data and control flows among the ISMTIF modules are numbered and have the following functionality:

1. The Datalog Text Editor supports writing, viewing and modification of Datalog rules to be stored as text fields in the Integration Object class instance.
2. The Data Extractor QBE Editor is the ICONS feature and it provides a specialized graphic user interface to define, view and modify definitions of QBE tables. The QBE table specifications may be referenced via the Materialized Table attribute.
3. The Integration Scenario Editor provides means to specify the Materialized Table fields, already defined in terms of a parameterized QBE table specification, the QBE table definition parameter values specified as references to the Integration Object instance fields, the Integration Object instance text field containing the

- corresponding Datalog rules, and the target Materialized Table or File Item field to store the integration result.
4. The Integration Scenario Viewer provides features to view integrated relational tables, the integration results, as well as to inspect all corresponding integration scenario information.
 5. The Datalog rules to be integrated with the appropriate fact table to construct a DLV program performing the corresponding integration scenario are persistently stored in an Integration Objects instance text field. Any number of Datalog rule text fields, to store distinct rule sets, may be persistently stored in an instance of an Integration Object class.
 6. The QBE materialized table specifications are stored in the appropriate control structures of the ICONS Portal and they are referenced via the corresponding Materialized Table field of an Integration Object instance.
 7. The Integration Scenario Editor stores all provided scenario parameters in the corresponding complex Integration Scenario control field, defined for the Integration Object class by the corresponding XML schema. Many such fields may be persistently stored within an Integration Object instance.
 8. The Integration Scenario Viewer retrieves contents of the requested Integration Object instance fields to present the scenario to the ISMTIF system users.
 9. The complete Disjunctive Datalog program, comprising the Datalog rules and the corresponding fact tables, is passed to the DLV Module via the standard DLV import functions, and the results are returned via the standard DLV export functions.
 10. The Integration Object instances are stored within the ICONS Portal Repository. Appropriate meta-data is retained in the ICONS Portal Repository semantic index to facilitate materialization of the required index trees providing navigational access to the Integration Object class instances.

The materialized external content, as well as the integration scenario execution result, are stored in the corresponding "Co_File_Fields" as a reference to a data file stored within the ICONS HSM hierarchical memory structure. The ICONS Repository version control features manage version control that may also apply to the integrated content. In the case of virtually materialized tables the corresponding data file reference is only valid during the "content object use" life-cycle and it is not persistently stored in the Content Repository.

The Concept Glossary objects are indicated in the partial repository schema to indicate the option of providing the name correspondence between the concepts representing the user view of the Universe of Discourse and the Materialized Table column names. In our opinion such name correspondence could provide a useful "semantic bridge" for users inspecting specifications of data to be integrated, the integration results, as well as the integration rules. Nevertheless, we do not enforce the "name correspondence constraint in the ISMTIF platform and the Integration Object class field naming conventions are at the Integration Specifier's discretion.

The structure of the ISMTIF module supporting execution of integration scenarios is show as an un-attributed CAD in Figure 7.3. The indicated objects are either stored as control structures in the meta-information areas of the ICONS Repository or they represent classes to be defined in the repository schema.

Integration of structured data is to be supported by the QBE Data Extractor using the corresponding data structures comprised in the schema. The semi-structured data requires a specialized internet crawler. It has been decided by the ICONS consortium that LIXTO is to be interfaced with the ISMTIF system to provide the required external data materialization. We assume that the extracted semi-structured content is to be mapped into the relational table format to subsequently be integrated into the DLV program.

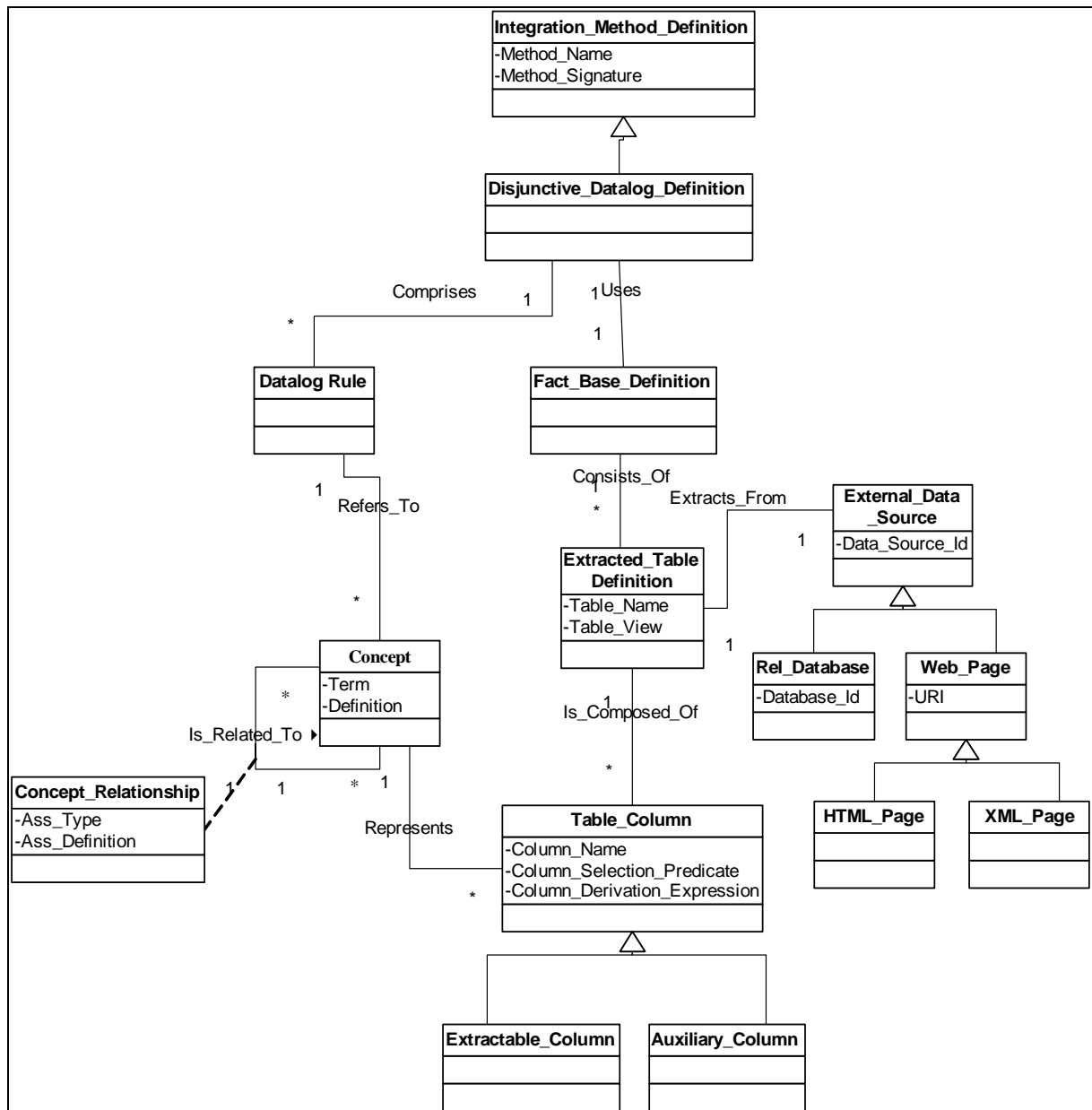


Figure 7.3. Semantic materialised table integration class association diagram

8. ICONS Intelligent Workflow based integration.

In order to assure the system flexibility, the ICONS Intelligent Workflow (IWf) implements the *WorkFlow Management Coalition* (WfM Coalition) standards. The ICONS IWf architecture is compliant with the WfMC's reference model ([WfMC-TC-1011]) and its process model is based on the WfM Coalition's process meta-model ([WfMC-TC-1003]). In *ICONS* every business process is defined in XML Process Definition Language (XPDL, [WfMC-TC-1025]). In addition, communication between different managers (engines) is based on Wf-XML interoperability binding protocol ([WfMC-TC-1023]).

8.1 The Intelligent Workflow core functionality

The core functionality offered by the ICONS IWf according to the WfM Coalition's standards includes:

- definition workflow tasks (activities) and relationships between them (transitions),
- running a workflow process (process instance) in favor of a piece of information (data container),
- assignment users (workflow participants) or applications agents (i.e. a piece of specialized software) to activities. Workflow activities can be executed by workflow participants, their groups, roles that the participants play, organizational units, or application agents.
- work sharing and parallel execution – often a given activity is divided into separate tasks that are executed in consecutive workflow tasks. In order to minimize time of execution such tasks, they (or some of them) can be executed in parallel. The system supports workflow participants in parallel activity/task execution. In addition the participants can check the status of a given workflow process at any time. According to the WfMC terminology, the system supports XOR-SPLIT and AND-SPLIT operations.
- work joining and task synchronization – analogous to work splitting, there is a need to join executed tasks. The join operation should take into consideration such aspects as merging information that come from different tasks and different time of finishing these tasks. The system process management functions enable tasks to be joined according to the above requirements. A workflow participant that joins several tasks is informed about appearing new tasks as they come. According to the type of join operation, activity execution can be done after all previous activities have been finished or at least one of them. For example, if a department director shares a work of making a review of a proposal amongst two reviewers, he/she will be informed about finishing this work by each of them as it comes. Eventually, the director accepts/rejects the proposal after both reviews came. According to the WfMC terminology, the system supports XOR-JOIN and AND-JOIN operations.
- execution of workflow activities conditionally – sometimes there is a need to execute a task only if a condition is satisfied. Such condition depends on processed information (i.e. its attributes) and process flow data. For example acceptance of a credit proposal can depend on the value of the requested credit. the system enables workflow to execute workflow tasks conditionally.
- definition of execution time for the process and its activities – in a case of rigorous time restrictions on process execution there is a need to control execution time of the whole process or individual activities. This can be useful to respond immediately to process/activity delay. The time modeling features support processes in definition and controlling of execution time.
- calling external application – the system allows to define that an activity is to invoke an external application.

Moreover, the ICONS IWf system extends implementation of WfMC standards of:

- definition of the dynamic workflow participant assignment – besides workflow participant assignment defined by WfM Coalition's standards, there is a need to extend it of dynamic aspects. An example of these dynamic aspects is expressed in the following assignment: 'this task will be executed by a person that prepared the proposal' – that is a workflow participant that executed the first task. In addition, *ICONS* makes possible to define whether a given task can be executed by one or all assigned workflow participant candidates (cardinality). The proposition how to express dynamic aspects in workflow participant assignment is presented in [Momotko2002].
- 'ad-hoc' decision – sometimes, especially when processes can be modified in the future and the modifications are hard to predict at present, there is a need to have an 'ad-hoc' decision. This kind of decision enables dynamic selection of a workflow participant that will execute a given activity from a list of participant candidates.
- time management – the systems provides features to both to notify users of delays that occurred as well as those that may occur in the future. In addition, it is also possible to check, if a delayed activity may also delay the whole process. The algorithm to calculate possible deadlines and durations is based on works presented by Eder [Eder2001].

- exception handling – during system maintenance some exceptions can occur. An example is an inappropriate workflow participant assignment, which may stop process execution. the system has several mechanisms that support exception handling:
 - default workflow participant assignment – when a list of participant that are obliged to execute a given activity is empty, this activity is executed by the participant attached to the default assignment,
 - transactional execution of activities – owing to using a transaction processing system to restore data, the system is able to execute activities in the transactional mode. (i.e. all modification or nothing),
 - process termination – there is a possibility to terminate process. The system saves information about activities and transitions that have been executed up to the termination moment.
- Flexible definition of Event-Condition-Action rules (ECA rules) – an effective workflow management system should be able to react to external events that come during process instance execution. An example of such events is process/activity delay. Reaction for such external events can be modeled by ECA rules. ECA rule defines that if an E event occurs and the C condition is satisfied; an action will be executed. In future, ECA rules will be attached to the workflow process definition.

8.2 The ICONS Workflow process integration

One of the most challenging features of Workflow Management Systems (WfMS) is workflow process interoperability. Such interoperability enables two or more workflow engines to communicate and work together to co-ordinate their work. General workflow integration models are presented in Figure 8.1

There are several different models of workflow co-operation, namely: the chained process model, the nested sub-process model, and the parallel synchronised model.

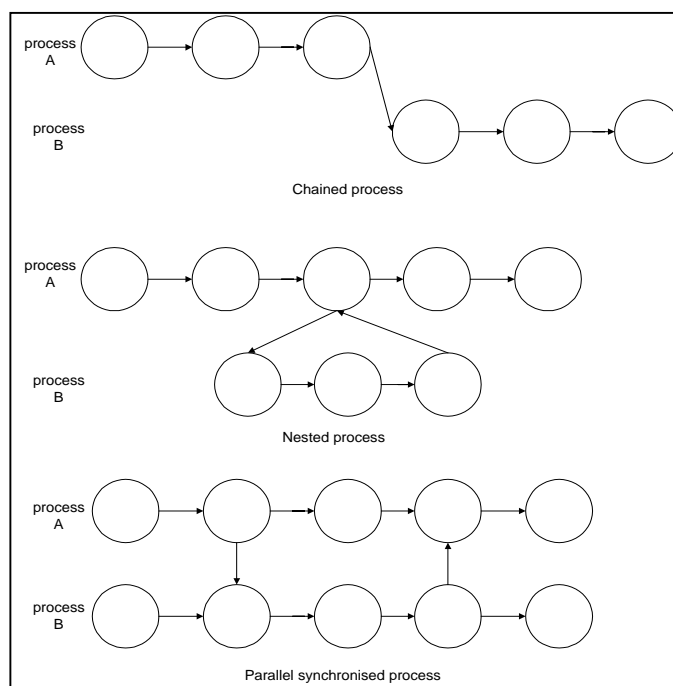


Figure 8.1. Models of workflow co-operation.

In the chained process model after one workflow process is completed, another workflow process inherits the processing and starts. This is the most basic model. In the nested sub-process model, one workflow process has a part of its processing done by another workflow process. In the parallel synchronised model, two workflow processes that are proceeding independently become synchronised at some point and exchange information, and then continue independently. When an activity reaches the synchronisation point, it waits for the other to arrive there, and then they exchange information.

On the basis of the WfMC's reference model, and the Interface 4 standard described in [WfMC1996], the Object Management Group (OMG) had developed JointFlow specification. JointFlow defines a framework for

distributed workflow applications in the world of business objects ([OMG1998]). This specification enables interoperability of workflow process components, monitoring and workflow execution, and association of workflow components to resources involved in a workflow process. In the next step a simple workflow access protocol (SWAP) has developed. SWAP was envisioned as a binding of the jointFlow object model and related WfMC standards to an HTTP-based interaction protocol. Finally, in 1999, WfMC has presented the Wf-XML specification. This specification enhances some of its predecessors' capabilities, providing:

- a structured and well-formed XML body protocol that consists of message containing headers and data
- logical interact model with synchronous, asynchronous, and batch capabilities
- independence from transport mechanisms
- easy extensibility through the use of XML and dynamic workflow context data.

In a synchronous messaging a process A can may wish to initiate a sub-process and suspend its normal processing until that sub-process completes. In an asynchronous messaging, the initiating process sends a request to the enacting process. The enacting process then sends only an acknowledgement back to the initiator, informing that the request has been received. At some later point in time, the enacting process sends a response to the initiating process. The initiating process sends then an acknowledgement back to the initiator, informing that it received the response. In the batch messaging it is possible to place multiple Wf-XML interaction in a single message.

In the ICONS project we implement Wf-XML specification and the SOAP protocol transported and use HTTP to transport XML workflow messages.

8.3 The ICONS Workflow information system integration

One of the fundamental assumptions of ICONS is the possibility to integrate with an independent external system. Such system should provide the ICONS IWf with at least with the following information:

- users and organizational structure,
- structure of information objects (application data),
- services/functions offer by the information management system (refers further to as IM system) to support a given kind of information object (e.g. proposal registration, proposal acceptance, etc.).

The first ICONS IWf requirement is to deliver by the IM system information about workflow participants. This information is used to assign the participants to appropriate activities (tasks).

Each workflow process is attached to an information object (IO). Such object can be complex and includes attributes that are other information objects. For example, a case can be connected with a given client. A given case can be comprises of a set of information objects that represent proposals received from/sent to the user.

The last system requirement is information about services/functions that are supported by the IM system for a given information object. When a given activity is started, the system executes appropriate service for a given information object instance. A service/function can implement by the IM system as:

- OLE object (Windows environment),
- Java class (Unix and Windows systems).

Bibliography

- Arens1996 Arens, Y, Knoblock, C. A., and Shen. W., Query reformulation for dynamic information integration. *J. of Intelligent Information Systems*, 6:99–130, 1996.
- Baek1999 Baek, S., Liebowitz, J., Prasad, S.Y., and Granger, M., Intelligent Agents for Knowledge Management – Toward Intelligent Web-Based Collaboration within Virtual Teams, in *Knowledge Management Handbook*, J. Liebowitz (Ed.), CRC Press LLC, 1999.
- Baumgartner2001 Baumgartner, R., Flesca, S., Gottlob, G., Declarative Information Extraction, Web Crawling, and Recursive Wrapping with Lixto, *Proc. of VLDB 2001*
- Bouzeghoub2001 Bouzeghoub, M., and Lenzerini, M., Special issue on data extraction, cleaning, and reconciliation. *Information Systems*, 2001.
- Calvanese1998 Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., and Rosati, R., Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13, 1998.
- Calvanese2001 Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., and Rosati, R., Data integration in data warehousing. *Int. J. of Cooperative Information Systems*, 10(3), 237–271, 2001.
- Eder2001 Eder, J.; Panagos, E., Managing Time in Workflow Systems, *Workflow handbook 2001*.
- Galhardas1999 Galhardas, H., Florescu, D., Shasha, D., and Simon, E.. An extensible framework for data cleaning. Technical Report 3742, INRIA, Rocquencourt, 1999.
- Hammer1995 Hammer, J., Garcia-Molina, H., Widom, J., Labio, W., and Yue Zhuge, The Stanford data warehousing project. *IEEE Bull. of the Technical Committee on Data Engineering*, 18(2):41–48, 1995.
- Huhns1993 Huhns, M.N., Jacobs, N., Ksiezyk, T., Wei-Min Shen, Singh, M.P., and Cannata, P.E., Integrating enterprise information models in Carnot. In *Proc. of the Int. Conf. on Cooperative Information Systems (CoopIS-93)*, pages 32–42, 1993.
- Hull1996 Hull, R., and Gang Zhou, A framework for supporting data integration using the materialized and virtual approaches. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 481–492, 1996.
- IBM1995 IBM, Intelligent Agent Strategy, White Paper, (<http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm>), 1995.
- ICONS D01 The IST-2001-32429 ICONS Consortium, Intelligent Content Management System. Project Presentation, www.icons.rodan.pl, April 2002
- ICONS D06 The IST-2001-32429 ICONS Consortium, Analysis and selection of the ICONS project research base, www.icons.rodan.pl, June 2002
- ICONS D07 The IST-2001-32429 ICONS Consortium, Extracting Knowledge From Complex Content Objects into an Ontology Base with Logic Inference Capabilities, www.icons.rodan.pl, January 2003
- ICONS D08 The IST-2001-32429 ICONS Consortium, Equivalence of UML semantic data model and the RDF content model, www.icons.rodan.pl, January 2003
- ICONS D09 The IST-2001-32429 ICONS Consortium, Capturing procedural knowledge from process class definitions and from process instance execution measures, www.icons.rodan.pl, February 2003
- ICONS D10 The IST-2001-32429 ICONS Consortium, a Multi-paradigm Integrated Knowledge Schema, www.icons.rodan.pl, February 2003.
- ICONS D16 The IST-2001-32429 ICONS Consortium, Specification of the ICONS architecture, www.icons.rodan.pl, December 2002
- ICONS D18 The IST-2001-32429 ICONS Consortium, Access algorithms and data structures underlying a distributed knowledge base, www.icons.rodan.pl, February 2003.
- ICONS D25 The IST-2001-32429 ICONS Consortium, The Knowledge-based Content Management Application Design Methodology, www.icons.rodan.pl, to be completed.
- ICONS D35 The IST-2001-32429 ICONS Consortium, The Structural Fund Project Knowledge Portal. Conceptual Design, www.icons.rodan.pl, December 2002
- Jarke2000 Jarke, M., Lenzerini, M., Vassiliou, Y., and Vassiliadis, P., editors, *Fundamentals of Data Warehouses*. Springer-Verlag, 2000.
- Lattes1998 Lattes, V., and Rousset, M-C., The use of CARIN language and algorithms for

- information integration: The Picisel project. In Proc. of the ECAI-98 Workshop on Intelligent Information Integration, 1998.
- Levy1995 Levy, A.Y., Srivastava, D., and Kirk, T., Data model and query evaluation in global information systems. *J. of Intelligent Information Systems*, 5:121–143, 1995.
- Lixto2002 Lixto Visual Wrapper User Manual, The Technical University of Vienna, 2002
- Momotko2002 Momotko, M., Subieta, K., Dynamic change of Workflow Participant Assignment, *Advances in Database Information Systems, ADBIS'2002*, Slovakia, Bratislava, 2002.
- Ullman1997 Ullman, J.D., Information integration using logical views. In Proc. of the 6th Int. Conf. on Database Theory (ICDT'97), volume 1186 of *Lecture Notes in Computer Science*, pages 19–40. Springer-Verlag, 1997.
- WfMC_TC_1003 Workflow Management Coalition, The workflow reference model, WfMC-TC-1003, issue 1.1, Jan 1995.
- WfMC_TC_1011 Workflow Management Coalition, Terminology & Glossary, WfMC-TC-1011, version 3.0, Feb 1999.
- WfMC_TC_1023 Workflow Management Coalition, Interoperability Wf-XML Binding, WfMC-TC-1023, version 1.0, May 2000.
- WfMC_TC_1025 Workflow Management Coalition, Workflow Process Definition Interface – XML Process Definition Language, WfMC-TC-1025, draft 0.03a, May 2001.