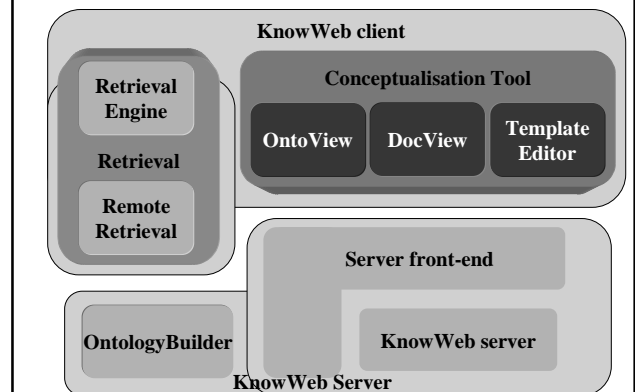


## Project KnowWeb

- KnowWeb: Web in Support of Knowledge Management in Company
- European Research project Esprit No. 29065
- December 1998 – June 2000
- 6 partners
  - Technical University Košice, Slovakia (R&D leader)
  - Luton Business School, UK
  - University of Vaasa, Finland
  - plus 3 companies (from Finland, UK and Slovakia)

## KnowWeb architecture



## KnowWeb Server functionality

- **Management** of Domain model;
  - Domain model browsing
  - Introduction of new concepts, relations
  - Modification of DM
- **Storage** of shared data - Domain model, Association links, source code of documents, stored procedures, etc.
- Various **retrieval** functions on server-side data (concepts, associations, documents)
- **Input to:**
  - **Administration** of server modules
  - **Statistics** and supporting functions for domain modelling

## Functionality of OntoBuilder tool

- Defining, changing, and deleting Domain Model items, namely:
  - **Concepts** → Classes and Instances,
  - **Attributes** of a Concept → Type and Value of attributes,
  - **Relations** between Concepts → *Subclass\_Of* and *Instance\_Of* relations,
- Saving/Loading current DM into/from shared database (stored on the KnowWeb Server),
- **Retrieval (Exact and Approximate)** in the space of DM concepts,
- DM administration and statistical functions.

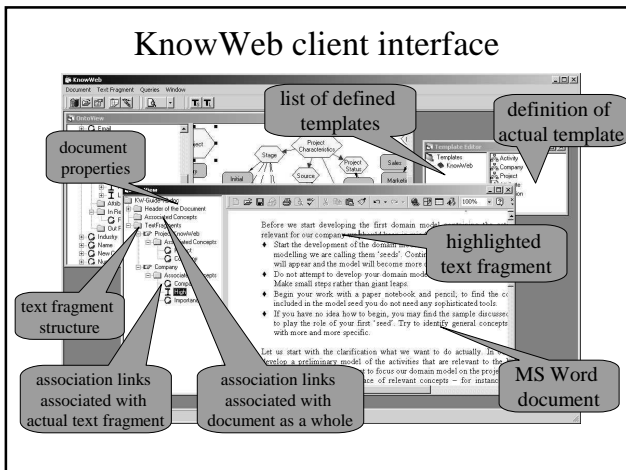
## OntoBuilder, Web interface

The screenshot shows the OntoBuilder web interface within a web browser. Callouts highlight several key features:
 

- Web browser with KnowWeb WWW interface:** Points to the browser window.
- Searching capabilities:** Points to the search bar.
- Editing capabilities:** Points to the ontology graph.
- Highlighted Instance:** Points to a specific instance in the graph.
- View customisation:** Points to the graph's layout options.
- On-line help:** Points to the help icon.
- Contents of the WWW page:** Points to the left-hand navigation menu.
- Some statistical information:** Points to the status bar at the bottom.

## KnowWeb client functionality

- Viewing and browsing current ontology
- Opening and possibly editing documents from KnowWeb database or from a URL
- Creating textual fragments in documents
- Linking documents and fragments to concepts
- Managing templates
- Query management
- Retrieving documents



### Support for different formats of files

- Fully implemented support for text fragment definitions and linking of textual documents in:
  - HTML format and
  - MS Word format
- Moreover, within the above mentioned kind of documents can be selected and linked also information in form of:
  - tables,
  - pictures,
  - multimedia embedded objects etc.

### Support for different formats of files (2)

- Any other type of information sources can be linked as whole (so called referenced documents) and later on retrieved e.g.:
  - spreadsheet files,
  - database files,
  - graphic files,
  - multimedia files,
  - executable files.

### Automatic linking

- Mostly based on templates (predefined set of rules for concepts' linking)
- Template Editor supports design of templates with rich set of tools for definition of linking rules
- System interface to be used for automatic linking by means of any EDMS has been implemented
- Interface tested for use by CONTACT2000

### Template Editor

- Templates can be grouped within a tree structure of categories
- Templates can be composed from:
  - concepts
  - attribute values (concept with equivalent name is looked for)
  - conditions based on attribute values

### Condition definition example

## System interface for automatic linking

- There are 4 functions defined for use from outside KnowWeb (for any EDMS).
1. Function `IntroduceToKnowWeb` taking two arguments:
    - path/URL of document
    - name of template
 → If document does not exist in DB, then it is saved. Then the template is applied on it and resulting concepts are linked to document.

## System interface for automatic linking (2)

2. Function `LinkThroughTemplate` with 2 arguments:
  - ID of document
  - ID of template/name of template
 → The template is applied on existing document and resulting concepts are linked to it.
3. Function `LinkToConcept` with these arguments:
  - link name
  - ID of document
  - ID of concept
 → Links specified concept to existing document.

## System interface for automatic linking (3)

4. Function `UpLoadDocument` taking these arguments:
  - path/URL of document
  - ID
  - name
  - title
  - author, owner
  - purpose
  - type
 → Saves document into DB.

## Types of retrieval

- Retrieval in document space
  - **Case-based retrieval** - search using knowledge model (i.e. by list of concepts associated with particular document),
  - Full-text search,
- Retrieval in ontology (DM concepts) space
  - **Exact retrieval** - search by attributes of concept and/or of relations,
  - **Approximate retrieval** - search by relations between concepts.

## Retrieval in ontology space - Conditions

**Condition** - 'meta-concept'; it can be expanded into a set of concepts, which satisfy some constraints.

E.g. "all the concepts, which are in the 2-neighbourhood of the concept named as "Company".

The screenshot shows a dialog box titled "Concept Conditions: Formulation and Management". On the left, there is a tree view under "Available Concept Conditions" with a root "Concept Conditions Root" and sub-items like "Companies", "Intra Company", "Classes", "Instances", and "Instances". A callout points to this tree: "List of available Conditions (and its items)". In the middle, there is a "New Condition" panel with buttons for "New Item", "Delete", "Clear", "Save To DB", "Refresh From DB", and "Apply". A callout points to the "Apply" button: "Highlighted Condition". On the right, there is a table titled "Resulting List of Concepts" with columns "Name", "Type", and "Relevance". The table contains the following data:

Name	Type	Relevance
Company	Class	1
Mactel, s.r.o.	Instance	1
Maccor Trade, a.s.	Instance	1
AKA Netlog, a.s.	Instance	1
Fodok.	Instance	1

A callout points to this table: "Resulting set of concepts".

## Retrieval in document space – Queries

**Query** - tree-like structure of *Items* and *Sub-Queries*, which can be expanded into the ordered set of documents.

E.g. "Find all the documents related to "Project" or "Activity", or related to something like "Company".

### Possible Items of Query :

- Single *Class* concept
- Single *Instance* concept
- FullText query string
- Condition
- Condition Item
- another *Query*

### Parameters of Query :

- Logical conjunctions of the items (AND / OR)
- Polarity (YES/NO)
- Importance degree (1-10)

## Managing Queries, standard interface

List of available Queries (and its items)

Highlighted Query

List of available Queries (and its items)

Resulting set of documents

Preview of resulting documents

## Managing Queries, Web interface

List of available Queries

Web browser with KnowWeb WWW interface

Selected Query

Commands for editing the Query

Contents of the WWW page

String representation of the Query

## Know Web Pilot Application

- Linking of Winpos sw with Know Web
- Converting Winpos reports to html format
- Testing in simulated retail chain environment
- Evaluation of results and feedback information to Universities

## Know Web Pilot Application

- Extending Winpos => Corporate memory
- Tracing of exceptional situations and disturbances
- Domain model for Retail Chains
- adocs - Automatically generated documents ( full integration with Winpos )
- mdocs - Manually generated documents ( know Web as such )

## Automatically generated documents, adocs

- End-of-day batch initiated from Reg or PXZ
- Cashier just clicks the "end-of-day button" => htm-files are generated
- Automatic linking to Know Web concepts using templates

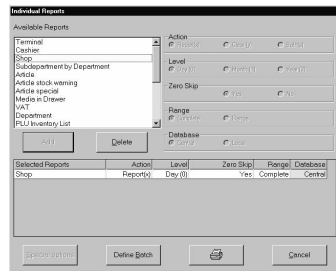
## Serial reports in INPOS

- Individual Reports grouped into serial reports
- Each individual report may have several concepts
- All linked to Know-Web using a single click

## Report examples in



- Individual Reports (PXZ)
- Batchreports (eg. End of Day report)
- Different levels (Day, Month, Year)
- Output to: A4 printer, Receipt printer, Ascii-file, html-file => Know Web



## Automatically generated documents, adocs

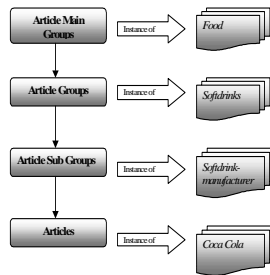
- Automatically create .Dat-files when running reports in Winpos
- Dat-file = Structured textfile
- Automatically create HTML-files from .Dat-files
- The operator simply run the report "normally" in Winpos

TYPE	SUB	BESC	DNT	101	101 ALL	PROGALL	TEST151	TEST152
HEAD		Complete		811	22.2.2000		09:05	X
TEXT								
TEXT		Sales						
TEXT		Gross Sales		578,28	578,28	578,28		
TEXT		Discounts		0,00	0,00	0,00		
TEXT		Returns		0,00	0,00	0,00		
TEXT		Trans. sold		0,00	0,00	0,00		
TEXT		NET Sales		578,28	578,28	578,28		
TEXT		Customers		11	11	11		
TEXT		Trans.		28	28	28		
TEXT		Ang. sale per customer		51,84	51,84	51,84		
TEXT		Dep. items per customer		2,8	2,8	2,8		
TEXT		Departments		0,00	0,00	0,00		
TEXT		Vines		208,28	208,28	208,28		
TEXT		Beers		152,18	152,18	152,18		
TEXT		Other drinks		82,28	82,28	82,28		
TEXT		Total		578,28	578,28	578,28		
TEXT		Estimated gross Profit		0,00	0,00	0,00		
TEXT		Media in drawer						
TEXT		OSDF		578,28				
TEXT		HISC		0,00				
TEXT		Loans						
TEXT		Pick-up		0,00				
TEXT		Rec. on account		0,00				
TEXT		Paid out		0,00				
TEXT		Home Curr. Sales		578,28				
TEXT		Foreign Curr. Sales		0,00				
TEXT		TOTAL		578,28				
TEXT		ENDALL		56				

A HTML-file will be automatically generated from Winpos report Dat-file.

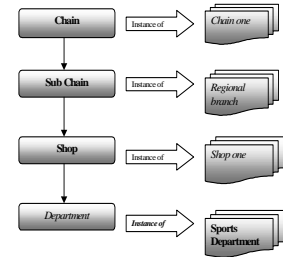
## Domain model sample: Item structures

- Hierarchy-structure
  - + MainGroup
  - + Department
  - + Subdepartment
  - + Article
- Direct sale on department possible
- "Alcohol - Vines - Red Vines - Cato Negro"



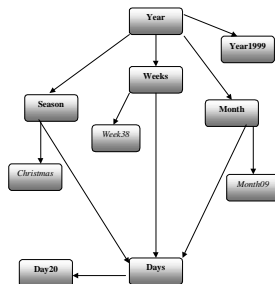
## Domain Model Sample: Chain structures

- Hierarchy-structure => Possible to get exact reports for the chain
- Unique numbers for Chain, Subchain, Shops => Possible to validate data for reports (eg. Sales for only Shop 1)



## Domain model Sample: Time structure

- Time-related information structured
- Possibility to compare two time intervalls (eg. Christmas 99 & 00)
- "A Year consists of Weeks - which consists of Days"



## Disturbance - Exceptional situations

- Exceptional situations that might affect sales
- Technical
- Environmental
- Personnel

