REPRO: Supporting Flowsheet Design by Case-Base Retrieval

Jerzy Surma, Bertrand Braunschweig
Artificial Intelligence Group
Computer Science and Applied Mathematics Division
Institut Francais du Petrole
1 et 4, av. de Bois-Preau, 92506 Rueil-Malmaison, France
Email: {Jerzy.SURMA│Bertrand.BRAUNSCHWEIG}@ifp.fr

Case-Based Reasoning (CBR) paradigm is very close to the designer behavior during the conceptual design, and seems to be a fruitable computer aided-design approach if a library of design cases is available. The goal of this paper is to presents the general framework of a case-based retrieval system: REPRO, that supports chemical process design. The crucial problems like the case representation and structural similarity measure are widely described. The presented experimental results and the expert evaluation shows usefulness of the described system in real world problems. The papers ends with discussion concerning research problems and future work.

1 Introduction

This paper presents a prototype of a case-based retrieval system named REPRO (**R**eutilisation d'**E**tudes de **Pro**cedes) which aims to support a designer through the earliest steps of designing petroleum and chemical processes. REPRO retrieves the best matching flowsheet: Process Flow Diagrams (PFDs) from a library of existing PFDs by computing how they are attributionally and structurally similar to a complete or partial input diagram under preparation. Once the most similar existing flowsheet has been selected, it may be used for:

- completing the partial PFD;
- retrieving other information associated with the reference PFD (such as Piping and Instrumentation Diagrams **-** P&IDs);
- retrieving other data, e.g. steady-state simulation models, operating conditions etc. which may constitute a good start for the preparation of specific elements for the new process being designed.

PFDs represent complex processes, that can be defined as systems that generate required output products from a given set of input feedstocks under appropriate operating conditions. These processes can be represented at different levels of abstraction corresponding to different phases in the design (Douglas 1988):

- conceptual (synthesis) phase: only the most important functions, connections, and input/output streams are represented in Block Diagrams;
- basic engineering design phase: all equipments, connections, streams, some operating conditions (pressure, temperature, flow rates) are represented in PFDs; additional detailed information such as the control systems, utilities, sensors are represented in the P&IDs; equipment lists are established;
- detailed engineering design: all process-specific information is supplied, including dimensions, materials etc.

In this paper we address both conceptual and the basic engineering design phases where block diagrams and PFDs are used. Drawing a flowsheet is not a simple task. Some PFDs for standard refining processes need several days of designer time and many pages of fine print quality. Designers use highly specific CAD software equipped with time-saving devices such as palettes of equipments. However, when preparing a new process, the designer does not like to start from scratch and always tries to use an existing design as a starting point for the next one (adding and removing process items as needed). The set of existing diagrams in a company, the corporate memory, is a good candidate for case-based retrieval as designers are neither able nor willing to look for the most similar diagram, due to the complexity of the cases and to time constraints.

Computing attributional and structural similarity between complex diagrams such as the PFDs needs to carefully address the complexity aspects, because this similarity implies to compare graphs made of typed nodes and typed links and because most algorithms for graph comparison are NP-complete. In our approach the complexity question is solved by using teleological information, that is defining the function of each equipment in addition to factual data such as the equipment type and connectivity. This functional information saves a lot of computation but implies more input from the user. The paper extends (Surma & Braunschweig 1996) by showing the detailed structural similarity calculation and some results on a relatively complex process example.

At present there is considerable growing interest in the use of CBR in design activities. An overview of the original attempts in case-based design systems has been given by Pu (Pu 1993). A recent general overview the state of art in the field is presented by Maher et al. (Maher et al. 1995). The comprehensive analysis of the computer aided design and presentation of the most important research problems is deeply presented in the FABEL Consortium publications (Voss et. al 1994a, 1994b).

In section 2 we present the object-oriented flowsheet representation. Then in section 3 the two structural similarity measures for flowsheets are introduced. In section 4 the REPRO system is shortly described and in section 5 the evaluation results are presented. The paper concludes with discussions concerning research problems and future work.

2 Flowsheet Representation

The research in knowledge representation for process engineering (Stephanopoulos 1987, Motard 1989, Fraga 1994) suggest object-oriented paradigm as a useful way of representing process information. We adopted this approach (Surma & Braunschweig 1995) and finally the following classes were defined:

- the generalization taxonomy of flowsheet components: the component taxonomy (e.g., Reactor).
- the class defining pipes (stream) connections between components: the class PIPE.
- the generalization taxonomy of available chemical processes: the process taxo-xonomy (e.g., Hydrogenation C3).

The instances in the process taxonomy are aggregation taxonomies that represent the flowsheet itself: the flowsheet aggregation (e.g. Flowsheet-1). Aggregation is the relationship (transitive and antisymetric) in which objects representing the components are associated with an object representing the entire assembly (Rumbaugh et al., 1991). Figure 1 illustrates the process taxonomy, and the component taxonomy that is the source of components for a flowsheet aggregation. For example a Flowsheet-1 consists of a Pump-1, Reactor-1, and Pipe-1.

In the component class the following slots concerning case retrieval are defined and are inherited by all the subclasses:

- local-root defines the type for the given object. The objects which belongs to the same local-root are the same kind of objects. For example in fig.1 the local-root for the Pump-1 and Pump-2 is a Pump. In general the local-root defines the sub-tree in generalization taxonomy and is used by similarity measure, for instance by the Most Specific Common Abstraction strategy during local components comparison. The local-root of the component x will be denoted: local-root(x).
- function defines the role (the task) of a given component in a flowsheet aggregation. The function of the component x will be denoted: function(x).
- quantity defines how many identical components perform the function specified in the function slot.
- weight is a real value $\in$ <0;1> that indicates the importance of a given components in the whole assembly. The value of a component x weight will be denoted: weight(x).

Additionally each subclass of the component class has domain-specific slots and methods for computing local similarity between two instance components. Thanks to polymorphism the proper domain specific formula is used.

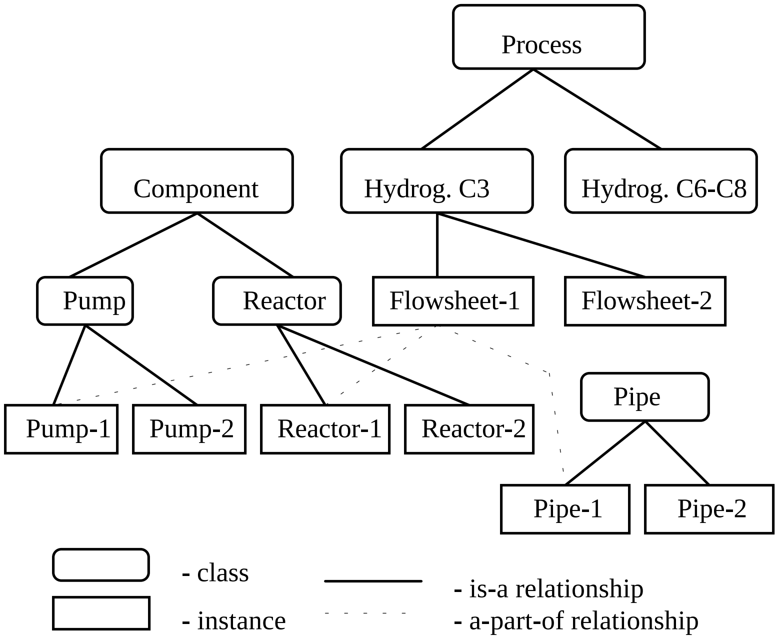


Fig.1. The object-oriented flowsheet representation.

3 Structural Similarity Measure

The similarity measure for flowsheets should interpret them as assemblies of components where the crucial factors are the internal components properties and the pipes connections between components. This mean that in opposite to the most CBR systems the similarity measure for flowsheets should be based not only on the local structure of the objects but on the relational structure existing between them as well. Several approaches were made in this field including: analogical reasoning (Holyoak & Thagard 1989, Falkenhainer et. al. 1989), conceptual graphs (Myaeng & Lopez-Lopez 1992, Maher 1993), and object-oriented representation (Bisson 1995). In CBR the mentioned FABEL project investigates several approaches to structural similarity based on: graph theory, term-based representation, gestalt indices and psychological theories of perception (Voss et al. 1994a).

The flowsheet might be treated as a graph, that implies the use one of the know approaches, which are mostly NP-complete. But the availability of the domain knowledge and the specific nature of the flowsheets allow to create a flowsheet-specific measure that is computationally acceptable. Thanks the local-root slot the set of possible matching components is narrowing to the same type ones.

Additionally the function slot selects among the same type components that are responsible for the same task. In fact in each flowsheet (after careful function labeling) there are no two or more of the same type components performing the same function: the uniqueness property. The practical utilization of this property and the role of the quantity slot is discussed in Surma and Braunschweig paper (Surma & Braunschweig 1996). The uniqueness property fundamentally reduces a complexity of the matching process, but requires deep knowledge acquisition. The similarity measure based only on the same type components is under consideration. One of the most promissing approaches for this problem is described in the Bisson paper (Bisson 1995).
Based on the expert suggestions two similarity measures between the input flowsheet aggregation (Ψ_I) and the retrieved flowsheet aggregation (Ψ_R) were created. First, the aggregation similarity treats the flowsheets as a set of components. Second, the connection similarity takes into account connections (pipes) between components. The aggregation and connection similarity measures are normalized, and might be used separately or by a linear combination.

3.1 Aggregation Similarity

The aggregation similarity focuses on the components and on internal (attributional) similarity between common components. Let X be a set of all instances in the component taxonomy, $S_I = \{ x \in X \mid x \text{ a-part-of } \Psi_I \}$, $S_R = \{ x \in X \mid x \text{ a-part-of } \Psi_R \}$, and $\Omega = S_I \cap S_R$ this means: $\Omega = \{ (x1 \in S_I, x2 \in S_R) \mid \text{local-root}(x1)=\text{local-root}(x2) \wedge \text{function}(x1)=\text{function}(x2) \wedge \text{weight}(x1) > 0\}$. The aggregation similarity between the input and the retrieved flowsheets is:

$$\mathrm{SIM}^A(\Psi_I, \Psi_R) = \frac{\mathrm{Card}(\Omega) * \dfrac{\sum_{(x1,x2)\in\Omega} \mathrm{weight}(x1) * \mathrm{sim}(x1,x2)}{\sum_{(x1,x2)\in\Omega} \mathrm{weight}(x1)}}{\mathrm{Card}(S_I) + \mathrm{Card}(S_R) - \mathrm{Card}(\Omega)}$$

where: $\mathrm{sim}(x1,x2) \in <0;1>$ is the similarity between component x1 and x2 computed on the slot (attributes) level.

This measure has the following properties:if for all $(x1,x2)\in\Omega$: $\mathrm{sim}(x1,x2)=1$ then: $\mathrm{SIM}^A(\Psi_I ,\Psi_R) = \mathrm{Card}(S_I \cap S_R) / \mathrm{Card}(S_I \cup S_R)$, if $\Omega = \varnothing$ then $\mathrm{SIM}^A(\Psi_I ,\Psi_R) = 0$, possibility of using recursively for the nested aggregation taxonomies, and antisimetricy because each argument of the SIM^A function plays a different role (the weights are taking only from an input aggregation).

3.2 Connection Similarity

The connection similarity focuses on the pipe connections between components. Let Y be a set of all instances in the class PIPE, $L_I = \{ y \in Y \mid y \text{ a-part-of } \Psi_I \}$, $L_R = \{ y \in Y \mid y \text{ a-part-of } \Psi_R \}$, in(y) be a component connected at the input of a pipe y, out(y) be a component connected at the output of a pipe y, $\Phi = L_I \cap L_R$ this means: $\Phi = \{ (y1 \in L_I, y2 \in L_R) \mid (in(y1),in(y2)) \in \Omega \wedge (out(y1),out(y2)) \in \Omega \}$, $sub(\Psi_I, \Psi_R) = \{g \mid g$ is the common subgraph between Ψ_I and Ψ_R defined by the elements of a set $\Phi\}$, and $L_g = \{ y \in Y \mid y \in g \}$. The connection similarity between the input and retrieved flowsheet is:

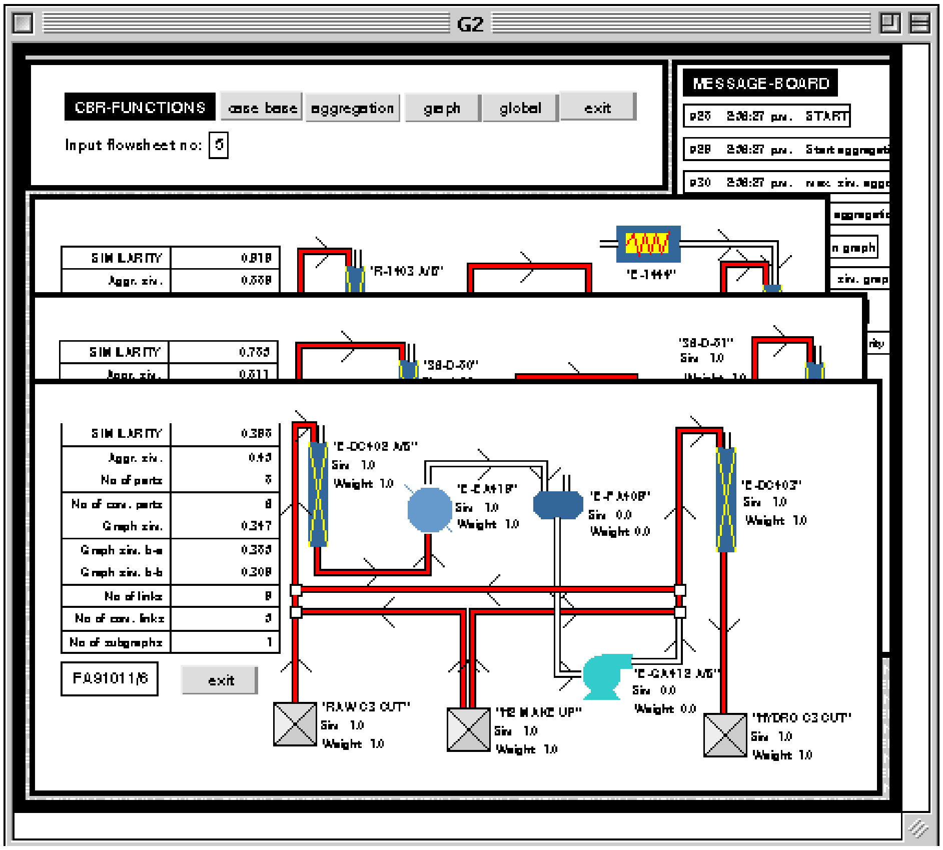
$$SIM^C(\Psi_I, \Psi_R) = \beta\alpha * \frac{Card(\Phi)}{Card(L_I) + Card(L_R) - Card(\Phi)} + \beta\beta * \frac{\sum_{g \in sub(\Psi_I, \Psi_R)} Card(L_g)^2}{Card(L_I)^2}$$

where: $\beta\alpha, \beta\beta \in <0;1> \wedge \beta\alpha + \beta\beta = 1$.

This measure has the following properties: if the common subgraph between Ψ_I and Ψ_R is identical to the graph Ψ_I then: $SIM^C(\Psi_I, \Psi_R) = Card(L_I \cap L_R) / Card(L_I \cup L_R)$, if $\Phi = \emptyset$ then $SIM^C(\Psi_I, \Psi_R) = 0$. The coefficient $\beta\alpha$ concerns the first part of the connection similarity, that interprets connections as a set of pipes. The $\beta\beta$ coefficient concerns the second part of the formula which support the expert heuristic: ''it is better to have small number of large common subgraphs than a large number of small ones''. Surprisingly this heuristic supports Gentner's systematic principle (Gentner 1993): " A predicate that belongs to a mappable system of mutually interconnecting relationships is more likely to be imported into target than is an isolated predicate".

4 Case Retrieval in REPRO

The knowledge representation and structural similarity for flowsheets introduced in the previous sections have been implemented in the REPRO system (Surma 1996). The main task of REPRO is the case-based retrieval of flowsheets based on the aggregation and/or connection similarity. The system has been implemented on a SUN 10 Sparc workstation with the G2 expert-system development environment. G2 provides an excellent object-oriented representation, a high-level programming language, and a flexible graphical environment. An example of the REPRO user interface is shown in fig.2 and a schematic layout is shown in fig.3.



Input flowsheet no: 5

```
#26   2:38:27 p.m.   START
#28   2:38:27 p.m.   Start aggregati
#30   2:38:27 p.m.   max sim. aggr
```

SIMILARITY	0.818
Aggr. sim.	0.838

SIMILARITY	0.783
Aggr. sim.	0.811

SIMILARITY	0.383
Aggr. sim.	0.43
No of parts	3
No of corr. parts	8
Graph sim.	0.347
Graph sim. b-a	0.333
Graph sim. b-b	0.308
No of links	8
No of corr. links	3
No of subgraphs	1

Fig.2. REPRO user interface.

The system consists of four modules: a graphical user interface (GUI), a case manager (CM), a case retrieval module (CR), and a case base / background knowledge repository (CB/BK).

Thanks to the CM module the user can easily define an input flowsheet by selecting the desired components and connecting them graphically with pipes. Additionally CM provide facilities like browsing, deleting, and editing existing flowsheets. A case manager is able to put into the case-retrieval module a draft (incomplete) input flowsheet, so the user (designer) has an opportunity to focus on desired details. After defining an input flowsheet, the CR module is ready for retrieval in the case base in order to find proper existing designs. It starts with automatically selecting the set of flowsheets from the case base. During this preliminary step a set of proper cases (from the same process class) is established.

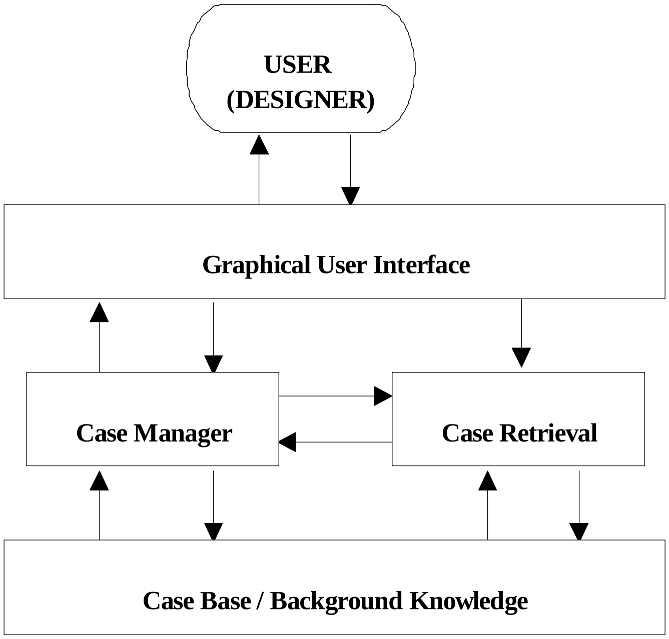


Fig.3. REPRO architecture.

Then the structural similarity between the input flowsheet and the given retrieved one is computed from the formula:

$$SIM(\Psi_I, \Psi_R) = \alpha * SIM^A(\Psi_I, \Psi_R) + \beta * SIM^C(\Psi_I, \Psi_R)$$

where: $\alpha, \beta \in <0;1> \wedge \alpha+\beta=1$

Necessary domain knowledge is taken from the CB/BK repository: for example a components-specific local similarity measure. The value of α and β coefficients are specified by the user, who can focus on the required kind of similarity. After computing the similarities, the CR module ranks the cases and return results to CM. Once relevant cases are retrieved from the case base, the designer can browse those cases in order to select the most applicable ones for the current situation. As it was mentioned in section 4 this approach is computationally tractable because of the flowsheets uniqueness property. In fact this complexity is O(NM) for the aggregation similarity and $O(N^2M^2)$ for the connection similarity in the worst-case performance, where N, M are the number of the components respectively in the input and the retrieved flowsheet.

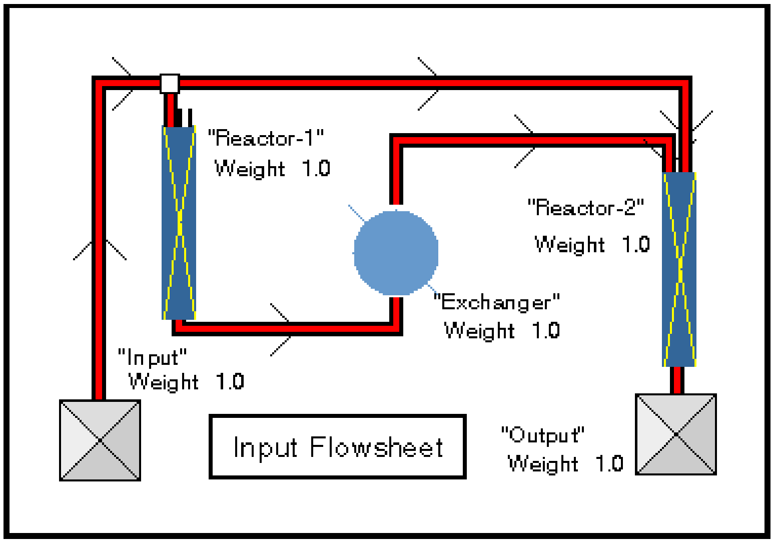
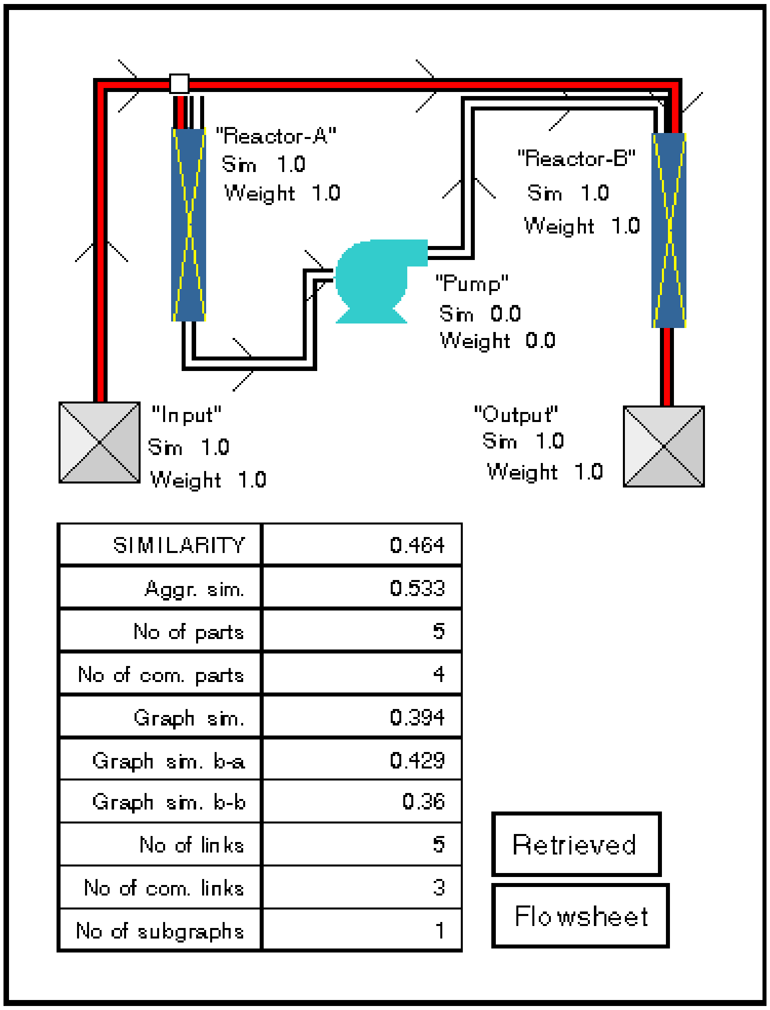


Fig.4. An example of an input flowsheet.

The following example shows the performance of REPRO from the user's point of view. Figure 4 presents a simple input flowsheet that consists of 5 connected components. Each component has a weight that expresses the user preferences. Figure 5 presents an example of a retrieved flowsheet. The results of comparison between an input and a retrieved flowsheets are shown to the user on three levels. First, the local similarity between components is shown next to each component on a retrieved case: sim(x1,x2), see fig.5. Second, the connection similarity is graphically shown by displaying the common pipes with a different color: sub(Ψ_I ,Ψ_R), see fig.5. Finally the general quantitative description of the similarity (computed from formulas introduced in sections 3.1 and 3.2) is presented in fig.5 as well. Thanks to this textual and graphical information the designer has a whole overview of the local and global similarity between flowsheets.

5 REPRO evaluation

REPRO was tested on flowsheets for the hydrogenation C3 and hydrogenation C6-C8 processes. The overview of the case base is presented in table 1. The hydrogenation C3 process was selected for introductory tests as a relatively simple and standard type of process. On the contrary the hydrogenation C6-C8 is a complex type of process with a large number of the interconnections. Before being introduced in the case base, all the available flowsheets were especially modified by excluding unimportant control links and by adding function descriptions to all the components. Unfortunately this task required the expert supervision for all the flowsheets.



SIMILARITY	0.464
Aggr. sim.	0.533
No of parts	5
No of com. parts	4
Graph sim.	0.394
Graph sim. b-a	0.429
Graph sim. b-b	0.36
No of links	5
No of com. links	3
No of subgraphs	1

Fig.5. An example of a retrieved flowsheet.

Finally the retrieval results and weights for all the available 27 flowsheets were obtained from the expert. Based on this data, REPRO was tested by means of a ”leave-one-out” method. For the C3 process the system retrieved the proper cases without error. This result was obtained using only the aggregation similarity (α=1,β=0). The same results (with longer time of retrieval) were obtained after adding the connection similarity (α=1/2,β=1/2,$\beta\alpha$=1/2,$\beta\beta$=1/2). This means that for simple processes like C3, the whole structure is often determined by the several components and topology of connections is steady, so there is no need for the connection similarity.

Table 1. The case-bases description.

Case-Base Name	Avg. no. of components	Avg. no. of connections	No of cases
Hydrogenation C3	8	8	13
Hydrogenation C6-C8	25	34	14

At it was expected the connection similarity positively influenced the retrieval accuracy for the complex cases. The result of experiments for hydrogenation C6-C8 are summarized in table 2. For the aggregation similarity the accuracy of retrieval was 62 % for the most similar case, and 54 % for the second nearest case. For the connection similarity REPRO achieved 69% accuracy for the most similar case, and increased it up to 77 % for the second nearest case. The superiority of the connection similarity has been confirmed by a qualitative evaluation. The graphical representation of the similarity between flowsheets has been found very useful by the expert.

Table 2. The result of experiments for hydrogenation C6-C8.

Retrieved Flowsheet	Aggregation Similarity $\alpha=1,\beta=0$ $\beta\alpha=0,\beta\beta=0$	Connection Similarity $\alpha=0,\beta=1,$ $\beta\alpha=2/5,\beta\beta=3/5$	Global Similarity $\alpha=1/2,\beta=1/2,$ $\beta\alpha=2/5,\beta\beta=3/5$
first nearest	62 %	69 %	69 %
second nearest	54 %	77 %	69 %

It should be emphasized that table 2 was created based on a binary decision: yes (good retrieval outcome) or no. Several times, when REPRO retrieved an improper flowsheet, very interesting local similarities were discovered that were previously not known by the expert. The performance of the system was very good when an input case was an incomplete flowsheet (part of the whole flowsheet), that is indeed the most common real life situation. Finally, REPRO almost always computed a very low global similarity for the flowsheets that were evaluated by the expert as a definitively not an acceptable solution.

6 Conclusions and Future Directions

The adaptation is well-known as a crucial task for the Case-Based Design Systems. At present this function was not implemented because of two reasons. First, this approach concerns supporting conceptual design and adaptation concerns mostly the final stages of a design. Second, in a flowsheet there are a lot of inter-related components. This means that a change in one of them can not be made without appropriate modifications to the related parts of a flowsheet. We have found very difficult the problem of representing those hidden inter-relations. Unfortunately this kind of knowledge is necessary during the adaptation process for a creation and/or deletion given components and connections.

The importance of the knowledge acquisition in this approach should be clearly underlined. The lack of exponential complexity and the acceptable retrieval outputs have a background in a good domain recognition and careful flowsheets preparation. Disadvantages of this approach are clear, especially during preparation of an input case, where the user must be conscious of a ”vocabulary” that was used for describing previous flowsheets. Nevertheless the system at the present level of development was accepted by the experts as a potentially useful tool for supporting the design activity. The integration these CBR functionality with the commercial CAD system is now under consideration.

Acknowledgments

We would like to thank Yves Charlez, Alan Charon, and Jean-Paul Dessapt from the Industrial Division at I.F.P. for providing the domain knowledge and helpful comments.

References

Bisson G. (1995). Why and How to Define a Similarity Measure for Object Based Representation Systems. In *Towards Very Large Knowledge Bases*, pp.236-246, IOS Press, Amsterdam.

Douglas J. (1988). *Conceptual Design of Chemical Processes.* McGraw-Hill.

Gentner (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, vol.7, pp. 155-170.

Falkenhainer B., Forbus K., Gentner D. (1989). The Structure-Mapping Engine: Algorithms and Examples. *Artificial Intelligence*, vol.41, no.1, pp.1-64.

Fraga E. (1994) The Implementation of a Portable Object-Oriented Distributed Process Engineering Environment. *Technical Report 1994-17*. Department of Chemical Engineering, Edinburgh University.

Holyoak K., Thagard P. (1989). Analogical Mapping by Constraint Satisfaction. *Cognitive Science,* vol.13, pp.293-355.

Maher M. et al. (1995). *Case-Based Reasoning in Design*. Lawrence Erlbaum.

Maher P. (1993). A Similarity Measure for Conceptual Graphs. *International Journal of Intelligent Systems*, vol.8, pp.819-837.

Motard R. (1989). Integrated Computer-Aided Process Engineering. *Computers & Chemical Engineering* 13(11-12), 1199-1206.

Myaeng S., Lopez-Lopez A. (1992). Conceptual graph matching: a flexible algorithm and experiments. *J.Expt, Theor. Artif. Intell.*, vol.4, pp.107-126.

Pu P. (1993). Introduction: Issues in Case-Based Design Systems. *AI EDAM*, 7(2), 79-85.

Rumbaugh J. et al. (1991). *Object-Oriented Modeling and Design*. Prentice-Hall.

Stephanopoulos G. (1987). Design-Kit: An Object-Oriented Environment For Process Engineering. *Computers & Chemical Engineering* 11(6), 655-674.

Surma J., Braunschweig B (1995). Reutilisation d'Etudes de Procedes. *Proceed. of the XVII Conf. Int. des Industries de Procedes: INTERCHIME'95*, Paris.

Surma J., Braunschweig B. (1996). Case-Based Retrieval in Process Engineering: Supporting Design by Reusing Flowsheets. *Enginnering Applications of Artificial Intelligence, Special Issue: AI in Design Applications* 9(4).

Surma J. (1996). *REPRO ver.1.3. User Manual and Implementation*. IFP Rapport - Juillet 1996.

Voss A. et al. (1994a). *Similarity concepts and retrieval methods.* FABEL Report No.13, Gesellschaft fur Mathematik und Datenverarbeitung mbH, Sankt Augustin.

Voss A. et al. (1994b). Retrieval of Similar Layouts- about a very hybrid approach in FABEL. *Artificial Intelligence in Design`94,* Gero J. and Sudweeks F. (eds.), Kluwer Academic Publ. 625-640.