PERGAMON

Engineering Applications of Artificial Intelligence 11 (1998) 449–459

Engineering Applications of
ARTIFICIAL
INTELLIGENCE

Contributed Paper

Interaction between physical and design knowledge in design from physical principles

N.M. Sgouros *

Department of Informatics, University of Piraeus, Karaoli & Dimitriou 80, Piraeus, 185 34, Greece

Received 1 September 1997; accepted 1 April 1998

Abstract

This article describes OUZO, a design system for chemical separation that supports engineering design from physical principles. The main idea behind OUZO is the orchestration of the interaction between physical and design knowledge. The program supports and controls the application of a set of representations and reasoning methods, thereby allowing different types of physical and design knowledge to interact effectively with each other. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords: Knowledge representation; Expert systems; Knowledge bases; Qualitative reasoning; Design from physical principles

1. Introduction

Design systems based on physical principles can be loosely defined as programs that concentrate on integrating representations taken from mathematics, physics and engineering in the design process. This article describes OUZO, a program that supports engineering design from physical principles. The main feature of this system is its ability to control the application of a set of representations and reasoning methods, thereby allowing different types of physical knowledge (i.e. qualitative reasoning domain theories and numerical models) to interact effectively with design knowledge (i.e. heuristics and design strategies). OUZO focuses on the early part of the engineering design process, commonly referred to as conceptual design. The system has been used in the conceptual design of separation systems, an important design problem in chemical engineering.

Although there has already been significant research in design systems based on physical principles, most of this work has concentrated on modeling either the design or the physical knowledge used in this process (Joskowicz and Williams, 1992). This paper complements previous research, by describing a design system that focuses on the interaction between the physical and the design knowledge. The implemented system provides flexible patterns of interaction between the design knowledge, which is predominantly heuristic, and the physical knowledge which describes design artifacts.

The next two subsections present an overview of the philosophy behind the design of OUZO. Section 2 describes the representations and the reasoning methods used for the various types of knowledge in OUZO. Section 3 explains how OUZO controls the application of these different types of knowledge in design. Section 4 describes an example of the system solving a typical separation design problem. Finally, Section 5 provides conclusions and an indication of future work.

1.1. The main ideas

OUZO supports a common distinction, drawn in informal descriptions of design, between experience-learned heuristics and science-based models. In this approach, physical knowledge consists of science-based models and represents the understanding that engineers have of physical systems. During the analysis phase, it generates descriptions of the behavior for possible designs. In addition, it computes design alternatives during the synthesis phase. Design knowl-

---

* Corresponding author. Tel: 0030 1 41 20 751; Fax: 0030 1 41 12 463; e-mail: sgouros@unipi.gr.

0952-1976/98/$ - see front matter
PII: S0952-1976(98)00037-2

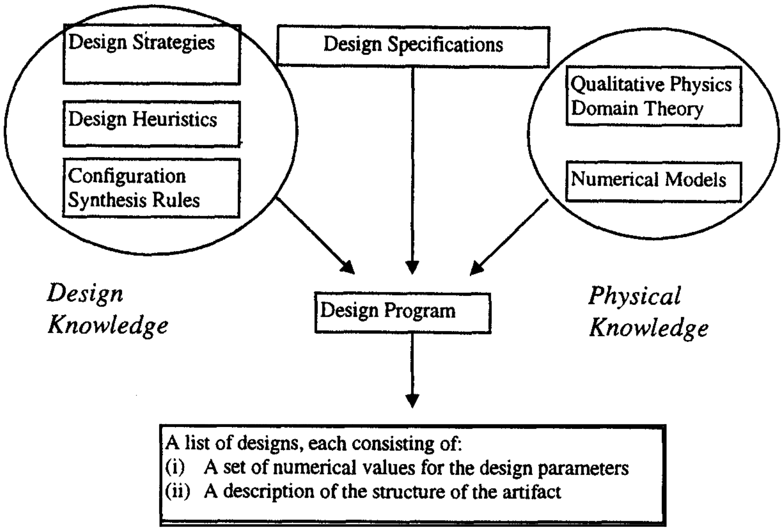

Fig. 1. OUZO architecture.

edge refers to heuristic knowledge that describes how design is performed. It is used during the synthesis phase to prune the number of design choices. Furthermore, it suggests physical conditions of particular parameter values that increase the efficiency of the analysis phase.

OUZO uses qualitative and numerical models for capturing the physical knowledge in design (Fig. 1). Heuristics, strategies and configuration synthesis rules represent the design knowledge in a domain. OUZO combines the physical and design knowledge with the design specifications, to come up with a list of designs, each consisting of a description of the structure of the artifact and a set of numerical values for the design parameters. The system includes a set of reasoning methods that deal with the qualitative and numerical representations in the physical knowledge component, along with an interpreter that transforms the design knowledge representations into appropriate rules, and implements the actions suggested by them, using a set of primitives.

OUZO models the design process as a sequence of cycles that generate, evaluate and implement design alternatives. It contains a novel controller algorithm that orchestrates the interaction of the different types of knowledge in the design cycle, and is independent of specific design methodologies (i.e. evolutionary or heuristic methods). This framework satisfies the need to integrate qualitative and numerical models with heuristic representations that has repeatedly been stressed in the engineering literature (Forbus, 1990).

1.2. Design algorithm.

OUZO proceeds through the generation, evaluation and implementation of design alternatives, i.e. mutually exclusive decisions that trigger changes to the structure of the artifact. For example, some of the design alternatives for multi-component mixtures in OUZO consist of mutually exclusive choices as to the separation method (e.g. distillation) for the current column. Decisions on these parameters determine the structure (e.g. diameter, height, number of stages, etc.) for the current separation unit[1]. The structure of the artifact during each design cycle is captured by the design description.

Design alternatives are processed in a sequence of design cycles (Fig. 2). In the beginning, a design cycle accepts as input the design specifications, and analyzes the current design description using the physical knowledge. The purpose of this analysis is to compute relevant features of the design description (such as the behavior it entails) and/or to elaborate on the design specifications and determine sets of parameters that are important for the current design problem.

In the rest of the cycle, OUZO generates and evaluates design alternatives using the physical and the design knowledge, and proposes and implements changes to the current description, using the design knowledge. At the end of the cycle, if the design description has changed, a new cycle starts with the current description and the design specifications as inputs; otherwise, the design cycle terminates. In the latter case, if the description does not satisfy the design specifications, the process exits with failure; otherwise, it terminates successfully.

A design method optimizes the generation and evaluation of design alternatives. For example, the

---

[1] An example of a decision that is not considered a design alternative in OUZO is the choice of notation for describing components in design diagrams. This decision does not have an impact on the structure of the artifact.

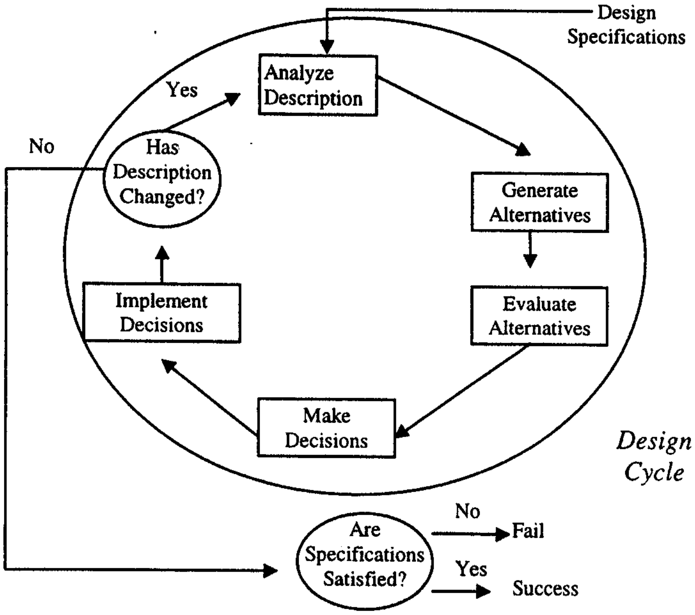

Fig. 2. The design algorithm in OUZO.

case-based design method optimizes the generation and evaluation of alternatives through the reuse and adaptation of previous cases. Evolutionary design is another design method, similar to the case-based methodology, that optimizes the generation and evaluation of alternatives using a standard set of heuristics for creating an initial design. This initial design is then modified by a set of evolutionary rules that challenge the decisions made by the initial set of heuristics.

A design strategy is a domain-specific instance of a design method. For example, one of the strategies in OUZO, the Nath and Motard strategy (Nath and Motard, 1981), is an instance of an evolutionary design method. As Section 2 explains in more detail, a design strategy represents an optimal application of a design method on a specific type of problem.

2. Physical and design knowledge

2.1. Representation

OUZO uses qualitative physics and numerical representations for capturing physical knowledge in design. Qualitative representations provide an ontological framework for describing physical phenomena, and represent the causal dependencies between its parameters and the modeling assumptions used in the description of a physical system. Numerical representations consist of systems of numerical relations (equations or inequalities) between the parameters of a system. This approach combines the rich modeling language of qualitative formalisms with the accuracy offered by numerical models. OUZO supports qualitative physics representations based on qualitative process theory (Forbus, 1990).

2.2. Reasoning methods

OUZO uses physical knowledge to generate design alternatives, and to determine their behavior during the design cycle. It does this using a set of physical principles to generate a set of alternatives and construct models for their behavior, based on numerical values for the system variables, on descriptions of the structure of possible designs, on sets of modeling assumptions, and on the results of the evaluation process performed by the design heuristics. Qualitative representations support this process with compositional modeling techniques (Falkenhainer and Forbus, 1989) that are sensitive to changes in all of these parameters. Three reasoning methods support the generation and analysis of design alternatives in the physical knowledge component: qualitative analysis, numerical model construction and numerical equation solving.

2.2.1. Qualitative analysis

Qualitative analysis generates qualitative models for the design description and the alternatives during each design cycle. It accepts as inputs the design description, the qualitative domain theory, and a set of modeling assumptions, and computes the minimal sets of conditions under which a model fragment (Falkenhainer and Forbus, 1989) is active (Fig. 3). The results of this step activate qualitative model fragments

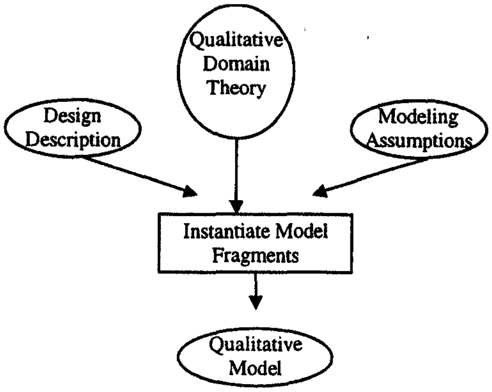

Fig. 3. Qualitative analysis flowchart.

that are consistent with the input parameters. This analysis corresponds to the first step of the qualitative analysis in SIMGEN (Forbus and Falkenhainer, 1992). OUZO demonstrates that this type of qualitative analysis is general enough to support typical conceptual design tasks such as the design of separation systems.

2.2.2. Numerical model construction

Numerical model construction (Fig. 4) creates numerical models for the design description and the alternatives during each design cycle. During this process, the numerical relations in the physical knowledge are combined with the results of the qualitative analysis and the decisions made by the design heuristics. This method ensures that these numerical models are consistent with all the analysis parameters and the decisions taken by the heuristics. Furthermore, it allows the design knowledge to decide on the type of analysis for each alternative.

Typically, in conceptual design, very approximate types of analysis are used at the beginning, to screen the alternatives. As the design evolves, the heuristics pick a set of promising candidates, which are combined with the results of the qualitative analysis to activate more detailed numerical models during the design cycle. OUZO accomplishes this task by using the results of the qualitative analysis and the current focus environment. The latter is an ATMS (assumption-based truth maintenance system) focus environment (Forbus and de Kleer, 1988), which consists of the major design decisions taken by a set of decision primitives described below (see Table 2). These decisions consist of the separation schemes for each column, along with predicates that denote which columns are being examined by OUZO at the current stage. For example, whenever the heuristic analysis decides on a particular separation for a column, the current focus environment is updated to reflect this decision. As a result, more-detailed numerical models that are now implied by the new focus environment are activated, resulting in a deeper level of analysis for the chosen separation.

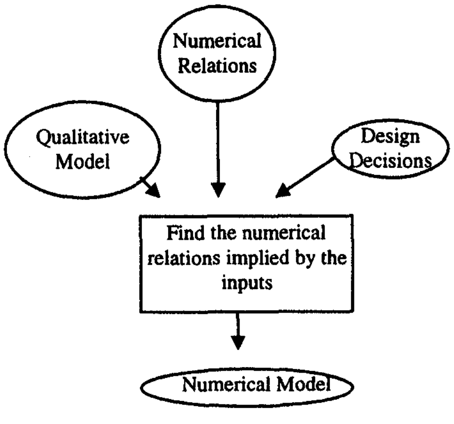

Fig. 4. Numerical model construction flowsheet.

Table 2. Primitives for design actions

Heuristic type	Primitive
Rejection	Reject an alternative for the current design cycle
Decision	Assert that an alternative holds in the rest of design Assume that an alternative holds in the rest of design Assume that an alternative holds for the current design cycle Indicate that all the design specifications have been satisfied
Analysis	Assign a value to a parameter Check whether an item is part of the current design description
Ordering	Establish an order of preference between two alternatives Ask the user to select between two alternatives
Evolutionary	Do not consider an alternative in the rest of design Check to see if an alternative has already been examined Store the current design description Reinstate the most recent design description

2.2.3. Numerical equation solving

Numerical equation solving applies algebraic techniques and numerical analysis procedures to solve numerical models. It accepts as inputs the results of the numerical model construction process, a set of numerical values for some of the design parameters, a set of rules for algebraic equation solving, and numerical analysis procedures, and tries to find numerical solutions for as many design parameters as possible (Fig. 5). OUZO performs this task by indexing the set of equations according to the quantities they involve, and then replacing these quantities by their numerical values in every equation in which they occur, as soon as the values have been computed. Equation solving

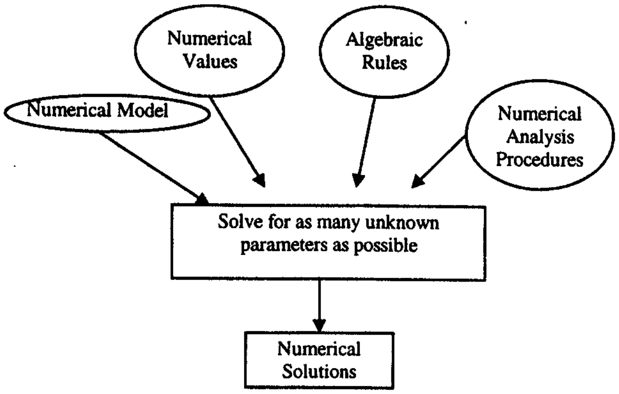

Fig. 5. Numerical equation solving flowchart.

continues as long as there are equations with at least one unknown quantity in them.

2.3. Design knowledge

Three types of representation are used to express the design knowledge in OUZO:

1. heuristics;
2. strategies—plans for optimizing the application of the heuristic rules;
3. configuration synthesis rules—rules for monitoring the design and producing the actual descriptions of the artifacts.

A design interpreter transforms these representations into appropriate rules, and implements the actions suggested by the heuristics via a set of primitives. The rest of this section describes these representations, and the interpreter commands, in more detail.

2.4. Heuristics

Heuristics are grounded in the physical knowledge about the domain. OUZO supports this grounding by providing qualitative and numerical models, in which the terms referenced by the heuristics are described. As a result, the representation and use of heuristic knowledge is significantly facilitated.

For example, one of the heuristics for designing separation systems suggests that the least tight separation should be preferable to any other alternative for the current separation unit. One of the actual heuristic rules for representing this heuristic in OUZO translates the least tight criterion into differences between the relative volatilities of the design alternatives, i.e. their relative tendencies to evaporate. Because these volatilities are defined in the physical knowledge component, the user can ask OUZO to display the model fragments in which these quantities are defined, and to describe the assumptions (e.g. reference conditions) under which these definitions hold. Furthermore, OUZO can display to the user the numerical models for computing these quantities under the current modeling assumptions.

Table 1 gives an example of some of the design heuristics in OUZO. It contains twelve filtering heuristics developed for this work, that describe the major heuristics in separation system design. These rules are based on the idea that separation systems are instances of filtering devices, since they isolate specific components from their input.

2.5. Representing heuristics

There are three kinds of knowledge involved in the representation of the conditions of heuristic rules: modeling and design assumptions about the problem, structural features of the design, and numerical values for parameters in the design. All these are grounded in the physical knowledge component. In addition to these kinds of knowledge, languages for heuristics (and design knowledge in general) must contain primitives that implement the actions suggested by these rules. In OUZO, these primitives are commands to the interpreter, describing how to update the set of design alternatives. The program uses thirteen primitives for capturing design actions, based on the following classification of heuristic rules:

1. Rejection rules prune the number of design alternatives by eliminating solutions that do not meet certain criteria. Examples include heuristics 2, 3, 5 and 10 in Table 1.
2. Ordering rules establish preferences between various design choices. Examples include heuristics 1, 6, 7, 8, 9, and 11 in Table 1.

Table 1. Filtering heuristics for design

#	Heuristic rule
1	Prefer to preserve the purity of the product during filtering
2	Avoid operations that allow interference between the components during filtering
3	Avoid damaging the filters
4	Preserve the original specifications for the filtering process
5	Avoid extreme operating conditions in the filter
6	Prefer operations that use fewer filters
7	Prefer the filtering process with the smallest energy requirements
8	Prefer filtering operations based on properties of the input for which there is the maximum variance between the components
9	Prefer to perform difficult filtering operations with the minimal amount of input
10	Avoid filtering operations that introduce extra agents in the input which cannot be easily removed
11	Prefer to perform difficult filtering operations with the minimal number of components in the input
12	Use the least expensive filter

3. Decision rules select a design alternative. Heuristic 12 in Table 1 provides an example.
4. Analytical rules propose numerical values for some of the parameters of the system, in order to facilitate the analysis of proposed designs. Heuristic 4 in Table 1 provides an example.
5. Evolutionary rules challenge the design decisions made by other heuristics. Typically, these rules are specific to a design strategy. For example a rule in OUZO challenges rule 4 in Table 1 by allowing for separation with different recoveries from the ones in the design specifications.

Table 2 contains the actual primitives that capture the design actions in OUZO.

2.6. Strategies

Some of the actions proposed by the design heuristics may be in conflict with each other. For example, in Table 1 heuristics 4 and 12 can be contradictory in cases where separations that preserve the original specifications do not turn out to be the cheapest ones. Therefore, it is necessary to create consistent subsets of heuristics, or to sequence their application in ways that resolve possible conflicts during design. For example, in separation system design, one can create sets of heuristics in which rules 4 and 12 do not coexist, or one can sequence their application so that rule 4 is always applied before rule 12. The latter strategy means always picking the least expensive among all the separations that preserve the original specifications. Design strategies provide ways of organizing the application of heuristic knowledge along these lines.

More specifically, the design strategies in OUZO are plans for sequencing the execution of heuristic rules in ways that were found to be capable of producing optimal designs, Table 3 contains part of the description of a strategy for the design of separation systems (Nath and Motard, 1981) that is represented in OUZO.

2.7. Configuration synthesis rules

These rules capture the knowledge associated with producing design descriptions (i.e. flowsheets), and monitoring the state of the design process. This organization allows the builder of the design knowledge base to separate more general forms of knowledge, like heuristics, that can apply to more than one class of systems, from more-specific forms of knowledge, like configuration synthesis rules, that deal with specific classes of devices. Table 3 contains an example of the English interpretation of such a rule in OUZO.

3. Controlling design

OUZO controls the design process by means of an algorithm that orchestrates the interactions between the various types of knowledge being used. It consists of three steps (Fig. 6):

3.1. Qualitatively analyze the current design description

This step performs a subset of the qualitative analysis used in SIMGEN (Forbus and Falkenhainer, 1992). It uses the current design description to instantiate a set of qualitative model fragments that are consistent with it. Furthermore, it determines the set of conditions under which each model fragment becomes active. Because qualitative analysis is used in determining the behavior of the design description and in generating design alternatives, this step is part of the analyze-description and generate-alternatives steps of the design cycle in Fig. 2.

3.2. Construct and solve the numerical models

The numerical model construction and equation-solving methods are applied at this point. Since numerical models are used in OUZO to describe the behavior of the artifact and the design alternatives created by the qualitative models more accurately, this step corresponds to the analyze-description and generate-alternatives steps of the design cycle as well.

3.3. Apply the design strategies and the configuration synthesis rules

During this step, the heuristics choose design alternatives, and the current design description is updated accordingly. In particular, strategies and heuristics evaluate alternatives and make decisions, while configuration synthesis rules implement these decisions and monitor the state of design. This step corresponds to the evaluate-alternatives, make-decisions and implement-decisions steps in Fig. 2.

Steps 2 and 3 are executed in an inner loop that ends when there are no more design decisions to be taken. In this case, if the design description has been modified, the system goes back to step 1. The design process ends when the design description remains unchanged during a cycle. In this case, if the design specifications have been satisfied, the algorithm terminates with success; otherwise, it exits with failure.

The loop between steps 2 and 3 does not correspond to a similar cycle in Fig. 2. Its purpose is to make OUZO more efficient. In particular, because qualitative analysis is computationally the most expensive stage in the design cycle, OUZO tries to do as much of the analysis as possible using the numerical models and

Table 3. An example of a design strategy description, along with the interpretation of a configuration synthesis rule in OUZO

Original Design Strategy Description

"Evolutionary rule 1 questions the validity of the heuristic rule, and is applied before any other evolutionary rule to resolve the question of the product set definition. Evolutionary rules 2, 3, 4 and 5 are treated equally, but of course cannot be applied at the same time. Therefore, starting from the feed stream forward, evolutionary rule 2 is applied next. If any modification is suggested by this rule, it is adapted in the starting structure, and a new structure is produced. The new structure, or the starting one, whichever is superior, is evolved further by applying rule 2 to the portion of the structure not checked by rule 2 in the earlier application. Evolutionary rule 3 is applied, starting from the feed stream forward, after no further structural modifications are suggested by rule 2."

Strategy Implementation

```
(defStrategy Nath-&-Motard-Evolutionary-Strategy ;; The name of the strategy
;; The classes of heuristics used by this strategy
:Heuristic-Classes (Evol-Rule-1 Evol-Rule-2 Evol-Rule-3 Evol-Rule-4 Evol-Rule-5)
;; The conditions under which the strategy is applicable
:Conditions ((Separation-System ?system)
    (Consider (Evolutionary-Strategy-for ?system Nath-&-Motard))
    (Consider (Design-Complete ?system ?cost)))
;; The problem-solving context under which the heuristic rules are interpreted
:ATMS-Context :Implied-By
;; The predicates in this slot direct the application of the strategy to specific points in the design description
:Focus-Predicates ((Apply-Strategy-to ?system))
;; The body of the rule in which the defStrategy form is translated consists of the contents of the
;; Action slot along with a set of functions in which the Execution-Order slot is translated.
:Action ((cond ((< ?cost (find-previous-cost)) (store-design) (assume-in-cycle (Apply-Strategy-to ?system)))
    (t (pop-design))))
:Execution-Order (:SERIAL Evol-Rule-1 Evol-Rule-2 Evol-Rule-3 Evo-Rule-4 Evol-Rule-5))
```

Strategy Code Interpretation

IF the design of a separation system has been completed
AND the Nath-&-Motard strategy is used to evolve the current design
THEN if the current separation system costs less than the previous design
  mark the separation system that is going to be evolved (the :Action part)
  and then apply the heuristics stored under the class Evol-Rule-1, followed by the heuristics stored under the class Evol-Rule-2, followed by the rest of the heuristic classes in the :Execution-Order slot of the form. If the separation system costs more, then pop the previous design description and apply the evolutionary heuristics in the order described above.

Interpretation Example for a Configuration Synthesis Rule

IF the steady-state design features for the column are active
  AND the column has a partial condenser and a partial reboiler
  AND the particular stages for these units have been defined
  AND the value for the number of stages in the column is N
  AND the current design description does not correspond to a column with N stages
THEN create a new design description for a column with a partial condenser, a partial reboiler and N stages.

the heuristics, before resorting to qualitative analysis during the analyze-description step in Fig. 2.

This controller has been applied on two types of problems in OUZO. The first one deals with a heuristic method for binary distillation design (Sgouros, 1992), while the second one supports the evolutionary design of separation systems for multi-component mixtures (Sgouros, 1993).

4. An example

This section describes a separation-system design problem, to illustrate how OUZO works. A separation system is a sequence of process units (e.g. distillation columns) that achieve a desired separation. Separation-system design is a good application domain for two main reasons. First, it involves significant

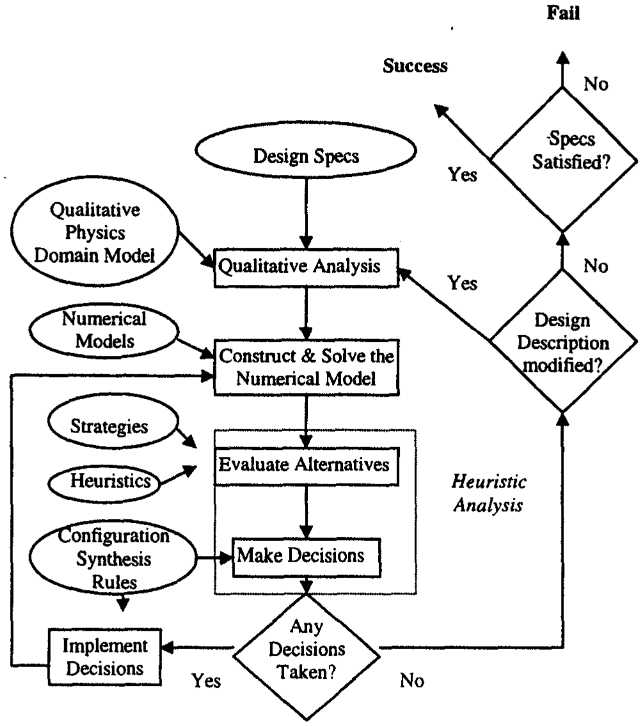

Fig. 6. The controller algorithm in OUZO.

physical and design knowledge. Representing these types of knowledge and modeling their interaction provides a challenge for any computer implementation. In addition, separation systems can be found in almost every chemical plant; therefore finding ways of automating their design can make chemical process design more efficient in general.

OUZO supports two of the most widely used separation processes in chemical engineering: distillation and extractive distillation. Distillation involves the separation of the components of a mixture on the basis of differences in their tendencies to evaporate at a given temperature. The process results in two products: the distillate, which is richer in components with low boiling points, and the bottoms, which contain mainly substances with high boiling points. A common analysis method for describing distillation columns that accept, as input, multicomponent mixtures starts by ordering the components of the mixture according to their boiling points.

Table 4. Physical properties for multi-component distillation input example

Species	Normal boiling point °C
Propane	−42.1
Isobutane	−11.7 ← light key
n-Butane	−0.5 ← heavy key
Isopentane	27.8
n-Pentane	36.1

In this case, two of the components of the mixture with neighboring boiling points are selected as the keys for the separation. Usually, the one with the lower boiling point is called the light key, while the other is called the heavy key. All the other components are called nonkeys. Under this description, distillation causes most of the light key and all the more volatile nonkeys to appear in the distillate, while most of the heavy keys and all the less volatile nonkeys end up in the bottoms. For example, if a distillation column accepts as input the mixture shown in Table 4 [extracted from Seader and Westerberg (1977)] and isobutane and *n*-butane are specified as the light and heavy keys, respectively, then the result will be a distillate containing most of the isobutane, all of the propane and some of the *n*-butane. The bottoms will consist of most of the *n*-butane, some of the isobutane, and all of the isopentane and *n*-pentane.

Extractive distillation is different from the ordinary case, in that it involves the addition of a new component to a mixture to facilitate the separation of the system by distillation (King, 1971). The added component changes some of the properties of the mixture in a direction that favors the desired separation.

In general, the problem of designing separation systems is defined as follows (Nath and Motard, 1981):

> "Given a feed stream of known conditions (i.e. composition, flow rate, temperature, pressure), synthesize a process that can isolate the desired (specified) products from the feed at minimum cost."

Table 5 describes the specifications [taken from Nath and Motard (1981)] for such a problem, which was run in OUZO, along with the final output of the program in this case. The input specifications require the design of a separation system for recovering the components of a 6-component mixture in four specific sets of products. In addition to this description, OUZO is given the modeling assumptions to use during the analysis phase, along with a set of design assumptions that include the name of a design strategy to use, and suggested values for some of the parameters. This example uses an evolutionary design strategy.

The final design in this case consists of five columns (Fig. 7). There are 1344 possible designs for this particular problem. The purpose of the design strategies coded in OUZO is to reduce this number to a manageable number of alternatives. The results of this program are consistent with the ones presented in the chemical engineering research literature (Nath and Motard, 1981).

Table 5. Problem specification for the *n*-butylene purification problem

Feed components	Mole fraction	Product components	Conditions
Propane	0.0147	Propane	Temperature = 53.89°C
n-Butane	0.5029	*n*-butane	Pressure = 5.62 kg/cm^2
Butene-1	0.1475	Butene-1/*trans*-butene-2/Cis-Butene-2	Total Flow rate = 303.04 kg mol/h
Trans-butene-2	0.1563	*n*-Pentane	
Cis-butene-2	0.1196		
n-pentane	0.0590		

The design cycle for this example (cycle 1 in Fig. 8) begins with an analysis of the design description suggested by the specifications during the analyze-description step. OUZO contains a qualitative domain theory that describes the separation properties of multicomponent mixtures, along with the structure and design features of process units, such as ordinary and extractive distillation columns. In addition, the program contains numerical models that describe these qualitative models more accurately. The analyze-description step uses both models and results in the introduction of relevant physical properties for the multicomponent mixture and its substances (e.g. boiling points, vapor pressures, etc.) which are instantiated and solved. The generate-alternatives and evaluate-alternatives steps that follow have no effect on the current design cycle, since the design description does not contain any specific process units at this point. During the make-decisions step, OUZO decides to create the first process unit (i.e. column) in the separation system. The implement-decisions step that comes next uses the configuration synthesis rules to modify the design description, and to create the description for the first column in the separation system.

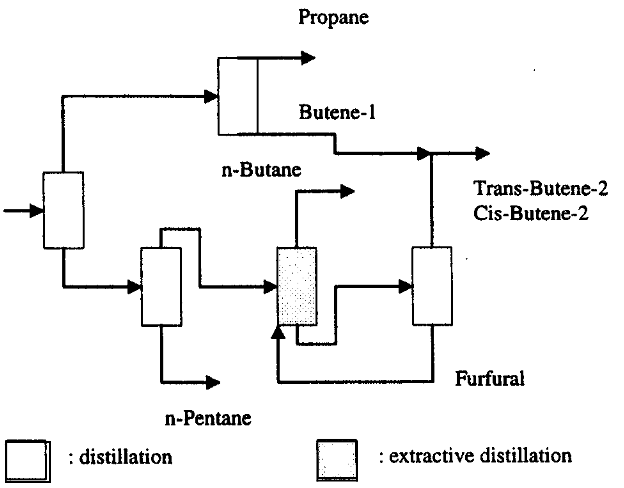

Fig. 7. Final design suggested by OUZO for the *n*-butylene purification problem.

A new design cycle (cycle 2 in Fig. 8) starts with the modified design description as its input. The analysis step instantiates a set of relevant design features for the current column, such as its dimensions and operating conditions. The generate-alternatives step uses the modeling assumptions in the design specifications and the qualitative domain theory to create a set of alternatives that cover both the type of separation process and the choice of keys for the current process units.

The next step evaluates all the design alternatives, using the design knowledge component and the results of the previous step. In particular, the design assumptions in the specification instantiate a design strategy which, in turn, activates the set of heuristic rules it contains, to establish a preference order between the alternatives. The heuristics use this ordering to decide on the kind of separation that will take place in the current column during the make-decisions step of the cycle.

During the rest of the make-decisions step, the configuration synthesis rules check whether the current design recovers all of the desired products specified in Table 5. The implement-decisions step that follows instantiates more-detailed numerical models, that calculate cost estimates for the chosen separation. In addition, it applies the configuration synthesis rules to change the design description to be consistent with the decisions taken during the previous step.

This design cycle is repeated until there are no more changes to the design description, resulting in the flowsheet in Fig. 7. At this point, OUZO checks whether a design for the current problem has been found, and exits with success or failure accordingly. OUZO does not specify a particular backtracking method in the case of failure. Instead, the method used is specified as part of each design strategy.

A toluene–benzene binary distillation problem (King, 1971) was used to test the physical and design knowledge components that deal with binary separations in OUZO (Sgouros, 1992). In addition, two well-known separation cases from the chemical engineering literature (the C_6 separation (Nath and Motard,

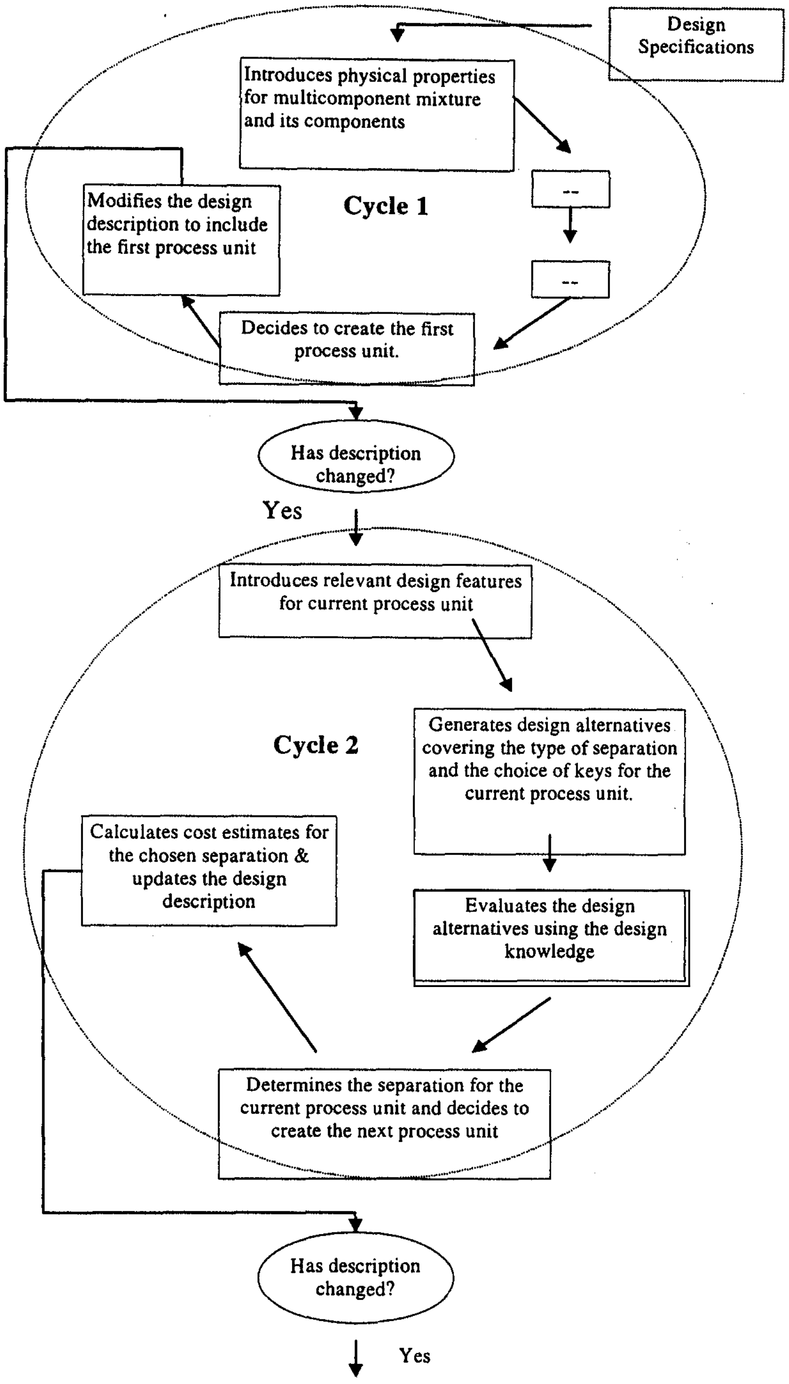

Fig. 8. Example of two consecutive design cycles in OUZO. The steps in each cycle correspond to those in Fig. 2.

1981) and the *n*-butylene purification (Seader and Westerberg, 1977) problems) were used to test the physical and design knowledge components for the multi-component mixtures (Sgouros, 1993). The results were consistent with those presented in the chemical engineering literature (Nath and Motard, 1981; Seader and Westerberg, 1977). OUZO is coded in LISP, and runs on an IBM RS/6000 workstation.

5. Conclusions and future work

This article describes OUZO, a chemical separation design system that provides effective patterns of interaction between the physical and design knowledge in design from physical principles.

Future work includes the development of more sophisticated explanation capabilities in OUZO, that justify the application of each design step on the physical and design knowledge used. Furthermore, work is in progress in integrating case-based and analogical problem-solving methods in this system.

References

Falkenhainer, B., Forbus, K.D., 1989. Compositional modeling: finding the right model for the job. Artif. Intell. 41, .

Forbus, K.D., 1990. Qualitative process theory. In: D.S. Weld, J. de Kleer (Ed.). Readings in Qualitative Reasoning About Physical Systems. Morgan Kaufmann, San Francisco, CA, USA.

Forbus, K.D., de Kleer, J., 1988. Focusing the ATMS. In: Proceedings of AAAI-88, St. Paul, MN, USA. AAAI Press, Menlo Park, CA, USA.

Forbus, K.D., Falkenhainer, B., 1992. Self-explanatory simulations: scaling up to large models. In: Proceedings of AAAI-92, San Jose, CA, USA. AAAI Press, Menlo Park, CA, USA.

Joskowicz L, Williams B (eds) 1992. Working Notes of the AAAI Fall Symposium on Design from Physical Principles. Cambridge, MA, USA, AAAI Press, Menlo Park, CA, USA.

King, C.J., 1971. Separation Processes. McGraw-Hill, New York.

Nath, R., Motard, R.L., 1981. Evoluionary synthesis of separation processes. AIChE J. 27 (4), .

Seader, J.D., Westerberg, A.W., 1977. A combined heurstic and evolutionary strategy for synthesis of simple separation sequences. AIChE J. 23 (6), .

Sgouros, N.M., 1992. Integrating qualitative and numerical models in binary distillation design. AAAI Fall Symposium on Design from Physical Principles. Cambrige, MA, USA. AAAI Press, Menlo Park, CA, USA.

Sgouros, N.M., 1993. Using qualitative, numerical and heuristic knowledge to support innovative design. The 7th International Workshop on Qualitative Reasoning about Physical Systems. Orcas Island, WA, USA, AAAI Press, Menlo Park, CA, USA.

Nikitas M. Sgouros received a Diploma in EECS from the National Technical University of Athens in 1988, a M.Sc. with distinction from the Department of Artificial Intelligence, University of Edinburgh in 1990 and a Ph.D. in Computer Science from Northwestern University in 1994. He is currently a lecturer in the Department of Informatics at the University of Piraeus, Greece. His research interests are in qualitative reasoning, engineering design, robotics and intelligent multimedia systems.