# A qualitative reasoning approach to chemical process design

**Ioa S. Gavrila[Ψ], Bert Bredeweg[#], Piet Iedema[Ψ]**

[Ψ] Department of Chemical Engineering
University of Amsterdam
Nieuwe Achtergracht 166
1018 WV Amsterdam
e-mail: ioa@chemeng.chem.uva.nl

[#] Department of Social Science Informatics
University of Amsterdam
Roetersstraat 15
1018 WB Amsterdam
e-mail: bert@swi.psy.uva.nl

**Abstract**

In this paper we illustrate the use of qualitative knowledge and reasoning to conceptual process design. For a certain class of chemical processes, we propose a design method in which qualitative and numerical decision making phases are alternated. Qualitative reasoning is performed when a rough scan of many design alternatives is desired; numerical decision making follows only for the promising looking qualitative alternatives. In all design phases Propose-Critique-Modify tasks are performed, to quickly evaluate design decisions and to prevent thus the entrance in unfeasible or sub-optimal parts of the design search space. A modular knowledge-based system has been implemented in which the modules follow the task decomposition of the design method.

## Introduction

Chemical process design is the activity in which ways are searched for creating new material wealth, by production of new materials or by upgrading the value of existing ones (Douglas, 1988). The complexity of the physical phenomena that must be understood and controlled for this purpose, as well as external constraints posed on the design process (e.g. time constraints) and the design itself (e.g. environmental constraints, costs aspects), make this design activity a very demanding one. The use of computers for its support seems obvious and even a prerequisite.

The current use of computers by designers may be a little surprising: they are almost exclusively used for numerical simulation, steady-state simulation being the most common variant but with dynamic simulation gaining in popularity. Aspen and Speed-Up are typical examples of commerical simulation packages used for this purpose. However, the tasks of determining the design structure and selecting appropriate models to describe unit behaviour, are left to the engineer.

Interest in other types of computer support is definitely present, as can be concluded from a number of publications in the last years. Design tasks for which proposals for automation have been undertaken include synthesis, model formulation and qualitative simulation (Dalle-Molle and Edgar, 1990; Catino et al., 1991; Sgouros, 1993). Nevertheless, the number of design systems in which synthesis, analysis and evaluation tasks are integrated, and in which switching between design alternatives is possible, is still very small (two examples are (Banares-Alcantara, 1994; Han et al., 1995)). This might be contributed to the same complexity that makes the use of design systems desirable.

The aim of the research reported in this paper is to build such a design system for conceptual process design. We focus on the design of processes that are characterized by the occurence of many interacting phenomena (e.g. with many side-reactions and long reaction chains to product), and propose a design method ( c.f. (Gavrila and Iedema, 1996)). A characteristic of this design method is its phenomena-orientedness, as opposed to the more traditional unit-oriented design methods. Further, we handle the uncertainty inherent to early design stages by the use of qualitative models and reasoning. This is in contrast to using less accurate numerical models, that give the illusion of correctness through their numerical precision.

The rest of the paper is organised as follows. Section 2 gives a short introduction to chemical process design topics, intended for readers unfamiliar with this domain. In section 3 the knowledge representation is discussed of some main entity types used by the design method. Next, the design method is presented in section 4, with emphasis on the qualitative design phases. In section 5 the method is illustrated by an example. A discussion and comparison with related work is found in section 6, followed by our conclusions in section 7.

## Chemical Process Design

The main goal of a chemical process is the production of new chemical components by reaction, or the purification of existing components by separation. The purity of the primary product in the product or output streams, as well as maximal allowable concentrations of by-products, are typical constraints found in the requirements specifications.

The design task amounts to deciding which operations to perform on the given raw material or feed streams to achieve optimal product streams at minimal costs (Nishida et al., 1981). A design alternative is completed at a conceptual level if it describes: 1. the structure of whole process: the types of unit operations and their interconnections; 2. the operating conditions within these units: their temperature, pressure; and 3. the streams, e.g. their phases, concentrations, flow rates.

An important body of knowledge needed during the design process, is information about reactions: their reactants, products, and rate dependency on temperature and component concentrations. For separations thermodynamic knowledge about phase equilibria is essential, e.g. what will the mole fraction of component X be in the vapour and liquid phase at a certain temperature and pressure ? In the rest of this section some background knowledge about reactions is given ((Levenspiel, 1972) is a good introduction to reaction engineering).

During a reaction certain chemical components, the *reactants*, are converted to other components, the *products*. The *stoichiometry* states the number of reactants and products involved, e.g. $\underline{2}A + \underline{3}B \rightarrow \underline{1}C$. Reactions may be reversible or irreversible; in the latter case the reaction cannot proceed in the opposite direction, e.g. $C \rightarrow 2A + 3B$ will not occur. During reaction heat is emitted or absorbed (exothermic and endothermic reactions respectively). This heat of reaction may be small, in which cases it can be neglected.

The rate of reaction is defined as the change in the number of moles N of one of its components, in a time interval dt and volume unit V, e.g. $r_A \equiv 1/V\ (dN_A/dt)$. For reactions that need only one phase to proceed (homogenous) this rate depends only on the concentration of reactants and the temperature. If the reaction rate is of the form $r_A = k(T)\ [A]^a\ [B]^b$ (as is often the case) then the powers a,b are called the *reaction orders* of reactants A and B respectively, and express the magnitude of the reactant concentration influence on the rate. If the reaction orders coincide with the stoichiometric coefficients of the reactants then the reaction is called *elementary*.

The temperature dependency is expressed by the *reaction rate constant* k (that is only constant at one temperature). For many reactions, especially for those where the rate can be expressed as above, Arrhenius' Law is a good approximation of the true temperature dependency (see below). The term $k_0$ is called the frequency factor and expresses how many collisions between reactant molecules are expected. The term E is the activation energy and R the universal gas constant; the exponential term in which they participate expresses the number of successful collisions, i.e. those in which the reactants react.

| | Arrhenius' Law |
|---|---|
| $r_C \equiv 1/V\ (dN_C/dt) = k(T)\ \Pi\ [R]^r$ | $k(T) = k_0\ e^{-E/RT}$ |

## Knowledge representation

This section describes the main entity types that the design system knows and reasons about. Figure 1 shows the relations by which they are related to each other. The entities CHEMICAL COMPONENT (or COMPONENT for short) and REACTION contain input data with which the physical world models (qualitative and numerical) inside the reasoning system are parameterised. The other entities shown in Figure 1 populate the 'Design Space': a Blackboard in which (partial) design alternatives are stored that have passed a certain amount of feasibility and optimality checking (see next section for more details).

Starting point of this modelling work were the results of (Stephanopoulos et al., 1990); their terminology has been used where appropriate for reasons of uniformity. Information about some entities is used in two resolution levels (Weld, 1992): qualitative and numerical. In the qualitative variant the continuos parameters can get only two values: > a threshold-value or <= that threshold-value. In the numerical variant the continuos parameters get numerical values. Besides, some entities are described by more properties.

***Reaction***. Only irreversible, homogenous liquid and gas-phase reactions are considered in the current version of the design method. The knowledge representation for the reaction entity type has therefore a property that states the phase. Further, a temperature and pressure range (that may not have a maximum) is used. For the rest a qualitative and numerical variant are distinguished. In the qualitative version only the names of the participating components are used, but not the stoichiometric coefficients. Further, the only rate information known is that the reaction rate is reasonably high for the given T,P ranges[1].

In the numerical variant the stoichiometry is known, together with the exact values of $k_0$ and E. Because only elementary, irreversible reactions are considered no data is needed about reaction orders and equilibria. The heat of reaction is assumed negligible, thus also this property is not needed.

[1] Note that this is a simplification of reality: if other reactions with an overlapping set of reactants also proceed then this reaction rate may become very small after all.

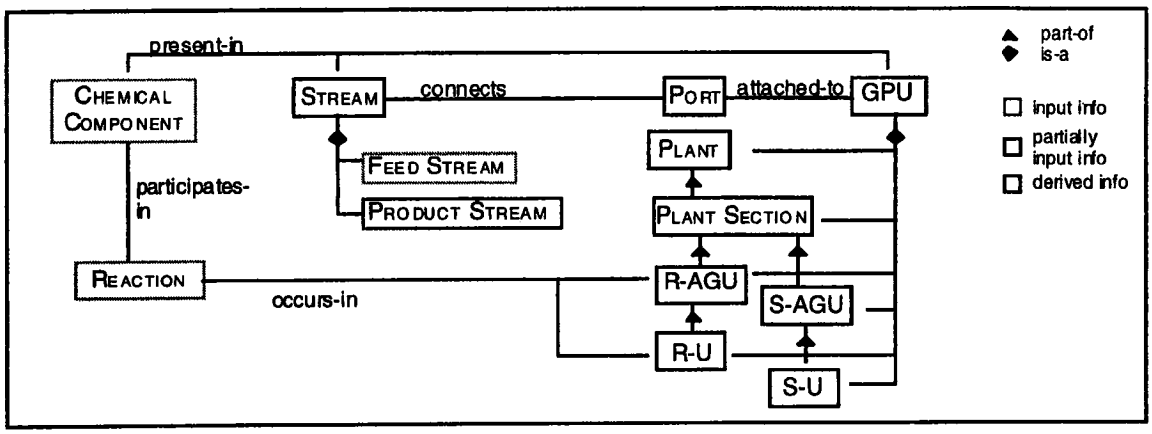


**Figure 1.** Relations between main entity types

***Component.*** The production of new chemical components or purification of existing ones is the main goal of a chemical process. In the current state of the design method only one physical property is needed: the boiling point as a function of pressure.

***Stream.*** Streams describe how the processing units of a process are interconnected and what the nature of the fluid is that flows between these units. The fluid is described by its phase and the present components. Only gas and liquid streams are considered here. Information about the components is available in a qualitative and numerical variant: in the former version only the names are used, in the latter also the mole fractions. The numerical variant has in addition also information about the volumetric flow rate ($m^3$/sec), temperature and pressure.

***Port.*** A port describes the assumptions about incoming and outgoing fluids that have been used while designing a processing unit. When the units are connected by streams then the stream description should match the port-description of the port to which they are attached. A port is described in the current design method version only by qualitative information: the expected phase and component names. For instance, if the port expects a liquid but the stream turns out (during simulation) to be a vapour stream, then the processing unit design might not be OK; the same holds for the arrival of unexpected components.

***GPU.*** A Generic Processing Unit is the highest abstraction of a processing system: it describes its input and output ports. Plants, plant sections, and single operating units are all examples of GPUs and inherit thus its properties.

***Plant.*** A plant entity describes the structure of the design of a whole process. It contains links to its subsections and the streams that enter and leave these subsections. Two types of subsections are possible: aggregated units (AGUs, see below) and groupings of AGUs, PLANT SECTIONS. The latter is a useful abstraction if different type of AGUs have the same functionality, e.g. production (in a R-AGU) and purification (in a S-AGU) of a certain needed reactant.

***R-AGU.*** A Reaction-Aggregated Unit is a process section that contains only reaction units: units whose main goal is the production of new components. When designing a R-AGU its functionality is determined: the reactions and gas-liquid phase changes[2] that should be stimulated in its units. Moreover, operating conditions ranges (T,P) are derived in which the rates of desired phenomena are reasonable and no not tolerable phenomena take place (e.g. the forming of toxics). A R-AGU design has a qualitative nature: only the names of the entering and leaving components are known (its PORT description is qualitative) and all the phenomena that can possibly take place (given the input components and operating condition).

***R-U.*** A Reaction Unit denotes here an ideal reactor. It is an abstraction of a real piece of reactor equipment and focuses on some important properties: the space-time, the amount of mixing of the fluid within the reactor, and the exact operating conditions. The *space-time* $\tau$ is a measure of the time that an entering fluid volume needs to achieve a certain conversion. The needed reactor volume V is directly calculated from the space-time and input volumetric flow-rate(s) $\upsilon$. The amount of mixing of reactor-fluid can greatly influence the product distribution and the needed space-time for a certain conversion. Here only the two extremes are considered: no mixing, in which case the R-U becomes a 'Plug Flow Reactor' (PFR), or complete mixing, making the R-U a 'Completely Stirred Tank Reactor' (CSTR). Given all this information, the reaction rates can be calculated at a certain $\tau$.

***S-AGU*** and ***S-U.*** A Separation-Aggregated Unit is a grouping of Separating Units whose main function is to separate components from each other and thus to purify streams. The design method does not focus on these parts of the chemical process yet; only S-AGUs are used and they are only described by their function: the names of the components to remove from the incoming streams.

---

[2] Phase changes are necessary if the next reaction leading to the product takes place in the other phase.

## Task structure

### Task decompositions and control

Starting from a design specification many design decisions must be made before arriving at a conceptual process design that is described with numerical precision. Some of these decisions are quite related to each other and should be taken together, others are less related. These considerations are reflected in the top-level task decomposition of the proposed design method. It consists of the following design phases (DP) and associated decisions:

*design R-AGU*: which needed phenomena should be stimulated in the same unit, what should its operating condition ranges be, what other phenomena will also occur under these conditions; DP 1

*connect R-AGUs*: to which R-AGUs should the feed streams be fed and how should the R-AGUs be connected by intermediate streams; DP 2

*design S-AGU*: which streams need purification and which components should be removed from them; DP 3

*design R-U*: for a certain R-AGU, make those decisions that make the calculation of output streams possible: decide on mixing pattern, exact temperature and pressure, space-time; DP 4

*design S-U*: for a certain S-AGU, decide which separation technology to use to implement (part-of) the S-AGU functionality, and, depending on the technology, make those decisions that fix the output streams. DP 5

The phenomena to be stimulated in a R-AGU are taken as the functional specification of R-U design. If it turns out that it is advantageous to alter the mixing pattern, then more R-Us are needed to implement the R-AGU (see for an example Figure 2). Similarly, the desired separations of a S-AGU are the starting point of S-U design.

The order in which the above-mentioned decisions are taken has a great influence on the effort needed to arrive at feasible and satisfying design alternatives. Unfortunately this order cannot be fixed completely for all cases. Process sections can be very dependent on each other; making a detailed design of one section and then finding out that the assumptions about its incoming streams are not correct because it turns out to be impossible to implement the functionality of a section up-stream, is a waste of effort.

The precise order of design phases should thus be determined by the design case, i.e. by information about where to expect problems. In general it will be an alternation of the qualitative design phases 1, 2 and 3 and the numerical phases 4 and 5. However, some constraints on the phase order can be made:

- before connecting R-AGUs, the R-AGUs should have been designed (obvious)
- before designing R-Us, their R-AGU should have been designed (but the R-AGU must not be connected yet);
- before designing S-Us, their S-AGU should have been designed.

To avoid getting into uninteresting parts of the design search space, decision making steps should be quickly alternated with evaluation steps. This idea is elaborated in the second-level task decomposition of the design method, that consists for all design phases of Propose-Critique-Modify task types (Chandrasekaran, 1990):

*generate:* propose different design alternatives by making different sets of decisions;

*select:* select a promising design proposed during the generate or modify task for closer examination;

*analyse*: derive the impact of the design decisions on the intermediate and output streams (simulate);

*evaluate*: search for undesired consequences of design decisions (e.g. production of toxics); compare derived stream properties with stream specifications;

*modify:* modify an analysed design, based on information from the evaluate step. Different modifications may be possible.

The order of task performance in the second-level task decomposition is easier to determine and depends on the success of task performance (e.g. OK, FAIL). For certain combinations of task type and termination status, the next task type is the same in all design phases, see DP 1 in Figure 2.

Only if the termination of an evaluate-task is OK, will a design be added to the Design Space. Figure 2 shows how the Design Space is changed by the insertion of new partial designs: DP1 adds isolated R-AGUs; DP2 adds designs in which already inserted R-AGUs are connected (if they have all needed phenomena occuring in them); DP3 inserts design alternatives that contain S-AGUs for designs produced by DP2; finally, DP4 inserts for R-AGUs numerical design alternatives: one or more R-Us.

### Description of design tasks

In the rest of this section the subtasks of the first and second design phase, i.e. 'design R-AGU' and 'connect R-AGUs', will be discussed in more detail. Their functionality will be partially specified with if-then rules, using a intuitive notation. For a discussion of design phase 4 the interested reader is referred to (Gavrila and Iedema, 1996).

Before the actual design process begins, however, an analysis of the current design case is performed in the task 'Role Assignment'.

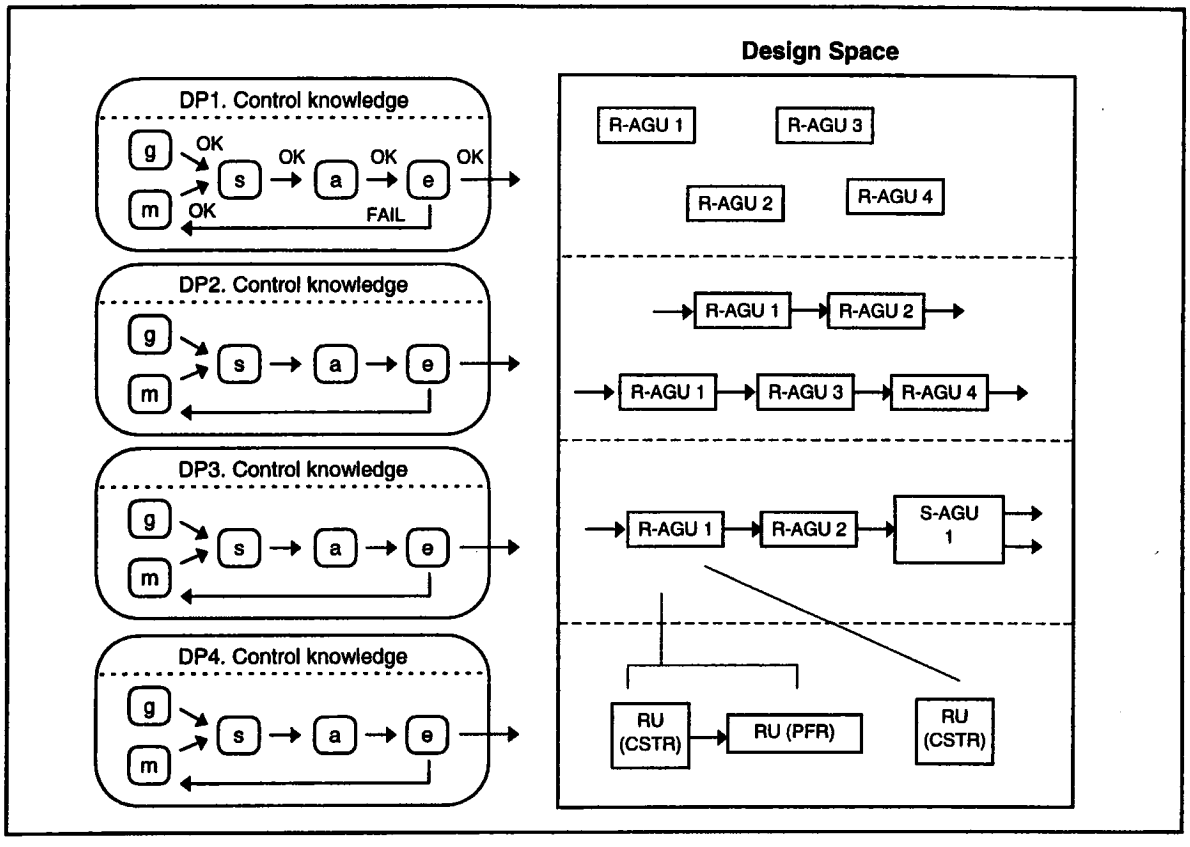


**Figure 2.** Design phases and Design Space

### *Role Assignment*

The aim of this task is to derive, for a given design case, the desirability of the occurrence of phenomena. Phenomena are taken here to be reactions and gas/liquid phase changes. The results have a global validity and are therefore design independent. Besides, they are independent of the units in which these phenomena may possibly occur.

This desirability is expressed by assigning 'roles' to all phenomena that might occur. In the following table all roles are shown and some of the situations in which they are assigned (R denotes a reaction, C a component).

In the design phases 1 to 3, that have a Boolean character (e.g. component is present or not, phenomena occurs or not), only the roles 'needed', 'no-impact' and 'not-tolerable' are used. A higher resolution is needed for making decisions based on the other two roles; this type of reasoning is deferred to the other two phases.

| Roles | Reactions | Gas / Liquid Phase Changes |
|---|---|---|
| needed | R lies on the reaction path to primary product | C is a reactant of a needed reaction but is first present (in feeds or as a reaction product) in 'other' phase |
| desirable | R is not needed but lies on the reaction path to valuable by-products | - next reaction on reaction path proceeds in other phase<br>- only primary product changes phase (a high purity product stream is created) |
| no-impact | R is a side-reaction that does not disturb the needed and desirable reactions | the presence of C in the other phase does not disturb the process |
| not-desirable | - R consumes reactants of needed reactions, and decrease thus selectivity<br>- R produces components that are toxics in the 'other' phase | - C is a reactant of a needed reaction that proceeds in the current phase (don't let C escape from it)<br>- C is in other phase a direct precursor of a toxic |
| not-tolerable | R produces components that are toxics in the phase in which the reaction occurs | C is in other phase a toxic |

### *Design Phase 1: design R-AGU*

In the first design phase physically feasible R-AGU alternatives are created by grouping needed phenomena. The goal here is to quickly get a rough overview of the number of reaction units needed to implement the chemical process.

Figure 3 shows the input and output entity types of the tasks of this design phase. 'Phena Grouping' is here the main local entity type. Entities of this type are created in the generate-task; selected for further examination in the select task; simulated in the analyse task; evaluated in the evaluate task; depending on the outcome of the evaluate task, stored as a R-AGU in the Design Space or given another chance in the modify task; and finally, discarded in the modify task if no cure is possible. Figure 3 shows for Phena Grouping only its newly derived properties.

#### Generate

The phenomena that have been assigned the role 'needed' in the Role Assignment task form a phenomena chain, with the last phenomena being the needed reaction that produces the desired product. These consecutive phenomena are now grouped into different sets. A group of phenomena is only considered if there is a temperature range at a certain pressure in which they all can proceed, and if in this temperature range no not-tolerable phenomena take place. For these derivations reaction information is needed (not the numerical part) and the boiling points of components.

The operating temperature range of a Phena Grouping is found, for a certain pressure, in the following way:

$$T_{min} = \max\{\ T_{min}(r_{needed}),\ T_{max}(r_{not\text{-}tol}),\ T_{bp}(\text{c-eva}_{needed}),\ T_{bp}(\text{c-cond}_{not\text{-}tol})\ \}$$

$$T_{max} = \min\{\ T_{max}(r_{needed}),\ T_{min}(r_{not\text{-}tol}),\ T_{bp}(\text{c-cond}_{needed}),\ T_{bp}(\text{c-evap}_{not\text{-}tol})\ \}$$

where $T_{min}$ and $T_{max}$ are the temperature ranges of a reaction r, $T_{bp}$ denotes the temperature of the boiling point, $\text{c-evap}_{needed}$ a component that should evaporate, etc.

The output of this task is a proposed phenomena grouping, described by: the names of the needed phenomena, a pressure and temperature range, and the needed input components, i.e. those components that are the reactants of the first reactions in (this part of) the phenomena chain.

#### Select

Very many phenomena groupings may be possible if many reactions and phase changes are involved. In this task heuristics are used to select one grouping for closer examination, if the Generate or Modify task have produced more alternatives. Preference is now given to Phena Groupings with many needed phenomena because this may lead to less units in the final process.

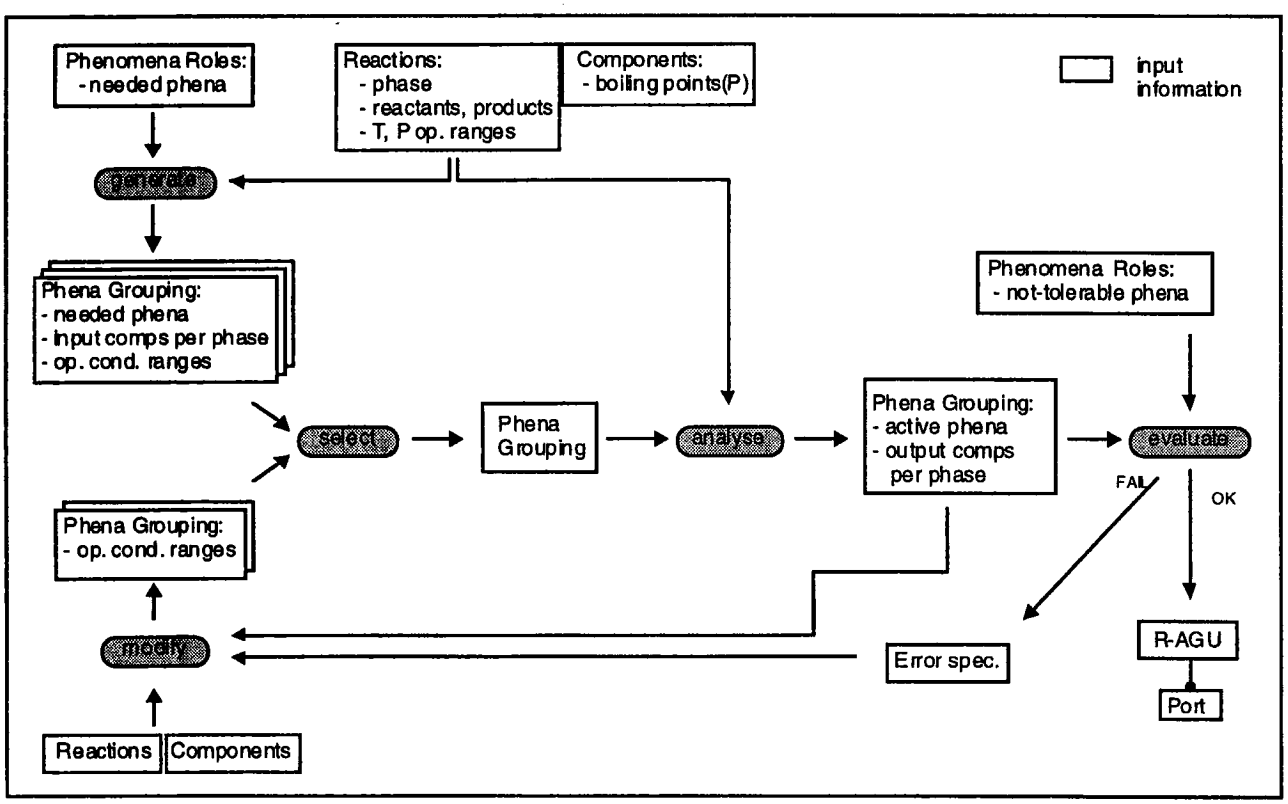


**Figure 3.** Inference structure of Design Phase 1

## Analyse

In this task a Boolean simulation is performed: based on the assumed input components and the operating condition ranges specified in a Phena Grouping, all phenomena that may proceed are derived. This set will include the needed phenomena grouped together, but also all side reactions and not-needed phase changes. From the active phenomena the output components are derived.

A reaction is marked as active if 1) all its reactants are present in the phase in which the reaction occurs and 2) there is an overlapping T and P range between the operating condition ranges of the Phena Grouping and that of the reaction. An evaporation or condensation is active if a component is present in a phase that is not its equilibrium phase for the operating conditions assigned to the grouping. This can happen if it enters the unit or if it is produced by a reaction in the non-equilibrium phase.

The following two rules show how the derivation of active liquid reactions is done. The first rule unpacks the needed information from the Phena Grouping, the second one fires as long as new active reactions are found ($? denotes a list-variable, ? a single-field variable).

```
* (phena-grouping (l-in-comps $?in-l) (Trange ?Tmin ?Tmax)))
                              ; get needed input information
  =>
  (present-l-comps $?in-l)
  (Trange ?Tmin ?Tmax)

* (Trange ?Tmin ?Tmax)
  (present-l-comps $?l-comps)  ;the currently derived comps
  (active-l-reactions $?a-r)   ; the currently derived reactions
  (reaction (id ?rid) (phase L)(reactants $?rs) (products $?ps))
        (Trange ?rTmin ?rTmax) ; a liquid-phase reaction
  (not (member ?rid $?a-r))    ; that has not been derived yet
  (subset $?rs $?l-comps)      ; its reactants are present
  (not (or (< ?rTmax ?Tmin)(> ?rTmin ?Tmax)))
                               ; and the T range is OK
  =>
  (active-l-reactions (new-list ?rid $?a-r))
                       ; a new reaction has been derived
  (present-l-comps (new-list $?ps $?l-comps))
                       ; and its products will also be present.
```

Because in this simulation no numerical data about reaction kinetics is used, it is not possible to derive how fast reactions occur and if the conversion of certain reactants is 100 % [3]. Thus all reactants and products of the derived reactions will be derived as output components in the right phase. Similarly, no numerical data about phase equilibria is used, so if phase changes occur it will not be known what the mole fractions of a component in both phases are (e.g. if it completely vanishes from a phase). Moreover, it is also not known if this phase equilibrium will be reached in the unit, something that is often assumed during conceptual process design. Therefore the reasoning system will derive that such a component will be present both as a gas and a liquid output component. Figure 4 illustrates the derivation of output components; the reactions and phase changes shown inside the boxes have been derived during simulation.

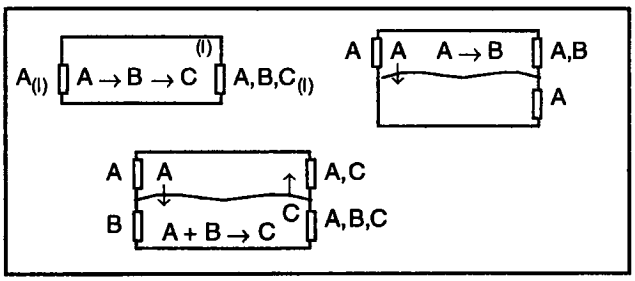


**Figure 4.** Example of output component derivation

## Evaluate

In the Analyse task not-tolerable phenomena may have been derived that where not detected during in the generate step. This happens because the initial checking was only performed based on the components of the needed reactions. If not-tolerable phenomena are derived and thus toxics are present, then the Phena Grouping does not pass the evaluation and the names of the causing phenomena and toxics is asserted. Otherwise no further information is derived and the Phena Grouping ends up as a R-AGU with associated PORTs in the Design Space.

The role information is needed to identify not-tolerable phenomena and components. The following rule shows how not-tolerable reactions are found in a Phena Grouping.

```
(phena-grouping (id ?id) (active-phena $?pha))
                  ; phenomena that have been derived
(phena-role (id ?p-id)(type REACTION)(role NOT-TOLERABLE))
                              ; a NOT-TOLERABLE reaction
(member ?p-id $?pha)          ; and it is found in the grouping
=>
(ntol-r-found (pg ?id) (reaction ?p-id))
```

## Modify

This task gets as input an analysed Phena Grouping and the name of some not-tolerable phenomena that have been found to be occurring. Here it is tried to adjust the operating range such that these unwanted phenomena will not occur anymore. This repair operation is performed in the same way as in the Generate task when not-tolerable phenomena are found. If the operation does not succeed then the task fails and the Phena Grouping is discarded.

### *Design Phase 2: Connect R-AGUs*

In this design phase the first flowsheets are produced: for a chosen set of R-AGUs it is decided how to assign the feed streams and how to connect the R-AGUs by intermediate

[3] Note that only irreversible reactions are considered at the moment, so no equilibria states must be taken into account.

streams, such that all R-AGUs get their needed input reactants. The produced flowsheets are worst cases with respect to impurities entering the R-AGUs. The local entity type of this phase is a 'flowsheet', containing units and streams.

### Generate

In the generate tasks R-AGUs from the Design Space are assigned input streams and connected to each other. The input-PORT information of a R-AGU is used while assigning it its input stream(s), since here it is stated which components the R-AGU needs for performing its function. Eventually all valid R-AGU groupings and interconnections are generated.

The procedure is as follows. First a grouping of R-AGUs is selected that together have all needed phenomena of the whole process occurring in them. Then the stream assignment and creation phase starts. A feed stream is assigned to a unit if the unit needs one of its components and does not get it already from elsewhere. Similarly, an intermediate stream is created between two units if the first one has components in its output port that the latter can use (and it does not get them from elsewhere). All additional components entering a unit other than the needed, are added to those already derived as output components. This propagation is needed for the stream assignment procedure. For example, suppose that a feed stream containing components A and D enter a first unit but D is only needed in a second. Then it should be possible to derive that the flowsheet of Figure 5a is a feasible alternative.

If at the end of the stream assignment step there still are units that don't get part of their inputs, then the process is not feasible; additional feed components are needed (or the declaration of new reactions).

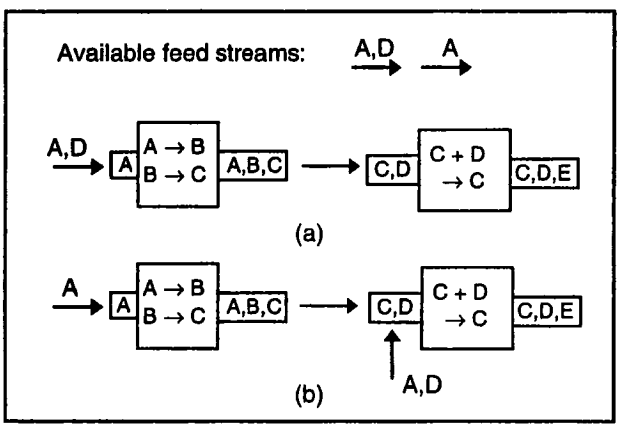


**Figure 5.** Example of stream assignment

### Analyse

During this analysis task again a Boolean simulation is performed. In contrast to the Boolean simulation of Design Phase 1, here a whole flowsheet is simulated: given the components in the feed streams and taking into account the operating ranges assigned to the units, all active phenomena and present components in output streams are derived.

For every unit the following steps are performed:

1. derive incoming components and their phase from incoming streams information
2. perform Boolean simulation: derive active phenomena and present components (same as in DP1)
3. derive components of outgoing streams (liquids and/or gas).

### Evaluate and Modify

These tasks have a similar functionality as those in DP1. During the Analyse task it may turn out that, due to unforeseen input components, not tolerable phenomena occur. Then it is again the task of Modify to look for a more restricted operating condition range. Only the first unit that contains these unwanted phenomena is adjusted during a repair and test phase; if this succeeds then the problems in the following units may also have been solved (see following example).

## Example

In this section the design phases 1 and 2 will be illustrated by an example. Only liquid-phase reaction phenomena will be used. Figure 6 shows the reaction network that is taken into consideration. To the right of it the temperature ranges in which the reactions have reasonable speeds are shown. The component specification is: E is the primary desired product, and T1 and T2 are toxics with 0 mole fraction allowance. The feeds specification states that two feed streams are available: one containing pure A, and one containing A and D (the former being most probably more expensive).

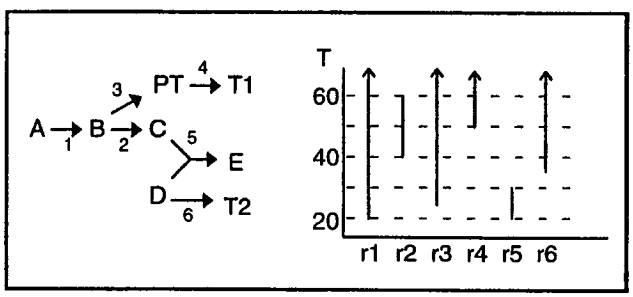


**Figure 6.** The example reaction network

**Role Assignment** will assign the role 'needed' to reactions r1, r2 and r5; 'not-desirable' to r3, and 'not-tolerable' to r4 and r6.

#### Design Phase 1

**Generate** tries first to propose a phenomena grouping in which all needed phenomena occur; this does not succeed because the temperature range of r5 is quite different from

those of r1 and r2. In a second trial two groupings are proposed, g1 and g2: the first one containing r1 and r2, the other containing r5. The temperature range of r2 determines the operating temperature assigned to g1; given this range and the components of r1 and r2 no not-tolerable phenomena are found to be occurring. For g2 the operating temperature determination is even easier; here no not tolerable reactions occur because of the temperature range in which r5 should be operated (otherwise r6 may also proceed, its reactant D is already present in the fluid !).

Suppose that **Select** chooses g1 first for examination. Based on the input components of g1 (A), **Analyse** deduces four active reactions r1 to r4. **Evaluate** then detects an active not tolerable reaction: r4. The Phena Grouping is now passed to **Modify**; in this case it is possible to restrict the T range. After a second Analyse and Evaluate round, g1 passes the test and is asserted as the first R-AGU in the Design Space. The second Phena Grouping g2 passes the test without problems and also ends up in the Design Space.

**Design Phase 2**

Given the two R-AGUs present in the Design Space, **Generate** finds out that only one R-AGU grouping is possible in which all needed phenomena occur (only two R-AGUs have been proposed). Thus design alternatives are only possible due to different stream assignments. Suppose that two flowsheet alternatives are generated (see Figure 5a and 5b). In the first one, fs1, only one feed stream is used (the one containing both A and D); in the second alternative, fs2, both streams are used: pure A entering unit 1, A and D entering unit 2. In both cases the interconnection by intermediate streams is trivial.

Flowsheet fs1 is first analysed. **Analyse** derives reactions r1, r2, r3 and r6. But, as **Evaluate** finds out, r6 is not tolerable ! **Modify** tries to repair the problem but: alas ! There is no cure possible: r6 takes place in the whole temperature range of unit 1. So flowsheet fs1 is no option anymore. Flowsheet fs2 is luckier: it passes the tests without any problems and ends up as the first PLANT alternative (containing only R-AGUs) in the Design Space.

## Discussion and related work

In the last years several applications of qualitative reasoning to chemical engineering tasks have been published. Automatic generation of qualitative models, given a chemical process description and operating assumptions, was reported in (Catino e.a., 1991). Here the QPT modeling framework (Forbus, 1984) was used to build model fragments of interest to the chemical engineer: descriptions of phenomena (e.g. equilibria reactions, heat transfer) and the behaviour of operating units (e.g. units with countercurrent flows). In (Dalle-Molle and Edgar, 1990) the use of the QSIM formalism (Kuipers, 1984) is discussed for modelling several types of reaction units and reaction systems.

For a related domain, design of thermodynamic cycles, (Everett, 1995) shows how the function of a process can be inferred from its flowsheet description. This type of reasoning is the opposite of the one described in this article; we try to produce a flowsheet given a functional description.

The only example of QR for chemical engineering design is, to our knowledge, the work of (Sgouros, 1993). There are several similarities between his and our work. First, we also make the distinction between, and strive for an integration of 1) design rules, that propose design alternatives, and 2) models at different resolution levels, that describe physical reality. Further, by using qualitative model knowledge during the synthesis tasks (i.e. generate and modify), we also wish to be exclusive in the generation design alternatives. This gives us the certainty that the 'best' alternatives will eventually be proposed, a situation that does not occur when only associational or heuristic knowledge is used (Simmons, 1992). Finally, we also use heuristics to select first the more promising designs for closer inspection.

A difference between our approaches is the contents of the qualitative models that are instantiated: in our case only the active phenomena are derived, but no quantity spaces are set up. The idea is to make first a rough simulation of all phenomena that may be active and to use this phenomena list while setting up the numerical model. For this last activity the phenomena to be taken into account must be known when describing the rate of change of a component. For the future it would be nice to have, even in the very first design rounds, a smart design assistant around, that predicts several possible behaviours depending on parameter ranges (Forbus, 1990).

To conclude this discussion, some words about the relation of our design method and the one described in (Douglas, 1988). A main feature of his method is a top-down, unit oriented, design strategy. The design process starts with the design of a design alternative at the highest level of aggregation: the whole process is seen as a black box with entering feed streams and leaving product streams. In the following design phases the design alternatives are refined by decomposing them into subsections until the lowest level of aggregation is reached: the black boxes denote unit operations. However, problems arise with this strategy when process sections are dependent on each other, as is the case when many interacting phenomena occur. In these cases it is better to follow a phenomena-oriented design strategy: make first a rough, qualitative analysis of expected phenomena interaction before fixing the process structure on the highest aggregation level.

## Conclusion

In this paper a design method for conceptual process design has been presented, developed to handle chemical processes which many interacting phenomena. The design method is characterised by 1. a division into qualitative and

numerical decision-making phases in which models with corresponding resolution are used; 2. a design case dependent, top-level control strategy; 3 the performance of Propose-Critique-Modify task types in all design phases; and 4. the certainty that the qualitative design that contains the optimal numerical design, will eventually be generated.

Future work will focus on investigating qualitative reasoning techniques that have a higher resolution than the one used at present, for example by using inequality reasoning. Another interesting research area is the management of design alternatives, e.g. alternatives described by different resolutions, describing the process at different aggregation levels, having slightly different functional specifications or assumptions about incoming streams.

## References


Banares-Alcantara, R. 1994. Design Support Systems for Process Engineering: II. KBDS: An Experimental Prototype. *Computers in Chemical Engineering*.

Catino, C.A., Grantham, S.D. and Ungar, L.H. 1991. Automatic generation of qualitative models of chemical process units. *Computers in Chemical Engineering* 15(8):589-599.

Chandrasekaran, B. 1990. Design Problem Solving: A Task Analysis. *AI Magazine* (Winter) 59-71.

Dalle-Molle, D.T. and Edgar, T.F. 1990. Qualitative Modeling of Chemical Reaction Systems. In: *Artificial Intelligence in Process Engineering*. Academic Press: pp. 1-36.

Douglas, J.M. 1988. *Conceptual Design of Chemical Processes*. McGraw-Hill Book Company.

Everett, J.O. 1995. A Theory of Mapping from Structure to Function Applied to Engineering Thermodynamics. In Proceedings of IJCAI, pp. 1175-1182.

Forbus, K. 1990. Qualitative Physics: Past, Present and Future. In: D. S. Weld, e.a. (ed.), *Readings in Qualitative Reasoning about Physical Systems,* Morgan Kaufmann, pp. 11-39.

Forbus, K.D. 1984. Qualitative Process Theory. *Artificial Intelligence* 24:85-168.

Gavrila, I.S. and Iedema, P. 1996. Phenomena-driven design, a knowledge-based approach. In Proceedings of ESCAPE 6. Pergamonn Press. Forthcoming.

Han, C., Stephanopoulos, G. and Douglas J. 1995. Automation in Design: The conceptual synthesis of chemical processing schemes. In Proceedings of ESCAPE 5.

Kuipers, B. 1984. Commonsense Reasoning about Causality: Deriving Behaviour from Structure. *Artificial Intelligence* 24:169-203.

Levenspiel, O. 1972. *Chemical Reaction Engineering*. John Wiley & Sons, New York.

Nishida, N., Stephanopoulos, G. and Westerberg, A. 1981. A Review of Process Synthesis. *AIChE Journal* 27(3):321-350.

Sgouros, N. 1993. Representing Physical and Design Knowledge in Innovative Design. Ph.D. diss, Northwestern University, Evanston, Illinois.

Simmons, R. 1992. The roles of associational and causal reasoning in problem solving. *AI Journal* 53:159-207.

Stephanopoulos, G., Henning, G. and Leone, H. 1990. MODEL.LA. A modeling language for process engineering-I. The formal framework. *Computers in chemical engineering* 14(8):813-846.

Weld, D.S. 1992. Reasoning about model accuracy. *Artificial Intelligence* 56:255-300.