

## MODELING AND UNDERSTANDING DIFFERENT TYPES OF PROCESS DESIGN ACTIVITIES

M. EGGERSMANN<sup>\*</sup>, S. GONNET<sup>‡</sup>, G. P. HENNING<sup>†</sup>,

C. KROBB<sup>\*</sup>, H. P. LEONE<sup>‡</sup>, W. MARQUARDT<sup>\*</sup>

<sup>†</sup>INTEC, Güemes 3450, 3000 - Santa Fe, Argentina  
ghenning@intec.unl.edu.ar

<sup>‡</sup>INGAR/UTN, Avellaneda 3657, 3000 - Santa Fe, Argentina  
{sgonnet, hleone}@ceride.gov.ar

<sup>\*</sup>Lehrstuhl für Prozesstechnik, RWTH Aachen, Templergraben 55, D-52056 Aachen, Germany  
{eggersmann, krobb, marquardt}@lfpt.rwth-aachen.de

**Abstract**— One of the major tasks addressed by the chemical industry is the design and revamping of production processes. Increasingly powerful computer-aided tools are available to face these complex tasks. Nevertheless, most design knowledge still rests in the minds of experienced designers. It is desirable to make it part of a computer support environment. Therefore, it is necessary to have a model of the design process. This contribution addresses this objective by introducing a model, based on the identification of three types of design activities: *Synthesis*, *Analysis*, and *Decision*. We discuss the intrinsic characteristics of each type of activity from two different view points: characteristic subactivities and products. Every type of activity consists of typical subordinate subactivities which are distinctive for the type and determine its behavior. On the other hand, activities operate on the results or products of the design process, called product data, including requirements, the representation of the design artifact itself, and arguments. These products also allow a specification of the three types. These ideas are exemplified by modeling the design of a separation system.

**Keywords**— Design Process Modeling, Modeling Languages.

### I. INTRODUCTION

Design in process engineering is a very complex and not completely mastered activity. The challenges it raises have motivated a lot of research aiming at understanding, systemizing, and improving the design process. Han et al. (1996) proposed a methodology having three components: planner, scheduler, and designer. These authors distinguish four design tasks during every stage of the design process: Synthesis, Analysis, Evaluation, and Refinement. Synthesis is the activity of generating structural designs, while Analysis solves the material and energy balances for the synthesized structure. During Evaluation, the economic potential of the artifact

being designed is calculated. Refinement is concerned with activities that allow the evolution from abstract to detailed models. We do not completely agree with this view but see Evaluation as a certain kind of Analysis; there is no conceptual difference between calculating a mass balance or an economic potential. In our view, Refinement of a model is a sequence consisting of *Synthesis* (*S*), *Analysis* (*A*), and *Decision* (*D*). In contrast to Han, we give a detailed characterization of these activities by specifying both their typical products and their distinctive subactivities. Linking activities are introduced which define the connection between the three main types of activities. Han's view of the design process is adapted from Douglas' (1988) hierarchical design procedure and does not directly apply to other design methodologies. It is limited to the synthesis of structural designs and focuses on the economic potential as a target function. This approach does neither take into account the refinement of a design and the underlying models across the design life cycle nor the modeling of design decisions and their underlying rationale. Precisely this last issue has been the aim of some other authors (King and Bañares-Alcántara, 1996; Ballinger et al, 1993; Brice and Johns, 1999) who recognize that design rationale should be a key component of the knowledge management strategy of any organization.

Depending on the domain and on the problem being addressed, design methodologies can vary. Boyle (1989) proposes a classification that splits design into three broad methodologies: analytical, procedural, and experimental design. Behind this classification is the concept of design objects, attributes of objects and operations on objects, as well as of the different roles that are assigned to humans and machines in these classes of design methods. The three categories proposed by Boyle can be summarized as follows:

(i) *Analytical or attribute centered design*, in which the attributes of the objects are used to determine the appropriate design actions. A design solution is automatically synthesized from the object attributes and the design objectives.

(ii) *Procedural or operation-centered design* is based on using procedures to perform operations on an object with the aim of transforming it into one having the desired attributes.

(iii) *Experimental search or object-centered design* involves working through an available set of objects in order to find one whose attributes best match the design objectives.

In this paper the focus is on procedural design, which is the most frequent case in chemical engineering. In this contribution design is viewed in a way similar to that of Boyle (1989) and some other authors (Brown and Chandrasekaran, 1989; Mittal and Araya, 1986) who consider it as an iterative process that operates under a generate-test-analyze-advise-modify paradigm. As design artifacts are generated, they are checked against design objectives. The design of a chemical process does not follow a predefined workflow and is not predictable in advance. In fact, many non-trivial design tasks are performed interactively; a person or a design team works with a computer-aided system in an attempt to solve a problem. Nevertheless, three typical sorts of activities can be identified. These types are considered to structure the design process as a first step. The aim of this paper is not to formally define the three types, but to provide a deeper understanding of what is happening during the different activities, as a basis for an ultimate (re-)engineering and computer-aided support of chemical engineering design processes. This paper discusses the intrinsic characteristics of each type of activity, explaining why a task is labeled S, A or D, and which are the specific product data it operates on. Between any two activities of different type linkage tasks take place. Three distinct "connection" sub-activities are characterized and presented in the paper. Moreover, the proposed model recognizes the fact that certain design tasks are of aggregated type (e.g. synthesis and analysis as one undistinguishable activity) and that the temporal order not always assumes the typical sequence of S, A, and D. These ideas are exemplified by modeling the design of a separation system.

## II. TYPES OF ACTIVITIES

### A. Main Types of Design Activities

During a design process three main types of activities can be identified: *Synthesis*, *Analysis*, and *Decision*. Whereas a *Synthesis* activity creates design objects, like process flow diagrams (PFD), reaction pathways, or mathematical models, an *Analysis* activity generates data about the design artifact, providing values for its characterizing attributes. Generally, *Analysis* is used to simulate behavior and to predict the performance of a prospective design for its intended use largely from some lumped interpretation of detailed calculations. Then, based on the available information, a *Decision* activity resolves whether the design artifact is selected, rejected, or kept in mind as a possible alternative. A

design artifact may be product data, i.e. the result of the design process (a flowsheet, models of processing units, etc.) or the design process itself. In other words, not only products may be synthesized but also work processes. This contribution focuses exclusively on activities that operate on product data. The classification proposed here does not aim at covering every activity, because for very detailed activities like logging on to a computer system or documenting some results it may not make sense to classify them according to the proposed scheme.

This contribution provides a detailed description of the three activity types from two different points of view. The design process view characterizes the types regarding the subordinate activities and the design product view describes the specific products used.

### B. Characterization of Types by Typical Subordinate Activities

High-level activities may be described in terms of their subordinate activities, which are in part responsible for giving main activities characteristics that serve to assign them a specific type. Therefore, each type of activity has a corresponding set of distinctive subordinate activities. This is not to be understood as a formal classification; Schank and Abelson (1977) doubt if this is even possible.

**Synthesis subactivities.** Activities like *Propose*, *Add*, *Remove*, *Modify*, *Merge*, *Select*, and *Request* have been identified. The *propose* activity conveys an action of suggesting something new as product data, while *add* incorporates a new element or idea into existing product data or expands an existing one. In contrast to *add*, a *remove* subactivity deletes or eliminates an element that is already part of the product data. A combination of *add* and *remove* subactivities gives rise to a *modify* subactivity, which alters an element, value or idea associated to product data. *Merge* creates something new by combining already existing elements in a consistent way, *select* chooses something from a given set of possibilities and *request* solicits additional information, allowing a *Synthesis* activity to be interactive.

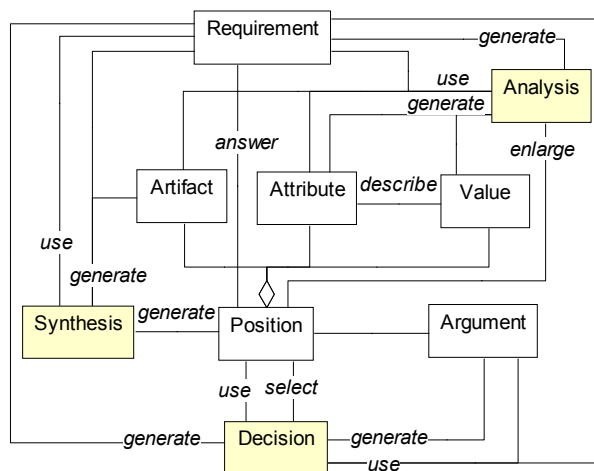
**Analysis subactivities.** *Calculate*, *Estimate*, *Determine*, *Experiment*, and *Request* can be grouped under this category. After a *Synthesis* activity, product data already exists, but some of its attribute values may be unspecified. These values are needed as a basis for subsequent *Decisions*. One of the most common cases is to perform calculations (*calculate*) in order to provide missing information. If calculations are not possible or too expensive, values can be *estimated*. In other situations, attribute values may be uncovered by studying literature or browsing databases (*determine*). The engineer may perform an *experiment* activity to test a proposed design and generate additional data.

**Decision subactivities.** These include *Select*, *Evaluate*, *Justify*, and *Request*. A *select* subactivity refers to the choice of one or more design products from a number of possible alternatives. Before, some information generated during analysis is compared with the requirements

to fulfill. Thus, an *evaluation* activity provides arguments to justify a decision. Similarly, *justify* offers a rationale for the selection of a certain alternative. The subactivity *request* solicits additional information, allowing a *Decision* to be interactive. As seen, requesting for new information can happen during every stage of the design process and is therefore not specific to any of the three types of activities.

### C. Product Data Participating in a Design Process

Besides having an associated set of subactivities, an activity type can also be classified by the kind of product data it uses and how it uses the data. Design activities operate on product data and are driven by requirements such as constraints, requests, goals, etc. Requirements specify the functions an artifact shall fulfill and its desired performance. The use of requirements is taken into account, but their generation is not investigated since it is too complex and a special research area on its own, called Requirements Engineering (Pohl, 1996). The following considerations are based on the Issue Based Information System (IBIS) (Rittel and Kunz, 1970). An *issue* is a question to be answered and a *position* is an alternative which exists for solving an issue. An *argument* either supports or objects a position. We refine this view by introducing *requirements*, which specify *issues*, and also by decomposing *positions* into *artifacts*, *attributes*, and *values* (See Fig. 1).



**Figure 1.** Simplified view of the relationships between Types and Products.

This classification of product data has the aim of providing a conceptual framework for keeping track of design driving forces (requirements) which may also include restrictions, mathematical constraints, and bounds as well as the underlying design rationale. *Synthesis* activities use requirements to generate artifacts which are part of positions and which may be missing relevant information. A position encompasses a design artifact such as a chemical reaction pathway, a flow-sheet structure, a mathematical model describing part of the chemical process being considered, the geometry of a piece of equipment, or the selection of a type of unit, its attributes and corresponding values. *Analysis* activi-

ties allow the enlargement of positions by providing attributes and values in order to have enough information to carry out *Decision* activities, which in turn use requirements and arguments to select positions. *Decision* activities are also responsible for building arguments. The function of an argument is to establish an evaluated link between a position and at least one requirement. It compares the position with the requirements and tests whether the position is capable of fulfilling the prescribed requirements. A position answers a requirement. Requirements are also used during *Analysis* activities to indicate the most important attributes to focus on. As seen, requirements are used in every activity, but with a different purpose or aim. Thus, *Synthesis* attempts to answer the question: "How can requirements be fulfilled?" and *Analysis* supplies data that will allow to check if requirements are reached. Finally, *Decision* activities weigh up requirements, establish which are the most important ones and test if these requirements are indeed met.

The inputs for a *Synthesis* activity are the issue, which is not depicted in Fig. 1, and the requirements. Moreover, new evaluation criteria may be generated during a *Synthesis* task as parts of a requirement. All this information is used by *Analysis* activities. Further inputs of *Analysis* tasks are the criteria that will lead the *Analysis*. These criteria are part of the requirements, but generally, the exact requirements are not necessary. For instance, it is sufficient to know that the purity of the distillate of a synthesized distillation column has to be calculated; the value of the minimum required purity is not needed for the analysis. Information about the design artifact in relation to certain pursued criteria is generated as output and enters a *Decision* task as input. *Decision* tasks may also need the relative importance of the evaluation criteria as input. This is either generated by the *Decision* task itself or required as an input. The output of a *Decision* activity is then the selection or rejection of a design product. Additionally, the *Decision* on how to proceed may be implicitly taken. The result might be that another *Synthesis* is necessary or that the *Analysis* has to provide supplementary data. This corresponds to either proceeding forward or going backwards.

In this proposal *Synthesis* activities are not restricted to create structural product data but may generate any kind of model; this makes it different from the approach of Han et al. (1996) where *Synthesis* only generates structural components. The generation of a requirement is not necessarily characterized as a *Synthesis* task. Indeed, requirements are not created on purpose but ad-hoc; this might happen during every type of activity.

### D. Decomposition of Design Processes

In a first approach a design process can be seen as an iterative sequence of *S-A-D* activities. Though in certain cases, it is possible to distinguish between the three main types of activities that participate in a design process, in others some sort of aggregation may appear. For instance, synthesis and analysis can be lumped and car-

ried out all at once, thus generating different alternatives along with all the necessary information to decide which is the most convenient one. Therefore, a decision activity can be performed afterwards. In some other cases, the three types of activities could be aggregated. This occurs when a computer aided-tool implicitly generates, evaluates and decides among different alternatives, like in the approach proposed by Floudas (1995) to synthesize heat exchanger networks.

Concepts introduced in Section II.A may lead to the assumption that a typical temporal order of activities takes place. In other words, after a synthesis task an analysis one occurs, which in turn is followed by a decision one. However, this is not always the case. In certain situations, during an analysis activity it may be discovered that a new synthesis task has to be undertaken. Similarly, a decision activity may reveal that it needs more information from an analysis task. In summary, the design process does not always follow the typical *S-A-D* sequence in a forward fashion. In addition to any of the sequential paths described above, activity types may also occur recursively. For instance, minor analysis and decision tasks can be performed during a synthesis one or the other way around.

### E. Connection Between Main Types of Design Activities, Requirements and Intermediate Activities

As it was already pointed out, requirements are the driving force of a design process, participating in the different types of activities with distinct roles. Moreover, they are the product data on which linkage or intermediate activities operate. These types of activities are part of the main ones and are responsible for “gluing” consecutive tasks. *Transform Requirement*, *Evaluate Requirement* and *Select Requirement* were identified as connecting activities (See Fig. 2).

At the beginning of an *Analysis* task, a *Transform Requirement* (*TR*) activity takes place. It converts the requirements into a set of relevant attributes, which is used as the input of the *Analysis* activity. Therefore, a *TR* task prepares the data before starting a genuine *Analysis* task. For instance, the requirement of environmental impact minimization can be transformed into

bounds on energy consumption and pollutant emissions. In turn, *Decision* activities begin with a *Evaluate Requirement* (*ER*) process, which transforms data from the *Analysis* into arguments on which the *Decision* will be based. This evaluation connects *Analysis* with *Decision* by comparing the artifact's actual behavior with the proposed requirements.

Finally, the gluing activity that occurs between a *Decision* task and its successor *Synthesis* activity is called *Select Requirement* (*SR*). As not all requirements are yet fulfilled (if they were, the design would be finished) it has to be decided which one shall be addressed next. Thus, *SR* is a sort of planning activity, which will choose the requirements to focus on during genuine *Synthesis* activities.

Consequently, from an abstract perspective, it can be seen that the main activities that were identified are linked by subactivities that allow to trail the following loop: *Select Requirement* – *Synthesis* – *Transform Requirement* – *Analysis* – *Evaluate Requirement* – *Decision* – *Select Requirement* – .... However, as it was pointed out before the design process does not always follow the *S-A-D* sequence in a forward fashion. Moreover, it is important to remark that Fig. 2 does not attempt to represent the algorithmic perspective of a design methodology.

### III. EXAMPLE

The design of the separation system associated to the recovery of ethylene and byproducts from the steam pyrolysis of light hydrocarbons is addressed. It is shown how the model of the design process proposed here fits the approach taken by Manley (1998) to tackle this problem.

Initially, the designer carries out a *Synthesis* activity and proposes a conventional design of the separation system. As input of this task he/she considers feed streams and output streams information (Table 1) and obtains as output a position containing a structural description of the system (Fig. 3). After pre-treatment of the feed, from which acetylene is removed by

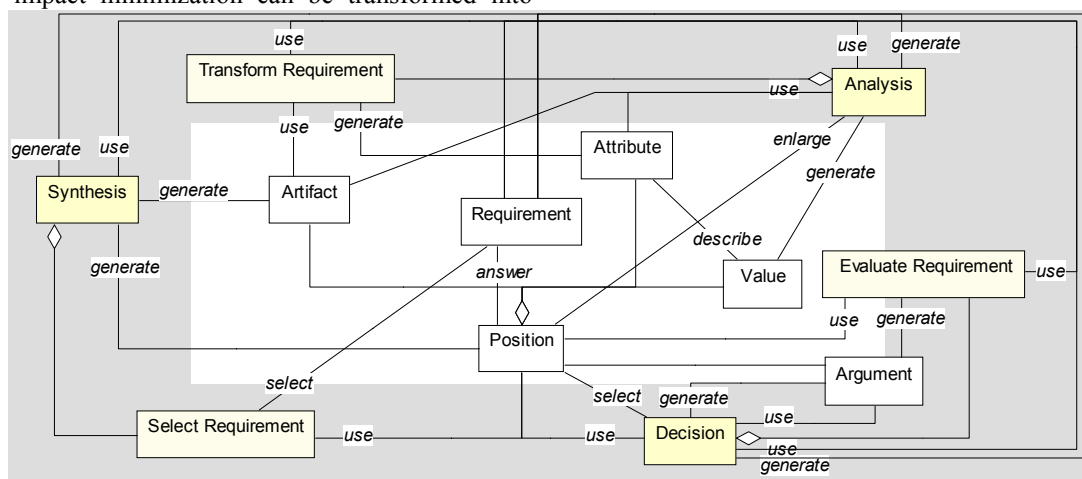


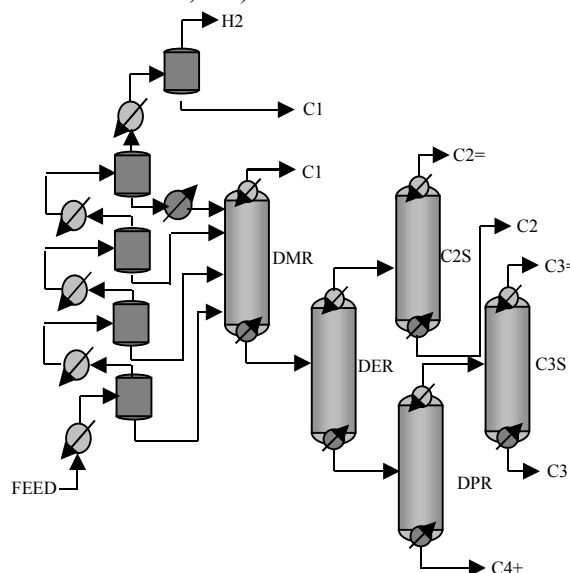
Figure 2. Proposed model for the classification of design activities.

hydrogenation and elimination of parts of hydrogen and methane, the liquids are fractionated under pressure into products in demethanizer (DMR), deethanizer (DER), depropanizer (DPR), C2 splitter (C2S) and C3 splitter (C3S) distillation columns. Requirements related to the minimization of capital and operating costs are considered, as well as the level of complexity of the system.

**Table 1.** Input stream and desired products associated to the separation system

	Streams						
	Feed	H2	C1	C2=	C2	C3=	C3
Temp °C	38.7	32.2	32.2	32.2	32.2	37.8	37.8
Pres Bar	1.4	34.5	6.2	111.3	6.9	15.6	13.0
Mole Percent							
H2	32.5	95.0	9.9				
CH4	4.8	5.0	90.0	0.05			
C2H4	31.7		0.1	99.90	0.5		
C2H6	24.6			0.05	99.0	0.05	
C3H6	0.5				0.5	99.90	0.5
C3H8	0.1					0.05	99.0
C4+	0.8						0.5
H2O	5.0						

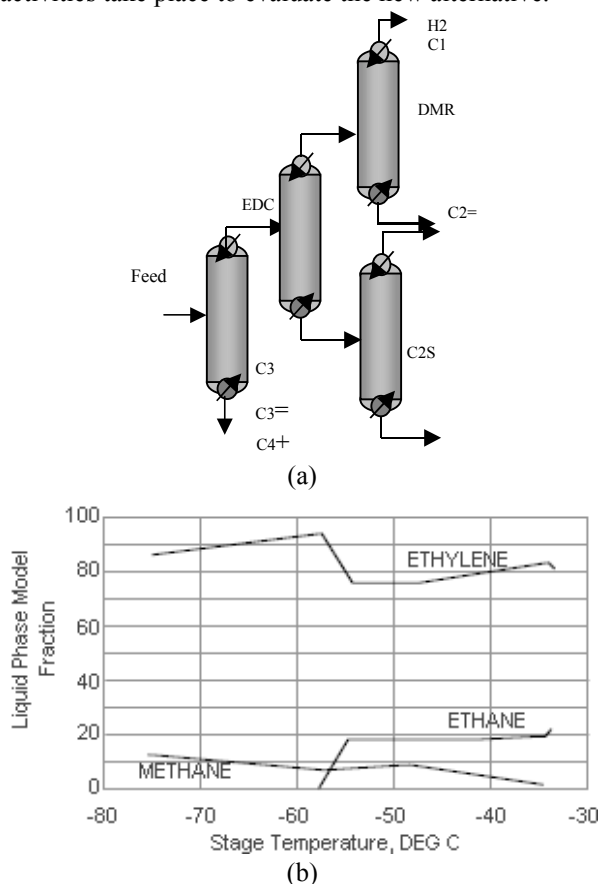
After this *Synthesis* activity an *Analysis* task takes place. It provides the relevant attribute values (utilities consumption, columns' number of stages, and reflux ratios, etc.) to later evaluate how well requirements are satisfied. This *Analysis* activity includes several other tasks of different type (e.g. select-type *Synthesis* subactivities that choose the right columns models, *estimate* and *calculate Analysis* subactivities that provide values to some physicochemical parameters and solve the models, *Decision* activities that judge the appropriateness of the models, etc.).



**Figure 3.** Conventional ethylene recovery distillation process.

Once the current position is enlarged with the calculated values, a *Decision* task takes place. It considers the current design as inadequate and its arguments are

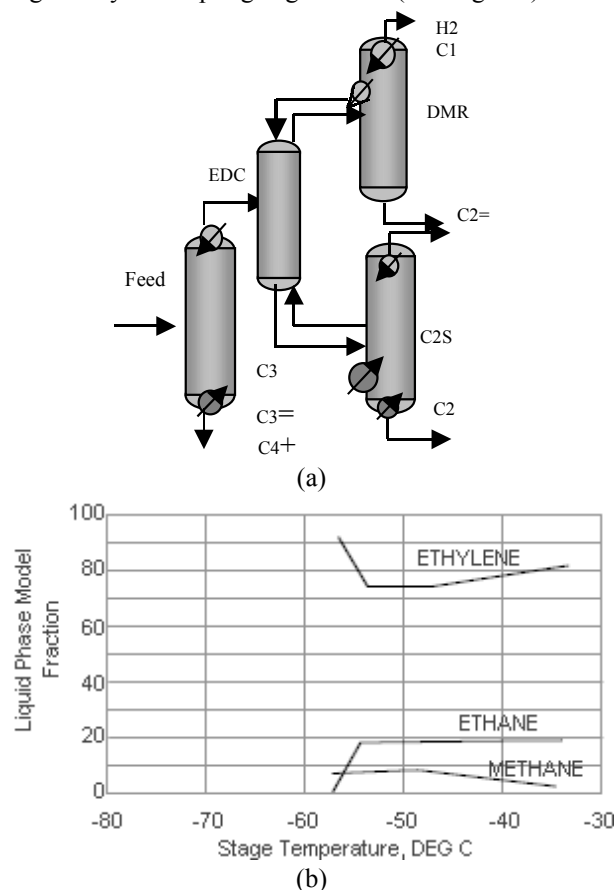
the high energy consumption of the system as well as the large sizes of the columns. By explicitly modeling the decision, the workflow is more easily understandable and later review of the design process is facilitated. In Han's approach (Han et al., 1996) *Decision* is not modeled as a separate activity and therefore the model is less comprehensible. A new *Analysis* task is demanded to determine the reason of such a poor performance. One of the causes the designer recognizes is the number of phase changes necessary to process the major component Ethylene (C2=). Ethylene enters the recovery process as a gas, is condensed in the DMR feed stream, is revaporized in the DER and is finally recondensed again as a liquid product from the C2S. Therefore, three complete phase changes must be accomplished for all the C2=. A similar analysis can be done for propylene. In consequence, a new requirement is added. It refers to the minimization of the number of component phase changes in the system. Then, another *Synthesis* activity takes place addressing the new requirement. The designer generates a new position that includes the use of an ethylene distributor column (EDC) (See Fig. 4.a), which reduces the number of phase changes and also the amount of ethylene which suffer such changes. Even though no details are given about this position here, new *Analysis* and *Decision* activities take place to evaluate the new alternative.



**Figure 4.** a) Separation scheme having distributed distillation of ethylene. b) Liquid composition profile for the ethylene distributor column.



It is concluded that the new alternative still is too expensive and a new *Analysis* task is demanded. It considers the liquid composition profile for the ethylene distributor column (See Fig. 4.b). The feed at about  $-54.5^{\circ}\text{C}$  is separated into C1 and C2= on the top stage at about  $-57^{\circ}\text{C}$  and into C2 and C2= on the bottom stage at about  $-34^{\circ}\text{C}$ . Due to the presence of H<sub>2</sub>, the condenser must cool the overhead to about  $-75^{\circ}\text{C}$  in order to generate the required reflux. Refluxing this condensate "remixes" C1 and C2= in the rectifying section, which have been more completely separated at the outlet of the column. Then, they must be "re-separated" in the downstream DMR column. To a lesser extent, a similar phenomenon occurs in the reboiler, where C2 and C2= are "remixed" and must be then "re-separated". After this *Analysis* a *Decision* activity creates new arguments that will lead to the rejection of this new design alternative. Moreover, the *Decision* activity creates an extra "avoid mixing and re-separating" requirement. Another *Synthesis* task takes place, and a new alternative that eliminates the mixing and separating phenomenon by resorting to recycle coupling is generated (See Fig. 5.a).

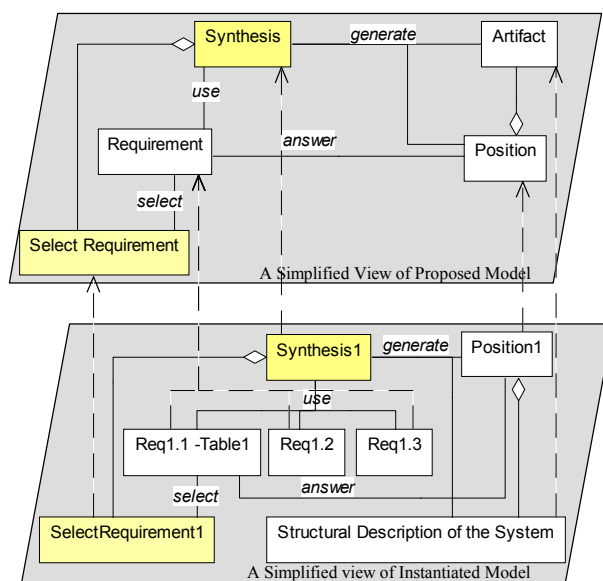


**Figure 5.** a) Ethylene distributor with recycle coupling. b) Liquid composition profile for the ethylene distributor column.

Then, a new *Analysis* activity takes place enlarging this position in order to determine whether requirements are met or not. As seen from Fig. 5.b, remixing is eliminated and the needed separations as well as the energy consumption are considerably reduced. Therefore, the

design is accepted by the final *Decision* activity.

This design process has been modeled using the ideas introduced in this paper. A simplified view of the first *Synthesis* activity performed is presented in Fig. 6, where Position1 is the position associated to the first separation scheme previously introduced. As seen, position Position1 is an aggregation of the Structural Description of the System, an instance of artifact. That position, representing a conventional ethylene recovery distillation process, has been generated by the *Synthesis* activity Synthesis1 by using a set of requirements (Req1.1, Req1.2, Req1.3).

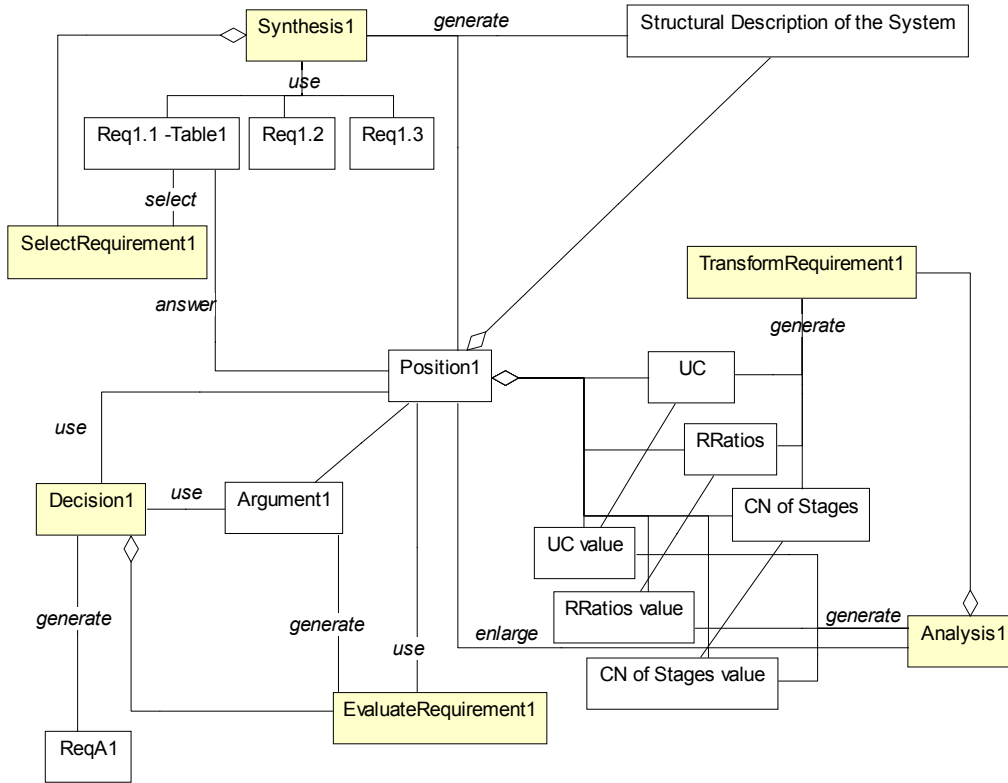


**Figure 6.** Overall view of the first *Synthesis* activity that was performed.

In Fig. 7 a simplified view of the first S-A-D sequence is presented. As seen, the first position has been enlarged by the *Analysis* activity Analysis1, which has generated values (UC value, RRs ratios value, CN of stages value) for the relevant attributes. Then, position Position1 is an aggregation of the Structural Description of the System, a set of relevant attributes such as utilities consumption (UC), columns' number of stages (CN of stages) and reflux ratios (RR ratios), among others, and their corresponding values.

The result of a *Decision* activity (Decision1) about this position is to consider it inadequate due to its high energy consumption and big sizes of the columns (Argument1).

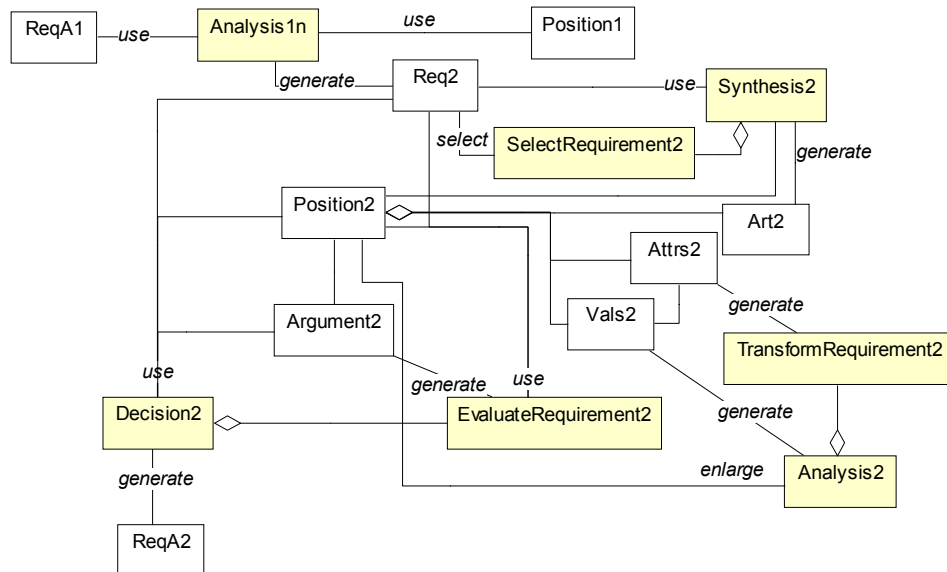
A new *Analysis* task (Analysis1n) is demanded to determine the reason of such a poor performance (Fig. 8). A new *Analysis* and *Decision* process (not completely shown in Fig. 8) will conclude that the "minimize the number of component phase changes in the system" requirement (Req2) has to be considered by the new *Synthesis* activity (Synthesis2). This task generates an alternative design represented by position Position2, which in turn is an aggregation of a structural description of a separation scheme having distributed distillation of ethylene (Art2), a set of attributes (Attrs2) and their corresponding values (Vals2).



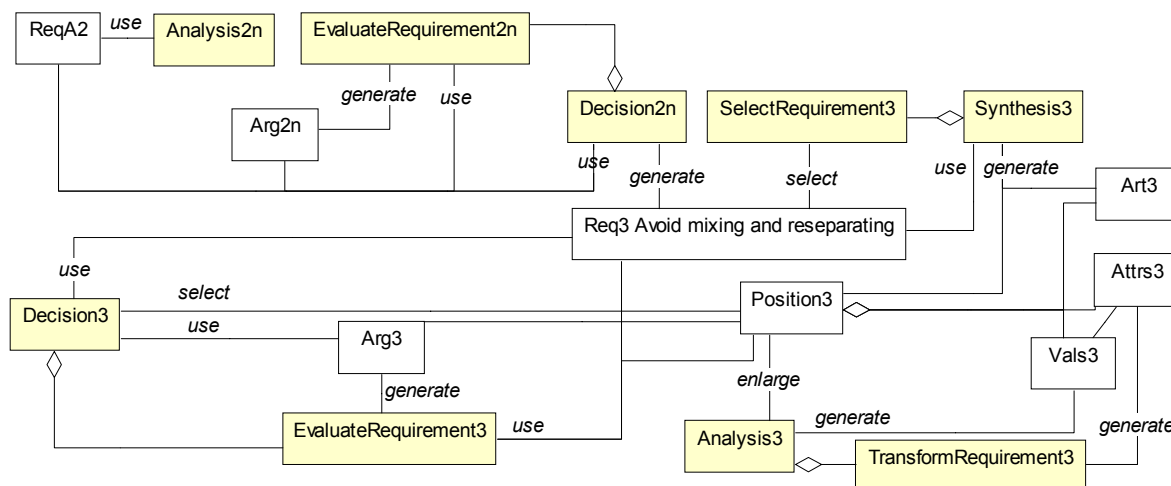
**Figure 7.** First sequence of S-A-D activities that was performed.

This new position Position2 is subsequently *analyzed* (Analysis2) to generate values (Val2) for the relevant attributes (Attr2) and a new *decision* (Decision2) is made about it, concluding that it is unsatisfactory. Once again, a series of *analysis* and *decision* processes (A2n, D2n), operating on a series of arguments (Arg2n) first enlarges Position2, later rejects the alternative and posts a new requirement (Req 3 Avoid mixing and re-separating) (See Fig. 9). This requirement will be selected by the SelectRequirement3 subactivity (along with some

other requirements not shown in the Fig. 9 for the sake of simplicity) to initiate a new *Synthesis* activity (Synthesis3) that generates Position3, an aggregation of the structural description of the separation scheme having an ethylene distributor with recycle coupling (Art3), its relevant attributes (Attr3) and their corresponding values (Vals3). A new series of *Analysis* and *Decision* activities takes place, this time concluding that this position is acceptable.



**Figure 8.** Second sequence of S-A-D activities that was performed.



**Figure 9.** Third sequence of S-A-D activities that was performed.

An analysis of Figs. 7, 8 and 9 illustrates the role of linkage subactivities associated to requirements, acting as driving forces in the design process. Moreover, this figures show *requirement selection* activities (SelectionRequirement1, SelectionRequirement2 and SelectionRequirement3) being components of major *synthesis* tasks (Synthesis1, Synthesis2 and Synthesis3 respectively), *requirements transformation* activities (TransformationRequirement1, TransformationRequirement2 and TransformationRequirement3) being part of *analysis* tasks (Analysis1, Analysis2 and Analysis3 respectively) and *requirements evaluation* activities (EvaluateRequirement1, EvaluateRequirement2 and EvaluateRequirement3) as parts of *decision* tasks (Decision1, Decision2 and Decision3).

#### IV. CONCLUSIONS

A classification of design activities into *Synthesis*, *Analysis*, and *Decision* has been presented. These types of activities were characterized from two different view points: typical subordinate activities and the design products they operate on. One of the products the three main activities deal with are requirements, which indeed are seen as the driving force of the design process. Moreover, requirements are the product data on which linkage activities (*Transform Requirements*, *Evaluate Requirements* and *Select Requirements*) operate on.

The classification introduced in this contribution helps to understand and to structure design processes and facilitates their planning within the framework of a support environment. In fact, future activities associated with this research line include the usage of the proposed characterization in combination with an actors' model (Eggersmann et al., 2001) to expand the current capabilities of the COPS (Context Oriented Process Support) environment (Krobb, 2002, Eggersmann et al., 2000).

The feasibility of this proposal has been demonstrated by modeling a separation process design example according to the ideas presented in the paper.

#### Acknowledgements

This publication is a result of a cooperation which is funded by the German Ministry of Education and Science (BMBF) and the Argentinean Secretary of Technology, Science and Productive Innovation (SeTCIP). The authors are grateful to acknowledge their support. They also want to thank the Deutsche Forschungsgemeinschaft for its funding for the CRC 476 IMPROVE and the Graduiertenkolleg Informatik & Technik. The Argentinean partners also appreciate the financial support from FONCYT under Grants 14-00000-01143 and 14-00000-07004, CONICET, Universidad Nacional del Litoral and Universidad Tecnológica Nacional.

#### REFERENCES

- Ballinger, G., R. Bañares-Alcántara and J.M.P. King, "Using an IBIS to record design rationale", Technical Report 1993-17, Department of Chemical Engineering, University of Edinburgh (1993).
- Boyle, J-M., "Interactive engineering systems design: a study for artificial intelligence applications", *Artificial Intelligence in Engineering*, Vol. 4, 2, 58-69 (1989).
- Brice, A. and B. Johns, *Improving design by improving the design process, A DRAMA white paper QSL-9002-WP-001*, Enviro Software Solution, <http://www.enviros.com/drama> (1999).
- Brown, D.C. and B. Chandrasekaran, *Design problem solving. Knowledge structures and control strategies*, Morgan Kaufmann Publishers, California (1989).
- Douglas, J.M., *Conceptual design of chemical processes*, McGraw-Hill, New York (1988).
- Eggersmann, M., G.P. Henning, C. Krobb and H.P. Leone, "Modeling of actors within a chemical engineering work process model", *Proceedings International CIRP Design Seminar*, Stockholm, Sweden, 6-8 June, 203-208 (2001).



- Eggersmann, M., C. Krobb and W. Marquardt, "A modeling language for design processes in chemical engineering", In: Alberto H.F. Laender, Stephen W. Liddle, Veda S. Storey (Eds.): *Lecture Notes in Computer Science 1920*, Springer, 2000, 369-382 (2000).
- Floudas, C.A., *Nonlinear and mixed-integer optimization: Fundamentals and applications*, Oxford University Press, Oxford, England (1995).
- Han, C., G. Stephanopoulos and J. Douglas, "Automation in design: the conceptual synthesis of chemical processing schemes", In *Intelligent Systems in Process Engineering*, Stephanopoulos Geo. and Han C. (Eds.), 93-143 (1996).
- King, J.M.P. and R. Bañares-Alcántara, "The extension and evolution of the process design representation", *Comp. Chem. Engng.*, Vol. **20**, Suppl., S171-S176 (1996).
- Krobb, C., *Workflow modeling and support for process modeling and simulation in chemical engineering*, PhD thesis. Lehrstuhl für Prozesstechnik, RWTH Aachen (in preparation) (2002).
- Manley, D.B., Thermodynamically efficient distillation: ethylene recovery", *Latin American Applied Research*, **28**, 1-6 (1998).
- Mittal, S. and A. Araya, "A knowledge-based framework for design", in *Proceedings of AAAI-86* (1986).
- Pohl, K., *Process-centered requirements engineering*, Wiley, New York (1996).
- Rittel, H.W.J. and W. Kunz., "Issues as elements of information systems", Institute of Urban and Regional Development. Working Paper, 131, University of California, Berkeley (1970).
- Schank, R. and R. Abelson, *Scripts, plans, goals, and understanding*, Lawrence Erlbaum Associates, Hillsdale, New Jersey (1977).

Received: September 16, 2001.

Accepted for publication: November 02, 2002.

Recommended by Guest Editors J. Cerdá, S. Díaz, and A. Bandoni.