SYMBOLIC SIMULATION OF CHEMICAL PRODUCTION PROCESSES

L. Winkelbauer

June 1987
WP-87-45

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

PREFACE

Applied systems analysis is – or should be – a tool in the hands of planners and decision makers who have to deal with the complex and growing problems of modern society. There is, however, an obvious gap between the ever-increasing complexity and volume of scientific and technological information and tools of analysis relevant to large socio-technical and environmental systems, and the information requirements at a strategic planning and policy level.

The Advanced Computer Applications (ACA) project builds on IIASA's traditional strength in the methodological foundations of operations research and applied systems analysis, and its rich experience in numerous application areas including the environment, technology, and risk. The ACA group draws on this infrastructure and combines it with elements of AI and advanced information and computer technology. Several completely externally-funded research and development projects in the field of model-based decision support and applied Artificial Intelligence (AI) are currently under way.

As an example of this approach to information and decision support systems, a major component of an R & D project sponsored by the CEC's EURATOM Joint Research Centre (JRC) at Ispra, Italy, in the area of hazardous substances and industrial risk management, is described in this paper. It focuses on the symbolic simulation of chemical production processes which provides an easy-to-understand explanatory system. It shows – by tracing the intermediate substances of the production process – how certain chemical production processes work. It is interconnected with several other simulation or optimization modules and databases of the larger information and DSS, and provides information about the feedstocks required, the interim products, the final products and the wastes of the chemical production process simulated.

By emphasizing a directly understandable problem representation, based on symbolic simulation and dynamic color graphics, and the user interface as a key element of interactive decision support systems, we attempt to make models of complex processes understandable and available to non-technical users. This is a step toward the increased direct practical usability of IIASA's research results.

Kurt Fedra
Project Leader
Advanced Computer Applications

TABLE OF CONTENTS

SYMBOLIC SIMULATION OF CHEMICAL PRODUCTION PROCESSES

L. Winkelbauer

1. INTRODUCTION

The safe and systematic management of hazardous substances is a basic requirement in order to ensure a substainable use of the biosphere for present and future generations.

To further this objective an *integrated risk management system*[*)] was developed to provide software tools for those engaged in the management of the environment, industrial production, products and waste streams, hazardous substances and wastes in particular (Figure 1).

This system comprises an information system (a set of interconnected knowledge and databases on hazardous chemical substances, industrial waste streams, production technologies of hazardous chemicals, accident reports, regulations of the EC, etc.) several simulation and optimization modules, and a graphics-based, menu-driven user interface which provides access to all the integrated modules for non-technical users as well as for experts.

The production process level of the IRIMS system concentrates on the symbolic simulation of chemical production processes which provides an easy-to-understand explanatory system showing (by tracing the intermediate substances of the production process) how certain chemical production processes work. The Production Process Module is interconnected with several other simulation/optimization modules and databases of IRIMS and provides information about the feedstocks required, the interim products, the final products and the wastes of the chemical production process simulated. It also allows for an *a posteriori* hazard evaluation of the whole process, for each hardware unit and each substance involved.

Symbolic simulation was chosen in order to enable simulation of chemical production processes without having to consider the huge amount of quantitative information called for in numeric simulation (Grauer and Fedra, 1986). The amount of quantitative information required certainly exceeds that normally within the grasp of non-experts in the chemical field – the target user group of the IRIMS system. The symbolic approach allows the simulation to be based on understandable (qualitative) rules without failing to represent the production process in chemical plants realistically.

2. THE INDUSTRIAL PRODUCTION SYSTEM

One of the components of IRIMS is the **Industrial Production System,** which represents consumption of raw material, energy, water, manpower, capital etc., and the generation of products, wastes and interim products (Fedra, 1985).

[*)]IRIMS, the Ispra Integrated RIsk Management Support system was developed by IIASA's Advanced Computer Applications (ACA) project under contract to the Commission of the European Communities (CEC), Joint Research Centre (JRC), Ispra, Italy.

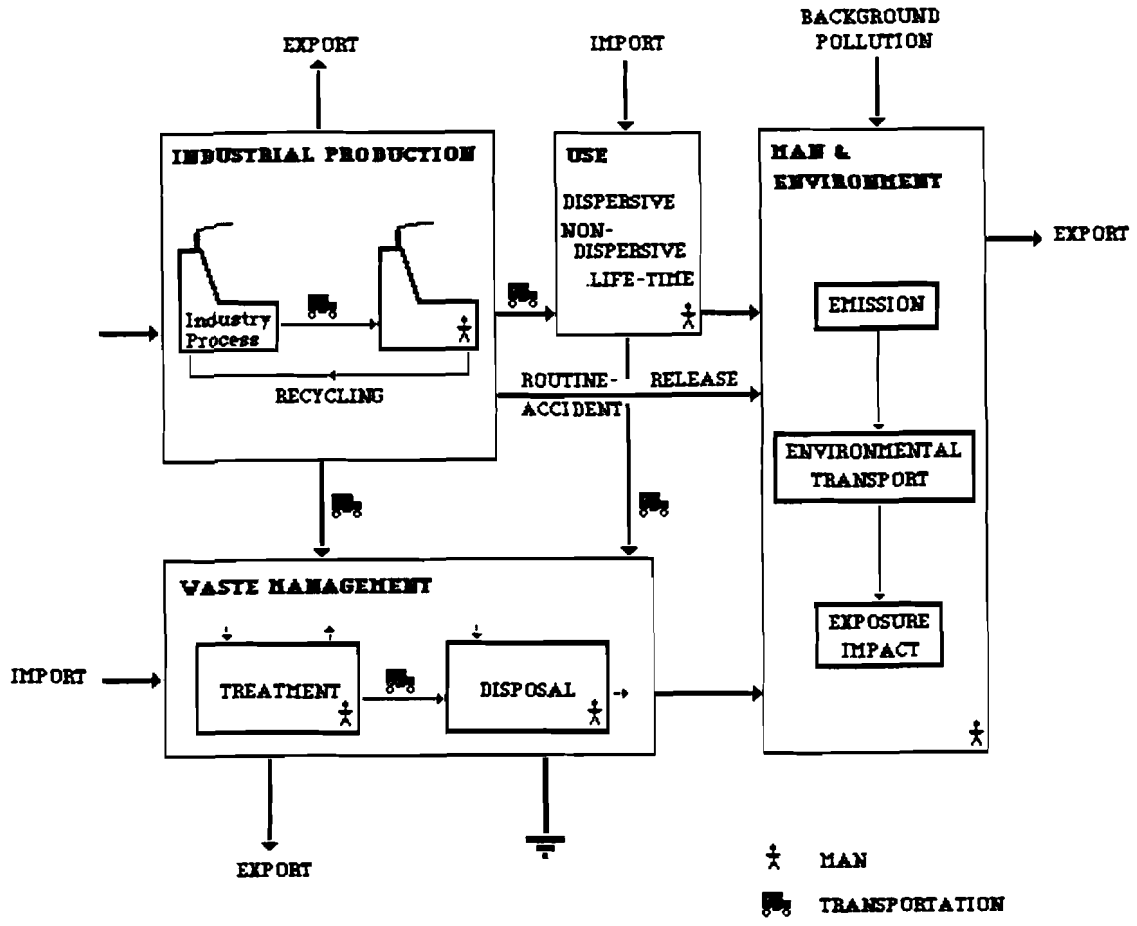

Figure 1: Conceptual elements of IRIMS (after Fedra, 1985, p. 14)

It estimates the amount of waste for a certain set of products, using a certain production technology or process *(normal operation mode)*, or it simulates an accident or gross mismanagement situation, where raw materials, interim products or final products may be released – more or less uncontrolled – into the environment *(accident mode)*.

In keeping with the first three levels of the decomposition hierarchy of Zanelli et al. (see Appendix A) conceptually there are three levels of software modules for the simulation of Industrial Production Systems (Figure 2).

The **Production Process Module** is the basic module in the Industrial Production System. It simulates the individual product or substance-oriented production processes, which are aggregated at the Chemical Plant level.

The **Chemical Plant Module** treats all production processes as "black-box-processes" with one input stream, one output stream and one waste stream. It combines the streams of all production processes to simulate the input, output and waste behavior of a chemical plant.

The **Chemical Industry Module** aggregates the chemical plants by their input, output and waste streams and sets up the connections to the environment of the Industrial Production System (e.g., market, waste management, biosphere as in Fedra et al., 1987).

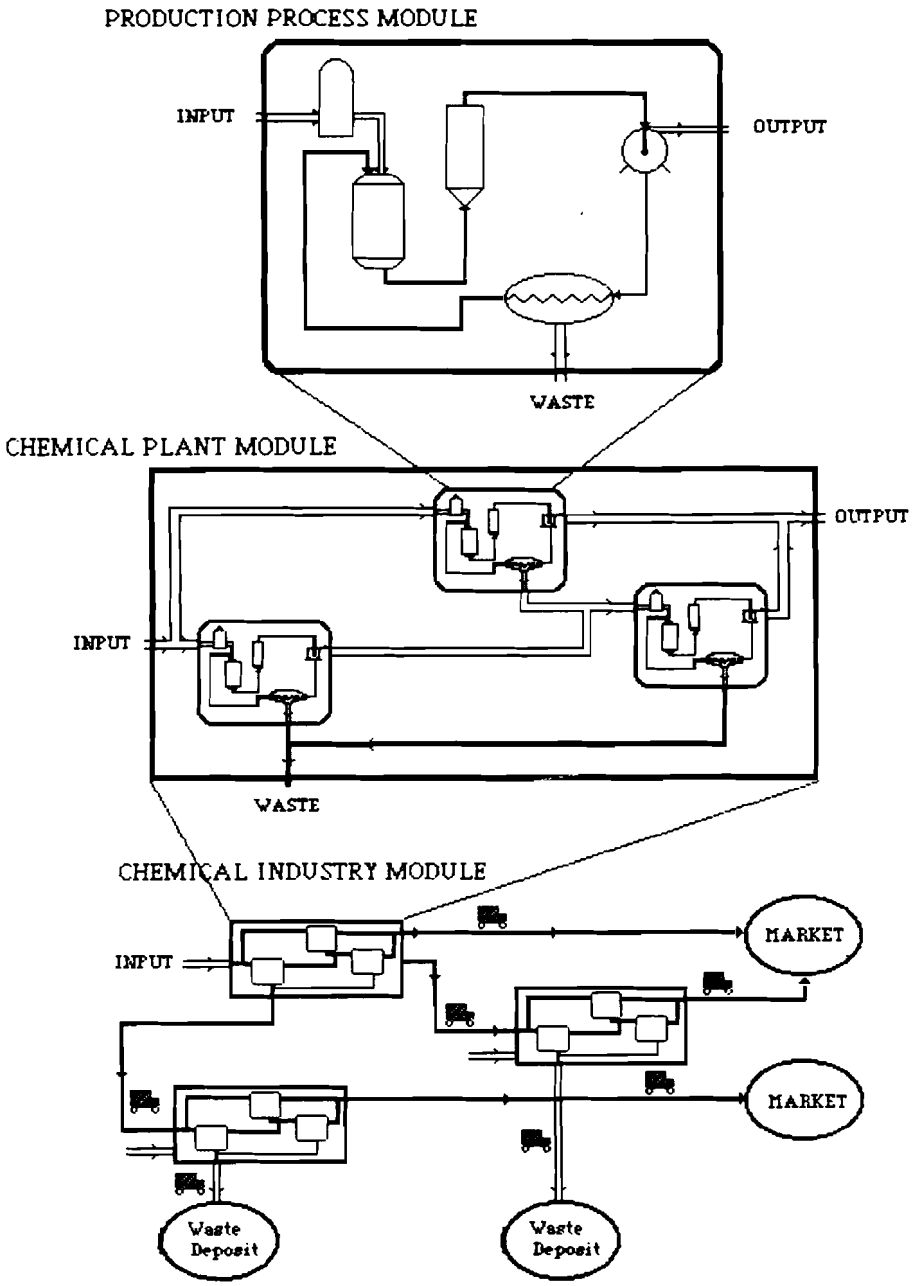

Figure 2: Industrial Production System modules (after Fedra 1985, p.18)

At the current stage of the implementation of IRIMS the Chemical Industry module integrates the Chemical Plant module so that so far there are only two levels of disaggregation (Zanelli et al., 1984). A more disaggregated solution is under development.

The Chemical Industry module is discussed at length in Fedra (1987). The Production Process module is described below.

3. THE PRODUCTION PROCESS MODULE

3.1. Model Description

The Production Process module simulates product-oriented production processes. Each production process consists of *Unit Activities* (Unit Processes [Herrick et al., 1979] and Chemical Processes) and *Units* (Zanelli et al., 1984), where the Unit Activity takes place.

The combination of a Unit Activity and a Unit, which is necessary if the process is to occur, is called an *Operating Unit* (see Figure 3).

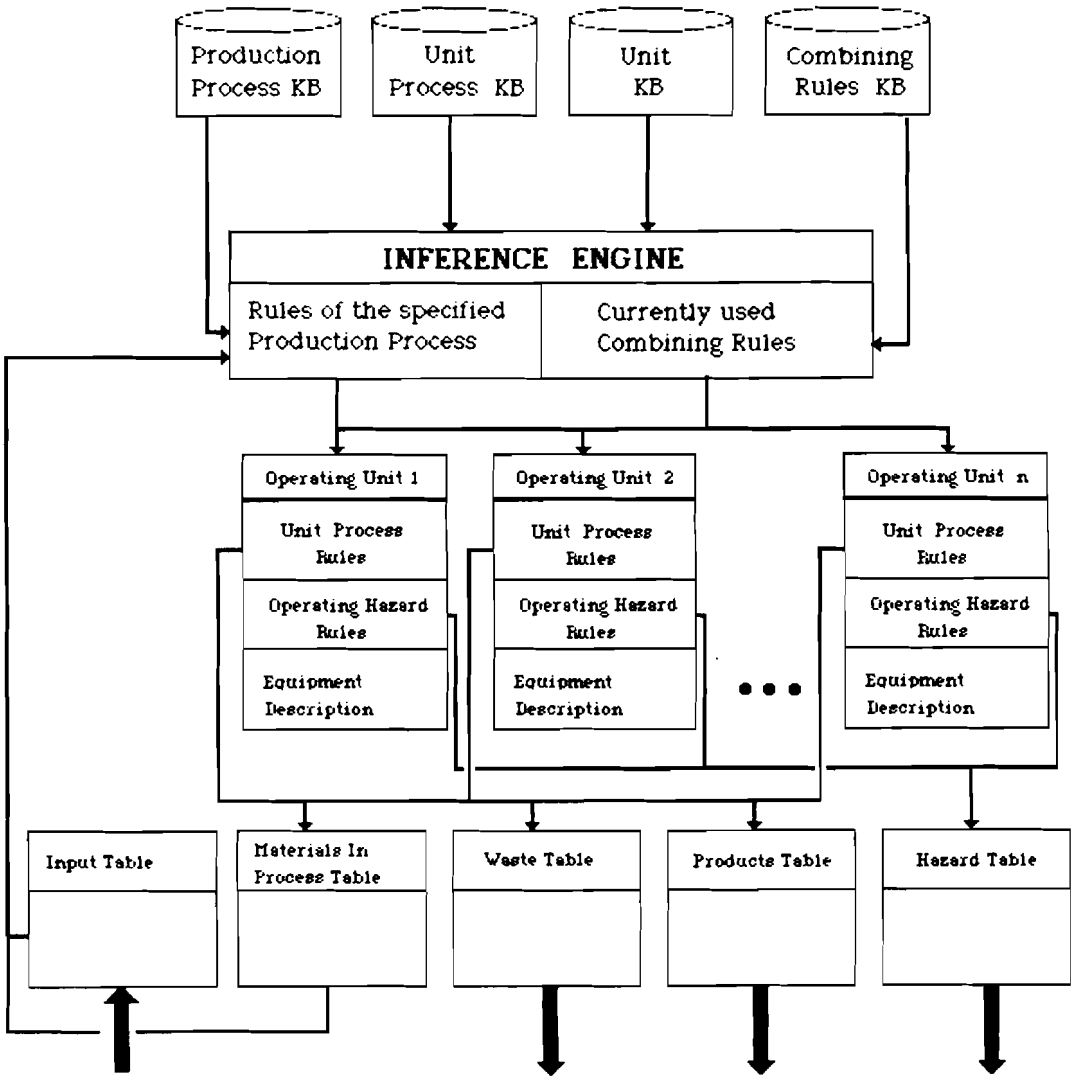

Figure 3: The Production Process Module

In order to satisfy a specific production goal the Operating Units are linked by their input/output streams (direct as well as indirect recursive). Certain input streams are connected to external input streams (i.e., input from outside the Production Process Module) and certain output streams are the waste and product output streams of the whole Production Process Module.

The production process starts as soon as input material is provided to the Operating Units which are connected to the external input streams. These Operating Units perform their Unit Activities depending on the input materials, the operating conditions of the Unit and the constituents of the Unit, and by this produce some output material, which are sent (via the linked input/output streams) to other Operating Units, which in turn are activated on receiving input material. They too perform their Unit Activities and produce output, thus activating other Operating Units and so on. After the production and the release of output material, an Operating Unit is deactivated until it receives new input material. This sequence of activation and deactivation of Operating Units by materials terminates when there is no more input material for any of the Operating Units, e.g., all external input has been transformed to the desired products, by-products and waste.

During the simulation of the production process the *Operating Hazards* of the Units and the hazards caused by the materials used and produced (e.g., input materials, interim products, end products, waste materials), the *Material Hazards,* are recorded and dynamically updated in the form of Hazard Ratings (NFPA, 1977; AICE, 1973; Sax, 1975).

3.2. Description of the Implementation

The Production Process Simulator is implemented in CommonLisp on a SUN-3 graphics workstation as a dynamic feature of a database on chemical production technologies, within the framework of the integrated software system mentioned in section one.

The simulation is performed and controlled by forward chaining rules which operate on the simulation objects (the *Operating Units*) and are executed using a dedicated self-developed inference engine (Winston, 1977; Cohen and Feigenbaum, 1982; Winston and Horn, 1984). It is designed to enable non-expert users to get an idea of how certain products are produced and where the potential hazards lie during the production process.

3.2.1. The Components of the Symbolic Simulator

The symbolic simulator consists of *knowledge bases*, which contain chemical expertise about Unit processes and Units as well as rules of control, and *dynamic information tables* (with dynamically instantiated simulation objects [Operating Units]) and an *inference engine* which applies the rules of the knowledge bases and thus performs the simulation.

a) Knowledge Bases

Unit Process KB:

All the rules required to simulate the Unit Processes of the production technologies listed in the production technologies database are stored here.

For the sample production process (chlorination of phenol) these rules comprise the following Unit Processes: *Halogenation* (8 rules), *Distillation* (2 rules), *Condensation* (3 rules), *Absorption* (1 rule), *Reflux* (2 rules) and *Pumping* (1 rule) (see Appendix C).

The rules represent input/output transformations under certain operating conditions of the simulation objects they are assigned to during a simulation run.

One of the *Halogenation rules* is listed below, as an example:

```
((IF ((in (Input (this Operating-Unit)) 'phenol)
      (in (Input (this Operating-Unit)) 'chlorine)
      (greater (temperature (this Operating-Unit)) '70C)
      (same (pressure (this Operating-Unit)) '1.3atm)))
((THEN ((material o__chlorophenol
          (From (this Operating-Unit))
          (To (port 0 (this Operating-Unit)))
          (Phase 'gas)
          (Hazard__ratings '(L L H H))
          (Status 'active))
      (material p__chlorophenol
          (From (this Operating-Unit))
          (To (port 0 (this Operating-Unit)))
          (Phase 'gas)
          (Hazard__ratings '(L L H H))
          (Status 'active))
      (material HCl
          (From (this Operating-Unit))
          (To (port 1 (this Operating-Unit)))
          (Phase 'gas)
          (Hazard__ratings '(L L H H))
          (Status 'active))
      (Status* '(phenol chlorine) 'inactive))))
```

In natural language:
If phenol and chlorine are supplied to the current Operating Unit and the temperature of the current Operating Unit is higher than 70°C and the pressure is exactly 1.3 atm
then o__chlorophenol, p__chlorophenol and hydrogen chloride are produced in the current Operating Unit and sent to the Operating Unit which is connected to the current Operating Unit via the pipe starting at port 0 of the current Operating Unit and the status of phenol and chlorine is set to 'inactive' (i.e., they are marked as used up by this Unit Activity).

Unit KB:

This KB contains the information about the Units (i.e., the hardware) required to run all chemical production processes contained in the technology database of the framework system. The information covers the following properties: *Type of Unit, Equipment Description* (i.e., real hardware), *Operating Conditions* (e.g., temperature, pressure,...), *Unit Activity* (i.e., the set of assigned production process rules – dynamically set by the appropriate Combining Rules) and *Operating Hazard Measurement* (i.e., the set of hazard rating rules for the dynamic evaluation of the hardware risk of the Operating Unit).

The Units required by the sample production process are: Stirred Batch Reactor, Batch Vacuum Distillation Column, Condenser, Codensing Trap, Absorption Tower, Recycle Pump, Reflux Drum and Flow Meter (see Appendix C).

The Unit description of a *Stirred Batch Reactor* is as follows:

```
(Stirred_batch_reactor
  (Description:
   Type Stirred_batch_reactor
   Equipment_description
            (Features:
             ports (connected:)
             graphic_representation 'Reactor)
   Operating_conditions
            (Features:
             temperature 150C
             pressure 1.3atm)
   Unit_process nil
   Operating_hazard_measurement 'Reactor_Hazard_Rules))
```

In natural language:
The unit *stirred batch reactor* is described by its type which is 'Stirred_batch_reactor', its equipment, i.e., its ports (connections to other operating units via pipes, dynamically assigned by the production process rules) and its graphical representation on the screen, its operating conditions, i.e., the temperature and the pressure (currently supplied with default values), its unit processes (dynamically assigned by the combining rules) and the operating hazard measurement rules (in the current state of development only a placeholder for a rule package to be developed later on).

Combining Rules KB:

Here all rules for combining Unit Processes with adequate Units – depending on various preference possibilities (for example: economic optimum, safety optimum, financial restrictions etc.) – are included. These rules cause the selection of the appropriate Units (see Unit KB) for the Unit Processes that are used to perform the desired chemical production process and combine the Units with Unit Processes to create Operating Units (i.e., Instances of Unit Descriptions referred to by unique names with Unit Process rules assigned to the Unit_Process descriptor).

In the case of the sample production process only seven combining rules are needed because the assignment of Units to Unit Processes used for chlorination of phenol is unique (see Appendix C).

Therefore the combining rules are as simple as the following one (although they may become extremely complicated in other cases):

```
((IF ((in (fact Unit-Processes) 'Halogenation)))
((THEN ((create Operating-Unit Stirred_batch_reactor Halogenation)
     (remove-fact Unit-Processes Halogenation))))
```

In natural language:
If one of the unit processes to be included in the simulation of the selected production process is 'Halogenation'
then a new operating unit will be created by combining the unit description of 'Stirred_batch_reactor' with the rule package of the production process 'Halogenation' and the fact that the production process 'Halogenation' is to be included is removed.

Production Process KB:

In this KB the process-specific rules of each implemented Production Process are located. These rules

- select the Unit Processes used for the simulation of the desired production process;
- initiate the creation of the Operating Units by selecting the Combining Rules;
- set up the linkage of the Operating Units by connecting the ports of the Units (see Unit KB, Equipment_description);
- distribute the external input materials using a set of input rules;
- provide the default operating conditions for the Operating Units which enable a standard run of the production process;
- activate Operating Units to which input material (external or from other Operating Units) has been sent, i.e., apply the Production Process rules of the Operating Units.

For the sample production process 7 Production Process rules are implemented (3 to select the Unit processes, 3 to interconnect the Operating Units and one to activate the Operating Units to which input material has been sent) (see Appendix C).

As an example, one of the rules which select the required Unit processes depending on the desired products is given below:

```
((IF ((same (fact Process) 'Chlorination_of_Phenol)
     (Or (same (fact Product) 'trichlorophenol_2/4/6)
         (same (fact Product) 'tetrachlorophenol_2/3/4/6))
     (empty (fact Unit-Processes))))
((THEN ((fact Unit-Processes
          ((Halogenation 2)
           (Distillation 2)
           (Condensing 1)
           (Codensation 2)
           (Absorption 1)
           (Reflux 2)
           (Pumping 1))))))
```

In natural language:

If the production process to be simulated is 'Chlorination_of_Phenol' and the desired product is either trichlorophenol_2/4/6 or tetrachlorophenol_2/3/4/6 and the unit processes which are to be used for the simulation are not already selected

then two halogenation processes, two distillation processes, one condensing process, two condensation processes, one absorption process, two reflux processes, and one pumping process are defined as required for the simulation of the production process.

Metarules:

To control the sequence of rule applications and to provide conflict resolution Metarules are used. In the case of the sample production process 4 Metarules have been implemented which may also be used for other production process simulations (see Appendix C). They schedule the application sequence of rule packages in four situations: start of the simulation, external material input, simulation of the production process and end of the simulation. An example of the Metarule for starting the simulation is shown below:

```
((IF ((not-empty Input-Table)
      (empty MIPT)))
((THEN (apply* Production_Process-Rules)
       (apply* Combining-Rules)
       (apply* Production_Process-Rules)
       (apply* Input-Rules))))
```

In natural language:
If there is input material to be supplied to the production process and there is no interim product produced so far,
then apply the following rule packages consecutively to the inference engine: Production Process rules, Combining rules, Production Process rules (again), Input rules.

b) Dynamic Information Tables

During a simulation run the simulation objects created and the deduced facts are represented by the following Dynamic Information Tables:

Operating Units:

This table includes all Unit Process/Unit combinations set up by the Production Process Rules via the Combining Rules. The Operating Units represent the simulation objects. They are the instances of the Unit KB (hardware) descriptions, i.e., there are values assigned to their slots and to each Operating Unit.

Input Table:

In this table the descriptions of the input materials which are required to run the chemical production process are collated. They are factors external to the simulation and are automatically provided by the simulator to enable a standard simulation to be run.

Materials In Process Table (MIPT):

This table holds all materials (interim products) that are or were sent from one Operating Unit to another. The material descriptions consist of the name of the material (a unique name for each material; e.g., o-chlorophenol-1), the information from which to which Operating Unit the material is (or has been) sent, the state of the material (gas, liquid or solid), the hazard ratings (high [H], medium [M] or low [L]) for relative pressure, flammability, toxicity and chemical burn risk for the material and the status of the material (active, inactive) which indicates if the material – at the current state of the simulation – is (*active*) or has been (*inactive*) present in the chemical production process.

A sample material representation (as created by the Unit Activity Rules assigned to Stirred_batch_reactor-1) is as follows (also see the description of the Unit Activity KB above):

```
(o_chlorophenol-1
        (From (this Operating-Unit))
        (To (port 0 (this Operating-Unit)))
        (Phase 'gas)
        (Hazard_ratings '(L L H H)
        (Status 'active))
```

In natural language:
The currently viewed interim substance o_chlorophenol is referred to by its unique name 'o_chlorophenol-1', has been produced in the currently active operating unit, is to be transferred to the operating unit which is connected via port 0 of the current active operating unit, is in a gaseous state, has low relative pressure, low flammability, high toxicity and high chemical burn risk, and its status is active.

Waste Table:

The information on all the wastes produced during the production process is to be found here. The descriptors and the internal representation are the same as for the materials in process and the products.

Products Table:

This table contains all the end products of the production process. The descriptors and the internal representation are the same as for the materials in process and the wastes.

c) The Inference Engine

To be able to apply the rules stored in the knowledge bases described above, an *inference engine* has been developed. A – forward chaining – inference engine is a program (in our case a CommonLisp program) which evaluates premises of rules and generates the consequences if all the premises have been fulfilled.

The simulation is started by applying the Metarules, which control the scheduling of the rule packages by forwarding the name of the package to be applied to the *apply*-function* of the inference engine, which then loads the referred rules from their knowledge base – if they are not already present - and hands rule after rule of the package over to the *rule-monitor* of the inference engine. The rule monitor evaluates the premises of the rule. If all the premises have been fulfilled the results activated by the rule are obtained (in other words the rule *is fired*).

A rule-package is applied as long as at least one rule of the package has been fired. When no more matching rules exist in a package the Metarules regain control and forward the next rule package to the inference engine. They are themselves applied by the same mechanism with the exception that the Metarule package is only once forwarded to the inference engine. When no more matching Metarule can be found no more inferences can be deduced and therefore the simulation is finished. The Lisp code of the central part of the inference engine, the *rule-monitor* is shown below (in a simplified version):

```
(defun rule-monitor (rule)
    (let ((premise (get-premise rule))
          (consequence (get-consequence rule)))
     (cond ((member 'f (execute premise)) nil))
           (t (progn (execute consequence) 'Yes)))))

(defun execute (sequence)
    (mapcar 'eval sequence))
```

3.2.2. Performance

a) Starting up

Using one of the entry options of the technology database of the framework system (see introduction), the user selects a specific production process. Then he chooses the desired product from the products list of the database information displayed and activates the inference engine, which reads the rules for the specified production process from the Production Process KB. Then the inference engine applies the rules for the selection of the unit processes. This in effect ensures that the rules of the selected unit processes are read from the Unit Process KB.

After this the rules for building the operating units are applied, which read and apply the rules for combining unit processes with adequate units from the Combining Rules KB, and create the operating units for the specified production process by adding the descriptions of the constituents from the Unit KB to the corresponding unit process rules.

Then the inference engine applies the rules for the linkage of the operating units, which connect the operating units by initializing the destination variables of the output rules of each operating unit.

After the set up is established the input distribution rules are activated, which read the input material descriptions from the Input Table, give each of them a hazard rating (if not provided in the Input Table description), the description of its source (e.g., "external input") and its destination (an operating unit name), set the status to "active" (i.e., currently in the production process) and write this extended material description onto the Materials In Process Table (MIPT).

b) The Iteration Sequence

The inference engine reads the descriptions of the "active" materials from the MIPT, sets their status descriptor to "inactive" (i.e., had been in the production process) and activates (i.e., applies the rules of) the operating units, which are mentioned in the destination descriptions of the former "active" materials. Before the unit process rules of the activated operating units are applied, the rules for setting the operating conditions of the activated operating units are brought into effect.

When the operating conditions are set – in a future version of the symbolic simulator with the active interaction of the user as well – the following occurs for each activated operating unit:

The unit process rules are applied and input material descriptions are transformed to output material descriptions, depending on the operating conditions of the operating unit. The output material descriptions are then written into the MIPT (with the status descriptor "active"), to the Waste Table or to the Products Table, depending on the values of the destination variables of the output rules of the operating unit. Their material hazard rating also is written to the Hazard Table, together with the operating hazard estimation of the operating unit which produced the specific material. After this the operating unit is deactivated.

The new material descriptions in the MIPT are read by the inference engine and the sequence described above is repeated with new material descriptions, new active operating units, and so on.

When no further "active" material descriptions can be found on the MIPT by the inference engine, then there are no more operating units to be activated and the simulation of the production process ends.

4. THE USER INTERFACE

Any comprehensive system for the management of industrial risk, and hazardous substances in particular, must provide for the simultaneous consideration of technological, economic, environmental and socio-political factors.

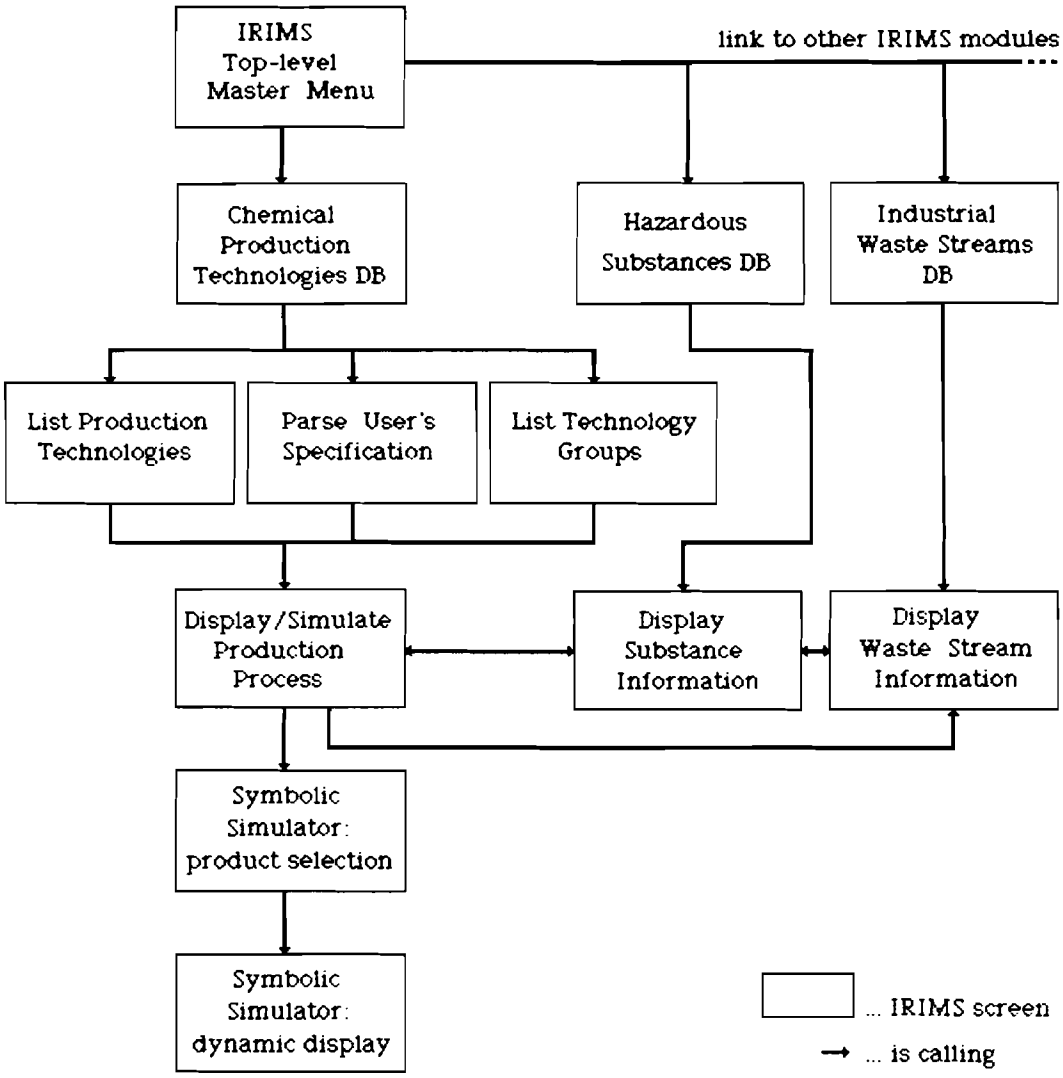

Figure 4: Integration of the Symbolic Simulator Module in IRIMS

The background information required for strategic planning and policy making is characterized by a broad range of disciplines and is subject to a variable degree of resolution and uncertainty. The management and decision-making process therefore requires a strong element of human expertise and judgement in addition to the more formal, scientifically-based, analytical techniques. Methods of applied systems analysis and risk assessment implemented using modern information processing technology with user friendly interfacing, can now support such a comprehensive, interdisciplinary approach to the management of industrial risk. This approach can provide a powerful interactive tool for planners and policy makers, because it makes access to a large number of relevant data-

bases and problem simulation modules easy.

In this section the implementation of the symbolic simulator within the framework of an enhanced prototype demonstration version of the IRIMS, release 3.0 1986-12 (Fedra, 1986) is described. The overall structure of the system and design guidelines have been described in Fedra (1985, 1986), Fedra and Otway (1986), Zhao et al. (1985), Fedra et al. (1986).

The symbolic simulator module is embedded in the framework of IRIMS and is designed to be interconnected with several other simulation/optimization modules and databases of IRIMS to provide information about the feedstocks needed, the final products, the interim products and the wastes of the chemical production process simulated and to allow for an *a posteriori* hazard evaluation of the whole process as well as for each hardware unit and each substance involved. So far it has been implemented as a dynamic feature of the production technology database, which is accessible from the top-level of the menu as well as from a number of other modules (Figure 4).

On the following pages the components of IRIMS which are related to the symbolic simulator and the way it functions are described in a screen- or display-oriented fashion.

1. REFERENCES AND SELECTED BIBLIOGRAPHY

AICE (1973) Dow's Safety and Loss Prevention Guide, Hazard Classification and Protection. American Institute of Chemical Engineers, New York.

Barr, A., Feigenbaum, E. A. (1982) The Handbook of Artificial Intelligence, Volume I & II, Pitman, London.

Buchanan, B.G., Shortliffe, E.H. (1984) Rule-Based Expert Systems. The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison Wesley. Reading , Massachusetts. 748p.

Charniak, E., Riesback, Ch.K., McDermott, D.V. (1980) Artificial Intelligence Programming. Lawrence Erlbaum Associates Publishers. Hillsdale, New Jersey. 323p.

Christen, H.R. (1973) Grundlagen der allgemeinen und anorganischen Chemie. Fundamentals of General and Inorganic Chemistry. Sauerländer-Salle, Frankfurt am Main (In German).

Christen, H.R. (1970) Grundlagen der organischen Chemie. Fundamentals of Organic Chemistry. Sauerländer-Diesterweg-Salle, Frankfurt am Main (In German).

Cohen, P.R. and Feigenbaum, E.A. (1982) The Handbook of Artificial Intelligence. Vol 3. Pitman Books Ltd., Los Altos, California. pp.99-100.

Davis, R., Lenat, D.B. (1982) Knowledge-Based Systems in Artificial Intelligence. McGraw-Hill. New York. 490p.

Fedra, K. (1986) Ispra Risk Management Support (IRIMS) System: Modifications for the Dutch Chlorine Study and System Enhancements. Interim Report, Study Contract No. 2942-86-05 ED ISP A. August 86. Submitted to The European Atomic Energy Community, JRC, Ispra Establishment, Ispra, Italy. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 56p.

Fedra, K. (1986) Ispra Risk Management Support (IRIMS) System: System's Documentation and Cross-reference Manual. Appendix to the Interim Report. Study Contract No.2942-86-05 ED ISP A. August 86. Submitted to the European Atomic Energy Community, JRC, Ispra Establishment, Ispra, Italy. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 204p.

Fedra, K. (1986) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part II. A Demonstration Prototype System. CP-86-10, International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 98p.

Fedra, K. (1985) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part I. Structure and Design. CP-85-18. International Institute for Applied Systems Analysis. A-2361 Laxenburg, Austria. 61p.

Fedra, K., Karhu, M., Rys, T., Skocz, M., Zebrowski, M. and Ziembla, W. (1987) Model-based Decision Support for Industry–Environment Interactions. A Pesticide Industry Example. CP-87-xx. International Institute for Applied Systems Analysis. A-2361 Laxenburg, Austria. Forthcoming.

Fedra, K., Weigkricht, E., and Winkelbauer, L. (1987) A Hybrid Approach to Information and Decision Support Systems: Hazardous Substances and Industrial Risk Management. Proceedings of the First Conference on Economics and Artificial Intelligence, Aix-en-Provence, 2-4 September, 1986. IFAC Publications, Pergamon. pp.169-175.

Fedra, K., Paruccini, M., and Otway, H. (1986) Model-based Decision Support and Risk Management. Proceedings of the International Research Symposium on Assessing and Controlling the Risks: Society Facing Major Hazards. Chantilly, France, January 20-22.

Fedra, K. and Otway, H. (1986) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part III: Decision Support and Expert Systems: Uses and Users. CP-86-14, International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 44p.

Fedra, K. and Loucks, D.P. (1985) Interactive Computer Technology for Planning and Policy Modeling. Water Resources Research, 21/2, pp. 114-122.

Forsyth, R. [ed.] (1984) Expert Systems. Principles and Case Studies. Chapman and Hall, London. 231p.

Goldfarb, A.S., Goldgraben, G.R., Herrick, E.C., Quellette, R.P. and Cheremisinoff, P.N. (1981) Organic Chemicals Manufacturing Hazards. Ann Arbor Science, Ann Arbor, MI. 430p.

Grauer, M. and Fedra, K. (1986) Intelligent Decision Support for Technology Assessment: The Chemical Process Industry. Paper presented at the VIIth International Conference on Multiple Criteria Decision Making. 18-22 August, 1986, Kyoto, Japan.

Hart, A. (1986) Knowledge Acquisition for Expert Systems. Kogan Page Ltd. London. 180p.

Hayes-Roth, F., Waterman, D.A. (1983) Building Expert Systems. Addison Wesley. Reading, Massachusetts. 444p.

Herbert, M., Williams, G. (1986) An Examination of Qualitative Plant Modelling as a Basis for Knowledge-Based Operator Aids in Nuclear Power Stations. In: Expert Systems and Optimization in Process Control. Mamdani, A., Efstathiou, J. [eds.]. Gower Technical Press in association with Unicom Seminars Ltd. Aldershot, UK. pp. 184-205.

Herrick, E.C., King, J.A., Quellette, R.P., and Cheremisinoff, P.N. (1979) Unit Process Guide to Organic Chemical Industries. Unit Processes Series, Organic Chemical Industries, Vol 1. Ann Arbor Science, Michigan.

Howard, J.C. (1980) Practical Applications of Symbolic Computation. IPC Science and Technology Press Ltd., Surrey, England. 394p.

Kates, R.W. (1978) Risk Assessment of Environmental Hazard. SCOPE Report 8. J. Wiley, Chichester.

Kuipers, B. (1985) The Limits of Qualitative Simulation. In: Proceedings of the Ninth International Joint Conference on Artificial Intelligence IJCAI-85. 18-23 August 1985, Los Angeles, California. Volume I. pp. 128-136.

Lees, F.P. (1980) Loss Prevention in the Process Industries. Vols 1 & 2. Butterworths, London.

Lounamaa, P., Tse, E. (1986a) The Simulation and Expert Environment. In: Coupling Symbolic and Numerical Computing in Expert Systems. Kowalik, J.S. [ed.]. Elsevier Science Publishers B.V. North Holland.

Lounamaa, P., Tse, E. (1986b) Lisp Based Simulation Modeling. In: Preprints of the International Conference on Economics and Artificial Intelligence, 2-4 September 1986, Aix-en-Provence. pp. 197-198.

McCabe, W.L., and Smith, J.C. (1956) Unit Operations of Chemical Engineering, McGraw-Hill, New York.

Novak, G. (1982) GLISP User's Manual, HPP 82-1. Department of Computer Science. Stanford University.

NFPA (1977) Fire Hazard Properties of Flammable Liquids, Gases, and Volatile Solids. NFPA 325M. National Fire Protection Association.

Peckham, R., Williams, J., Fedra, K., Otway, H. (1986) A Decision Support System for Risk Management. Paper presented at the conference on Multi-Attribute Decision Making via OR-Based Expert Systems, University of Passau, FRG, April 20-25.

Powers, C.J. and Tompkins Jr., F.C. (1974) Fault-Tree Synthesis for Chemical Processes, American Institute of Chemical Engineers J. 20(2):376-387, New York.

Sax, N.I. (1975) Dangerous Properties of Industrial Materials, Van Nostrand Reinhold Company, New York.

Seinfeld, J.H., and Lapidus, L. (1974) Mathematical Methods in Chemical Engineering. Volume 3: Process Modeling, Estimation, and Identification, Prentice-Hall Inc., New Jersey.

Shreeve, R.W. (1956) Chemical Process Industries, McGraw-Hill Kogakusha Ltd., Tokyo.

Sol, H.G. [ed] (1983) Processes and Tools for Decision Support. Proceedings of the Joint IFIP WG8.3/IIASA Working Conference on Processes and Tools for Decision Support, 19-21 July 1982, Laxenburg, Austria, North-Holland, Amsterdam.

Stolfo, S.J. (1983) Knowledge Engineering: Theory and Practice. In: Proceedings, Trends and Applications 1983. Automating Intelligent Behavior. Applications and Frontiers. 97-104, IEEE 83CH1887-9.

Technica (1984) The SAFETI Package. Computer-Based System for Risk Analysis of Process Plant. Vol.I-IV and Appendices I-IV. Technica Ltd., Tavistock Sq., London.

Vauck, R.A., and Müller, H.A. (1978) Grundoperationen chemischer Verfahrenstechnik. Unit Operations of Chemical Operations Technology. VEB Deutscher Verlag für Grundstoffindustrie, Leipzig. (In German).

Weigkricht, E. and Winkelbauer, L. (1987) Knowledge-based Systems: Overview and Selected Examples. WP-87-xx. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. Forthcoming.

Weiss, S.M., Kulikowski, C.A. (1984) A Practical Guide to Expert Systems. Rowman & Allanheld. Totowa, New Jersey. 174p.

Wilensky, R. (1984) LISPcraft. W.W. Norton & Company. New York. 385p.

Winston, P.H. (1977) Artificial Intelligence. Addison Wesley, Reading, Massachusetts. pp.135-136.

Winston, P.H., Horn, B.K.P. (1984) Lisp (Second Edition), Addison Wesley. Reading, Massachusetts. 434p.

Zanelli, S., Bello, G.C., and Frattini, B. (1984) Classification Scheme for Process Plants. Intermediate Progress Report, November 1984. Joint Research Center, Ispra.

Zhao, Ch., Winkelbauer, L., Fedra, K. (1985) Advanced Decision-oriented Software for the Management of Hazardous Substances. Part VI. The Interactive Decision-Support Module. CP-85-50, International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria. 39p.

APPENDIX A

DECOMPOSITION HIERARCHY OF THE INDUSTRIAL PRODUCTION SYSTEM

Classification Scheme for Process Plants – The Set Up
as in Zanelli et al. (1984)

1. Chemical Industry Level

2. Chemical Plant Level

3. System *

4. Unit

Level 1 (Industry) is defined as a set of industrial plants having in common the basic raw materials and/or the product range and/or the technology used. As an example, one can quote Petroleum Refining - common basic raw materials - , the Pesticide Industry - common products range - , the Electrolytic Industry - common technology used.

Level 2 (Plant) is defined as a set of systems characterized by a common production target, together with their relevant auxiliary and recovering facilities. The common production target may be accomplished by different arrangements of the same basic systems, according to the various licensing technologies.

Level 3 (System) is defined as a set of units univocally characterized by their primary function within the plant. The primary functions have been split under two major headings, i.e., physical elementary functions (PHEFs) and chemical elementary functions (CHEFs).

Level 4 (Unit) is defined as an aggregate of mechanical, electrical, instrumental components constituting a well-identified element within the system.

*Production Process Level (Fedra et al., 1987)

APPENDIX B

SAMPLE PRODUCTION PROCESS DESCRIPTION: CHLORINATION OF PHENOL (after Goldfarb et al., 1981)

The chlorination of phenol (Figure 5) has been selected as a representative example of a chemical production process and is used as a guiding example throughout the description of the symbolic simulator.

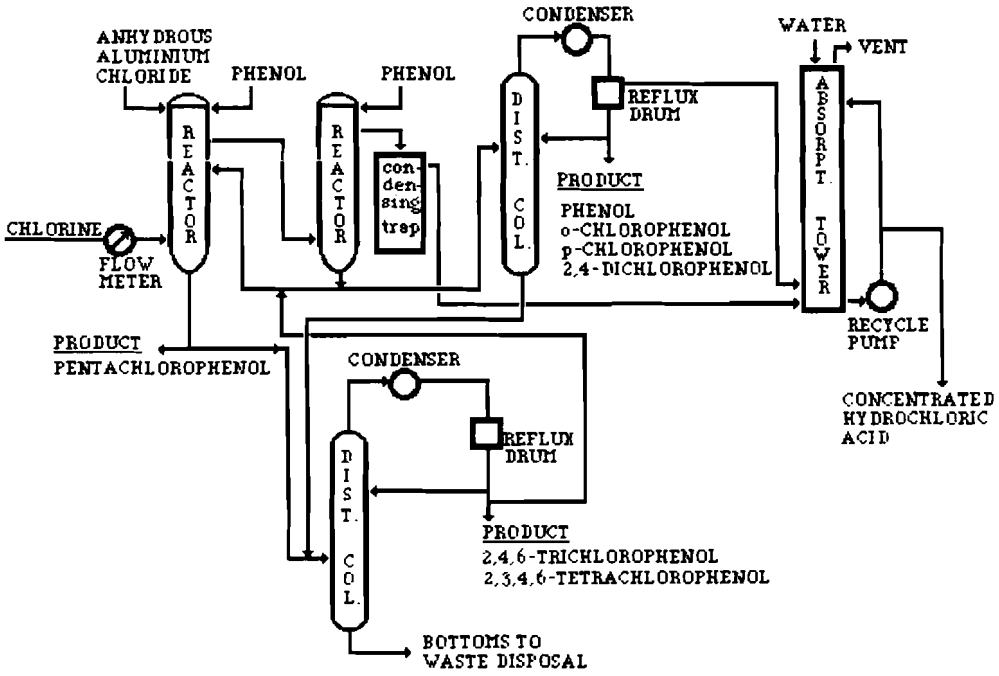

Figure 5: Production Process Description: Chlorination of Phenol after: Goldfarb et al., 1981

Products:
The chlorination of phenol proceeds stepwise so that six chlorophenols can be obtained: *o-chlorophenol, p-chlorophenol, 2,4-dichlorophenol, 2,4,6-trichlorophenol, 2,3,4,6-tetrachlorophenol* and *pentachlorophenol.*

Commercial Use:
2,4-dichloropenol is used as an intermediate in the manufacture of 2,4-dichlorophenoxyacetic acid (2,4,-D) and its derivatives, selective herbicides, and pentachlorophenol (PCP) which is used as a wood preservative on account of its fungicidal properties.

Process Description:
At the beginning phenol is charged into the primary reactor and into the secondary scrubber-reactor in which the temperature is maintained at 70-120°C (depending on the desired end products) and the pressure is about 1.3 atm. Then chlorine is added to the primary reactor and the offgas of the primary reactor (chlorine and hydrogen chloride) is passed to the secondary reactor where sufficient phenol is present to ensure complete reaction of the chlorine. The hydrogen chloride offgas from the secondary scrubber reactor is recovered by dissolving it in water in an absorption tower to produce concentrated hydrochloric acid suitable for commercial use. A small amount of anhydrous aluminum chloride is added as a catalyst to the primary reactor when chlorination has proceeded to the dichlorophenol stage.

For the production of monochlorophenols the temperature in the secondary scrubber-reactor is maintained at about 70°C. The o-chlorophenol and p-chlorophenol are separated in a batch vacuum distillation column. To produce 2,4-dichlorophenol the temperature in the secondary reactor is increased to 120°C. The 2,4-dichlorophenol is separated from the 2,6-dichlorophenol in a batch vacuum distillation column. The temperature of the phenol in the primary reactor ranges from 65 to 130°C until the melting point of the product reaches 95°C. If 2,4,6-trichlorophenol and 2,3,4,6-tetrachlorophenol are desired as products, the reactor contents are then sent to a batch vacuum distillation column for separation. If pentachlorophenol is the desired product, the chlorination in the primary reactor is continued and the reaction temperature is progressively increased to maintain a differential temperature of 10°C over the product melting point. When a product melting point of at least 174°C is reached chlorination is terminated. For technical-grade pentachlorophenol no further purification is required.

APPENDIX C

KNOWLEDGE BASES FOR THE SIMULATION OF THE PRODUCTION PROCESS: CHLORINATION OF PHENOL

U A K B - Unit Activity Knowledge Base:
contains the Rules for the chemical and unit processes (i.e., unit activities) of the production processes implemented so far.

```
(Unit_Activities
  (Halogenation
   Distillation
   Condensation
   Absorption
   Reflux
   Pumping))
```

Halogenation Rules

```
(Halogenation

((HA1 HA2 HA3 HA4 HA5 HA6 HA7 HA8)

 (Rules

  HA1
  (($IF ((in (Input (this Operating-Unit)) 'phenol)
      (in (Input (this Operating-Unit)) 'chlorine)
      (greater (temperature (this Operating-Unit)) 70)
      (same (pressure (this Operating-Unit)) '1.3atm)))
   ($THEN ((material o-chlorophenol
              (From (this Operating-Unit))
              (To (port 0 (this Operating-Unit)))
              (Phase 'gas)
              (Hazard_ratings '(L L H H))
              (Status 'active))
       (material p-chlorophenol
              (From (this Operating-Unit))
```

```
            (To (port 0 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H H))
            (Status 'active))
      (material HCl
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H H))
            (Status 'active))
      (Status* '(phenol chlorine) 'inactive))))


HA2
 (($IF ((in (Input (this Operating-Unit)) 'o-chlorophenol)
     (in (Input (this Operating-Unit)) 'p-chlorophenol)
     (in (Input (this Operating-Unit)) 'aluminum_chloride)
     (greater (temperature (this Operating-Unit)) 90)
     (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((material dichlorophenol_2/4
            (From (this Operating-Unit))
            (To (port 0 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H H))
            (Status 'active))
      (material dichlorophenol_2/6
            (From (this Operating-Unit))
            (To (port 0 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H H))
            (Status 'active))
      (Status* '(o-chlorophenol p-chlorophenol) 'inactive))))


HA3
 (($IF ((in (Input (this Operating-Unit))
         'dichlorophenol_2/4)
     (in (Input (this Operating-Unit))
        'dichlorophenol_2/6)
     (in (Input (this Operating-Unit)) 'aluminum_chloride)
     (greater (temperature (this Operating-Unit)) 105)
     (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((material trichlorophenol_2/4/6
            (From (this Operating-Unit))
            (To (port 0 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (Status* '(dichlorophenol_2/4 dichlorophenol_2/6) 'inactive))))


HA4
 (($IF ((in (Input (this Operating-Unit)) 'trichlorophenol_2/4/6)
     ($OR (same (fact Product) 'trichlorophenol_2/4/6)
```

```
        (same (fact Product) 'tetrachlorophenol_2/3/4/6))
     (in (Input (this Operating-Unit)) 'aluminum_chloride)
     (greater (temperature (this Operating-Unit)) 105)
     (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((material aluminum_chloride
              (From (this Operating-Unit))
              (To (port 3 (this Operating-Unit)))
              (Phase 'solid)
              (Hazard_ratings '(L L H M))
              (Status 'active))
      (material tetrachlorophenol_2/3/4/6
              (From (this Operating-Unit))
              (To (port 3 (this Operating-Unit)))
              (Phase 'liquid)
              (Hazard_ratings '(L L H M))
              (Status 'active))
      (material pentachlorophenol
              (From (this Operating-Unit))
              (To (port 3 (this Operating-Unit)))
              (Phase 'liquid)
              (Hazard_ratings '(L L H M))
              (Status 'active))
      (material tetrachlorophenol_2/3/4/6
              (From (this Operating-Unit))
              (To (port 0 (this Operating-Unit)))
              (Phase 'liquid)
              (Hazard_ratings '(L L H M))
              (Status 'active))
      (Status* 'trichlorophenol_2/4/6 'inactive))))


HA5
  (($IF ((in (Input (this Operating-Unit))
         'trichlorophenol_2/4/6)
     (in (Input (this Operating-Unit)) 'aluminum_chloride)
     (greater (temperature (this Operating-Unit)) 130)
     (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((material tetrachlorophenol_2/3/4/6
              (From (this Operating-Unit))
              (To (port 0 (this Operating-Unit)))
              (Phase 'liquid)
              (Hazard_ratings '(L L H M))
              (Status 'active))
      (Status* 'trichlorophenol_2/4/6 'inactive))))


HA6
  (($IF ((in (Input (this Operating-Unit))
         'tetrachlorophenol_2/3/4/6)
     (in (Input (this Operating-Unit)) 'aluminum_chloride)
     (greater (temperature (this Operating-Unit)) 130)
     (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((material pentachlorophenol
              (From (this Operating-Unit))
```

```
            (To (port 2 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (Status* '(tetrachlorophenol_2/3/4/6 aluminum_chloride)
            'inactive))))


HA7
 (($IF ((in (Input (this Operating-Unit)) 'phenol)
     (in (Input (this Operating-Unit)) 'HCl)
     (greater (temperature (this Operating-Unit)) 90)
     (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((Status* '(phenol HCl) 'inactive)
      (material phenol
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material o-chlorophenol
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material p-chlorophenol
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/4
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/6
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material HCl
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active)))))
```

```
HA8
 (($IF ((in (Input (this Operating-Unit)) 'phenol)
     (in (Input (this Operating-Unit)) 'o-chlorophenol)
     (in (Input (this Operating-Unit)) 'p-chlorophenol)
     (in (Input (this Operating-Unit)) 'dichlorophenol_2/4)
     (in (Input (this Operating-Unit)) 'dichlorophenol_2/6)
    (greater (temperature (this Operating-Unit)) 105)
    (same (pressure (this Operating-Unit)) '1.3atm)))
  ($THEN ((Status* '(phenol o-chlorophenol p-chlorophenol
              dichlorophenol_2/4 dichlorophenol_2/6)
            'inactive)
      (material phenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material o-chlorophenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material p-chlorophenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/4
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/6
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material trichlorophenol_2/4/6
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material HCl
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active)))))))
```

Distillation Rules

```
(Distillation

((DI1 DI2)

(Rules

 DI1
 (($IF ((in (Input (this Operating-Unit)) 'phenol)
     (in (Input (this Operating-Unit)) 'p-chlorophenol)
     (in (Input (this Operating-Unit)) 'o-chlorophenol)
     (in (Input (this Operating-Unit)) 'dichlorophenol_2/4)
     (in (Input (this Operating-Unit)) 'dichlorophenol_2/6)
     (in (Input (this Operating-Unit)) 'trichlorophenol_2/4/6)
     (in (Input (this Operating-Unit)) 'HCl)))
  ($THEN ((material phenol
              (From (this Operating-Unit))
              (To (port 1 (this Operating-Unit)))
              (Phase 'gas)
              (Hazard_ratings '(L H H M))
              (Status 'active))
       (material o-chlorophenol
              (From (this Operating-Unit))
              (To (port 1 (this Operating-Unit)))
              (Phase 'gas)
              (Hazard_ratings '(L H H M))
              (Status 'active))
       (material p-chlorophenol
              (From (this Operating-Unit))
              (To (port 1 (this Operating-Unit)))
              (Phase 'gas)
              (Hazard_ratings '(L H H M))
              (Status 'active))
       (material dichlorophenol_2/4
              (From (this Operating-Unit))
              (To (port 1 (this Operating-Unit)))
              (Phase 'gas)
              (Hazard_ratings '(L H H M))
              (Status 'active))
       (material HCl
              (From (this Operating-Unit))
              (To (port 1 (this Operating-Unit)))
              (Phase 'gas)
              (Hazard_ratings '(L H H M))
              (Status 'active))
       (material dichlorophenol_2/6
              (From (this Operating-Unit))
              (To (port 2 (this Operating-Unit)))
              (Phase 'liquid)
```

```
        (Hazard_ratings '(L H H M))
        (Status 'active))
    (material trichlorophenol_2/4/6
        (From (this Operating-Unit))
        (To (port 2 (this Operating-Unit)))
        (Phase 'liquid)
        (Hazard_ratings '(L H H M))
        (Status 'active))
    (Status* '(phenol o-chlorophenol p-chlorophenol
        dichlorophenol_2/4 dichlorophenol_2/6
        trichlorophenol_2/4/6 HCl)
      'inactive))))
```

```
DI2
(($IF ((in (Input (this Operating-Unit)) 'dichlorophenol_2/6)
    (in (Input (this Operating-Unit))
        'trichlorophenol_2/4/6)
    (in (Input (this Operating-Unit))
        'tetrachlorophenol_2/3/4/6)
    (in (Input (this Operating-Unit))
        'pentachlorophenol)
    (in (Input (this Operating-Unit))
        'aluminum_chloride)))
 ($THEN ((Status* '(dichlorophenol_2/6 trichlorophenol_2/4/6
        tetrachlorophenol_2/3/4/6 pentachlorophenol
        aluminum_chloride)
      'inactive)
    (material dichlorophenol_2/6
        (From (this Operating-Unit))
        (To (port 1 (this Operating-Unit)))
        (Phase 'gas)
        (Hazard_ratings '(L H H M))
        (Status 'active))
    (material trichlorophenol_2/4/6
        (From (this Operating-Unit))
        (To (port 1 (this Operating-Unit)))
        (Phase 'gas)
        (Hazard_ratings '(L H H M))
        (Status 'active))
    (material tetrachlorophenol_2/3/4/6
        (From (this Operating-Unit))
        (To (port 1 (this Operating-Unit)))
        (Phase 'gas)
        (Hazard_ratings '(L H H M))
        (Status 'active))
    (material polynuclear_polychlorinated_tars
        (From (this Operating-Unit))
        (To (port 2 (this Operating-Unit)))
        (Phase 'liquid)
        (Hazard_ratings '(L H H H))
        (Status 'active))
    (material aluminum_chloride
        (From (this Operating-Unit))
```

```
            (To (port 2 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L H H H))
            (Status 'active))
      (material pentachlorophenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L H H H))
            (Status 'active))))))))
```

Condensation Rules

```
(Condensation

((CD1 CD2 CD3)

 (Rules

  CD1
  (($IF ((in (Input (this Operating-Unit)) 'phenol)
      (in (Input (this Operating-Unit)) 'p-chlorophenol)
      (in (Input (this Operating-Unit)) 'o-chlorophenol)
      (in (Input (this Operating-Unit)) 'dichlorophenol_2/4)
      (in (Input (this Operating-Unit)) 'dichlorophenol_2/6)
      (in (Input (this Operating-Unit)) 'HCl)))
   ($THEN ((Status* '(phenol o-chlorophenol p-chlorophenol
              dichlorophenol_2/4 dichlorophenol_2/6 HCl)
            'inactive)
      (material phenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material o-chlorophenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material p-chlorophenol
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/4
```

```
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/6
            (From (this Operating-Unit))
            (To (port 2 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material HCl
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H H))
            (Status 'active)))))


CD2
(($IF ((in (Input (this Operating-Unit)) 'phenol)
     (in (Input (this Operating-Unit)) 'p-chlorophenol)
     (in (Input (this Operating-Unit)) 'o-chlorophenol)
     (in (Input (this Operating-Unit)) 'dichlorophenol_2/4)
     (in (Input (this Operating-Unit)) 'HCl)))
 ($THEN ((Status* '(phenol o-chlorophenol p-chlorophenol
              dichlorophenol_2/4  HCl)
           'inactive)
      (material phenol
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid_&_gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material o-chlorophenol
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid_&_gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material p-chlorophenol
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid_&_gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material dichlorophenol_2/4
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid_&_gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material HCl
            (From (this Operating-Unit))
```

```
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid_&_gas)
            (Hazard_ratings '(L L H M))
            (Status 'active)))))

CD3
(($IF ((in (Input (this Operating-Unit)) 'dichlorophenol_2/6)
    (in (Input (this Operating-Unit))
      'trichlorophenol_2/4/6)
    (in (Input (this Operating-Unit))
      'tetrachlorophenol_2/3/4/6)))
 ($THEN ((Status* '(dichlorophenol_2/6 trichlorophenol_2/4/6
            tetrachlorophenol_2/3/4/6)
          'inactive)
     (material dichlorophenol_2/6
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))
     (material trichlorophenol_2/4/6
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))
     (material tetrachlorophenol_2/3/4/6
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))))))))
```

Absorption Rules

```
(Absorption

((AB1)

 (Rules

  AB1
  (($IF ((in (Input (this Operating-Unit)) 'H2O)
      (in (Input (this Operating-Unit)) 'HCl)))
   ($THEN ((Status* 'HCl 'inactive)
       (material hydrogen_chloride
            (From (this Operating-Unit))
```

```
        (To (port 1 (this Operating-Unit)))
        (Phase 'gas)
        (Hazard_ratings '(L L L L))
        (Status 'active))
    (material concentrated_hydrochloric_acid
        (From (this Operating-Unit))
        (To (port 2 (this Operating-Unit)))
        (Phase 'liquid)
        (Hazard_ratings '(L L H H))
        (Status 'active))))))))
```

Reflux Rules

```
(Reflux

((RF1 RF2)

 (Rules

  RF1
  (($IF ((in (Input (this Operating-Unit)) 'phenol)
      (in (Input (this Operating-Unit)) 'p-chlorophenol)
      (in (Input (this Operating-Unit)) 'o-chlorophenol)
      (in (Input (this Operating-Unit)) 'dichlorophenol_2/4)
      (in (Input (this Operating-Unit)) 'HCl)))
   ($THEN ((Status* '(phenol o-chlorophenol p-chlorophenol
              dichlorophenol_2/4  HCl)
            'inactive)
      (material phenol
          (From (this Operating-Unit))
          (To (port 2 (this Operating-Unit)))
          (Phase 'liquid_&_gas)
          (Hazard_ratings '(L L H M))
          (Status 'active))
      (material o-chlorophenol
          (From (this Operating-Unit))
          (To (port 2 (this Operating-Unit)))
          (Phase 'liquid_&_gas)
          (Hazard_ratings '(L L H M))
          (Status 'active))
      (material p-chlorophenol
          (From (this Operating-Unit))
          (To (port 2 (this Operating-Unit)))
          (Phase 'liquid_&_gas)
          (Hazard_ratings '(L L H M))
          (Status 'active))
      (material dichlorophenol_2/4
          (From (this Operating-Unit))
```

```
            (To (port 2 (this Operating-Unit)))
            (Phase 'liquid_&_gas)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material HCl
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'gas)
            (Hazard_ratings '(L L H H))
            (Status 'active)))))


RF2
(($IF ((in (Input (this Operating-Unit)) 'dichlorophenol_2/6)
    (in (Input (this Operating-Unit))
    'trichlorophenol_2/4/6)
    (in (Input (this Operating-Unit))
      'tetrachlorophenol_2/3/4/6)))
 ($THEN ((Status* '(dichlorophenol_2/6 trichlorophenol_2/4/6
            tetrachlorophenol_2/3/4/6)
          'inactive)
      (material trichlorophenol_2/4/6
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))
      (material tetrachlorophenol_2/3/4/6
            (From (this Operating-Unit))
            (To (port 1 (this Operating-Unit)))
            (Phase 'liquid)
            (Hazard_ratings '(L L H M))
            (Status 'active))))))))
```

Pumping Rules

```
(Pumping

((PU1)

 (Rules


 PU1
 (($IF ((in (Input (this Operating-Unit))
        'concentrated_hydrochloric_acid)))
  ($THEN ((Status* 'concentrated_hydrochloric_acid 'inactive)
      (material concentrated_hydrochloric_acid
            (From (this Operating-Unit))
```

```
(To (port 1 (this Operating-Unit)))
(Phase 'liquid)
(Hazard_ratings '(L L H H))
(Status 'active))))))))
```

U N I T K B - Unit Knowledge Base:
contains the descriptions of the hardware units of the process plants (= frames for the operating units).

```
(Units
 (Stirred_batch_reactor
  Batch_vacuum_distillation_column
  Condenser
  Condensing_trap
  Absorption_tower
  Recycle_pump
  Reflux_drum
  Flow_meter))
```

Stirred Batch Reactor Description

```
(Stirred_batch_reactor
 (Description
  Type Stirred_batch_reactor
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Reactor)
  Operating_conditions
   (Features
    temperature 150
    pressure 1.3atm)
  Unit_activity nil
  Operating_hazard_measurement 'Reactor_Hazard_Rules))
```

Batch Vacuum Distillation Column Description

```
(Batch_vacuum_distillation_column
 (Description
  Type Batch_vacuum_distillation_column
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Column)
  Operating_conditions
   (Features
    temperature nil
    pressure nil)
  Unit_activity nil
  Operating_hazard_measurement 'Column_Hazard_Rules))
```

Condenser Description

```
(Condenser
 (Description
  Type Condenser
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Condenser)
  Operating_conditions
   (Features
    temperature nil
    pressure nil)
  Unit_activity nil
  Operating_hazard_measurement 'Condenser_Hazard_Rules))
```

Condensing Trap Description

```
(Condensing_trap
 (Description
  Type Condensing_trap
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Condensing_trap)
  Operating_conditions
   (Features
    temperature nil
    pressure nil)
  Unit_activity nil
  Operating_hazard_measurement 'Condenser_Hazard_Rules))
```

Absorption Tower Description

```
(Absorption_tower
 (Description
  Type Absorption_tower
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Tower)
  Operating_conditions
   (Features
    temperature nil
    pressure nil)
  Unit_activity nil
  Operating_hazard_measurement 'Absorption_Hazard_Rules))
```

Recycle Pump Description

```
(Recycle_pump
 (Description
  Type Recycle_pump
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Pump)
  Operating_conditions
   (Features
    temperature nil
    pressure nil)
  Unit_activity nil
  Operating_hazard_measurement 'Pumping_Hazard_Rules))
```

Reflux Drum Description

```
(Reflux_drum
 (Description
  Type Reflux_drum
  Equipment_description
   (Features
    ports (connected)
    graphic_representation 'Drum)
  Operating_conditions
   (Features
    temperature nil
    pressure nil)
  Unit_activity nil
  Operating_hazard_measurement 'Reflux_Hazard_Rules))
```

Flow Meter Description

```
(Flow__meter
 (Description
  Type Flow__meter
  Equipment__description
   (Features
    ports (connected)
    graphic__representation 'Flow__meter)
  Operating__conditions
   (Features
    temperature nil
    pressure nil)
  Unit__activity nil
  Operating__hazard__measurement nil))
```

P P K B - Production Process Knowledge Base:
contains the process specific rules for the different production processes (e.g., required unit activities, hardware units, connections to the Operating Units, graphics display, etc.).

```
(Rule-List (PP1 PP2 PP3 PP4 PP5 PP6 PP7))

(Rules

   (PP1
    (($IF ((same (fact Process) 'Chlorination_of_Phenol)
       (same (fact Product) 'pentachlorophenol)
       (empty (fact Unit-Activities))))
    ($THEN ((fact Unit-Activities ((Halogenation 1))))))

    PP2
    (($IF ((same (fact Process) 'Chlorination_of_Phenol)
       ($OR (same (fact Product) 'o-chlorophenol)
          (same (fact Product) 'p-chlorophenol)
          (same (fact Product) 'dichlorophenol_2/4))
       (empty (fact Unit-Activities))))
    ($THEN ((fact Unit-Activities
            ((Halogenation 2)
             (Distillation 1)
             (Condensing 1)
             (Condensation 1)
             (Absorption 1)
             (Reflux 1)
             (Pumping 1))))))

    PP3
    (($IF ((same (fact Process) 'Chlorination_of_Phenol)
       ($OR (same (fact Product) 'trichlorophenol_2/4/6)
          (same (fact Product) 'tetrachlorophenol_2/3/4/6))
       (empty (fact Unit-Activities))))
    ($THEN ((fact Unit-Activities
            ((Halogenation 2)
             (Distillation 2)
             (Condensing 1)
             (Condensation 2)
             (Absorption 1)
             (Reflux 2)
             (Pumping 1))))))
```

```
PP4
(($IF ((same (fact Process) 'Chlorination_of_Phenol)
    (not-empty (fact Operating-Units))
    ($OR (same (fact Product) 'o-chlorophenol)
       (same (fact Product) 'p-chlorophenol)
       (same (fact Product) 'dichlorophenol_2/4))
    (empty (fact linked))))
 ($THEN ((port 0 'Stirred_batch_reactor-1
          'Stirred_batch_reactor-1)
      (port 1 'Stirred_batch_reactor-1
          'Stirred_batch_reactor-2)
      (port 2 'Stirred_batch_reactor-1
          'PRODUCT)
      (port 1 'Stirred_batch_reactor-2 'Condensing_trap-1)
      (port 2 'Stirred_batch_reactor-2
          'Batch_vacuum_distillation_column-1)
      (port 1 'Condensing_trap-1 'Absorption_tower-1)
      (port 2 'Condensing_trap-1 'Stirred_batch_reactor-2)
      (port 1 'Batch_vacuum_distillation_column-1
          'Condenser-1)
      (port 2 'Batch_vacuum_distillation_column-1
          'WASTE)
      (port 1 'Condenser-1 'Reflux_drum-1)
      (port 1 'Reflux_drum-1 'Absorption_tower-1)
      (port 2 'Reflux_drum-1 'PRODUCT)
      (port 1 'Absorption_tower-1 'WASTE)
      (port 2 'Absorption_tower-1 'Recycle_pump-1)
      (port 1 'Recycle_pump-1 'WASTE)
      (fact linked Yes))))


PP5
(($IF ((same (fact Process) 'Chlorination_of_Phenol)
    (not-empty (fact Operating-Units))
    ($OR (same (fact Product) 'trichlorophenol_2/4/6)
       (same (fact Product) 'tetrachlorophenol_2/3/4/6))
    (empty (fact linked))))
 ($THEN ((port 0 'Stirred_batch_reactor-1
          'Stirred_batch_reactor-1)
      (port 1 'Stirred_batch_reactor-1
          'Stirred_batch_reactor-2)
      (port 2 'Stirred_batch_reactor-1
          'PRODUCT)
      (port 3 'Stirred_batch_reactor-1
          'Batch_vacuum_distillation_column-2)
      (port 1 'Stirred_batch_reactor-2 'Condensing_trap-1)
      (port 2 'Stirred_batch_reactor-2
          'Batch_vacuum_distillation_column-1)
      (port 1 'Condensing_trap-1 'Absorption_tower-1)
      (port 2 'Condensing_trap-1 'Stirred_batch_reactor-2)
      (port 1 'Batch_vacuum_distillation_column-1
          'Condenser-1)
      (port 2 'Batch_vacuum_distillation_column-1
          'Batch_vacuum_distillation_column-2)
```

```
     (port 1 'Condenser-1 'Reflux_drum-1)
     (port 1 'Reflux_drum-1 'Absorption_tower-1)
     (port 2 'Reflux_drum-1 'PRODUCT)
     (port 1 'Absorption_tower-1 'WASTE)
     (port 2 'Absorption_tower-1 'Recycle_pump-1)
     (port 1 'Recycle_pump-1 'WASTE)
     (port 1 'Batch_vacuum_distillation_column-2
          'Condenser-2)
     (port 2 'Batch_vacuum_distillation_column-2 'WASTE)
     (port 1 'Condenser-2 'Reflux_drum-2)
     (port 1 'Reflux_drum-2 'PRODUCT)
     (fact linked Yes))))
```

```
PP6
(($IF ((same (fact Process) 'Chlorination_of_Phenol)
    (not-empty (fact Operating-Units))
    (same (fact Product) 'pentachlorophenol)
    (empty (fact linked))))
 ($THEN ((port 0 'Stirred_batch_reactor-1
                 'Stirred_batch_reactor-1)
     (port 1 'Stirred_batch_reactor-1 'WASTE)
     (port 2 'Stirred_batch_reactor-1 'PRODUCT)
     (fact linked Yes))))
```

```
PP7
(($IF ((same (Status (material in process)) 'active)
    (same (fact linked) 'Yes)))
 ($THEN ((fact* 'Current_Operating-Unit
          (destination (material in process)))
     (activate (fact Current_Operating-Unit)))))))
```

C R K B - Combining Rules Knowledge Base:
contains the rules for the creation of the Operating Units by combining the Unit hardware descriptions (UNITKB) with the corresponding Unit Activities (UAKB).

```
(Rule-List (CR1 CR2 CR3 CR4 CR5 CR6 CR7))

(Rules

 (CR1
  (($IF ((in (fact Unit-Activities) 'Halogenation)))
   ($THEN ((create Operating-Unit Stirred__batch__reactor Halogenation)
       (remove-fact Unit-Activities Halogenation))))

  CR2
  (($IF ((in (fact Unit-Activities) 'Distillation)))
   ($THEN ((create Operating-Unit Batch__vacuum__distillation__column
             Distillation)
       (remove-fact Unit-Activities Distillation))))

  CR3
  (($IF ((in (fact Unit-Activities) 'Condensation)))
   ($THEN ((create Operating-Unit Condenser Condensation)
       (remove-fact Unit-Activities Condensation))))

  CR4
  (($IF ((in (fact Unit-Activities) 'Absorption)))
   ($THEN ((create Operating-Unit Absorption__tower Absorption)
       (remove-fact Unit-Activities Absorption))))

  CR5
  (($IF ((in (fact Unit-Activities) 'Reflux)))
   ($THEN ((create Operating-Unit Reflux__drum Reflux)
       (remove-fact Unit-Activities Reflux))))

  CR6
  (($IF ((in (fact Unit-Activities) 'Pumping)))
   ($THEN ((create Operating-Unit Recycle__pump Pumping)
       (remove-fact Unit-Activities Pumping))))

  CR7
  (($IF ((in (fact Unit-Activities) 'Condensing)))
   ($THEN ((create Operating-Unit Condensing__trap Condensation)
       (remove-fact Unit-Activities Condensing))))))
```

I R K B - Input Rules Knowledge Base:
contains the rules for the input of the substances required in the production process.

```
(Rule-List (IR1 IR2 IR3 IR4 IR5))

(Rules

 (IR1
  (($IF ((in (fact Input-Table) 'phenol)
     (not-same (fact Product) 'pentachlorophenol)))
   ($THEN ((material phenol
                (From 'Input-Table)
                (To 'Stirred_batch_reactor-1)
                (Phase 'liquid)
                (Hazard_ratings '(L M H H))
                (Status 'active))
       (material phenol
                (From 'Input-Table)
                (To 'Stirred_batch_reactor-2)
                (Phase 'liquid)
                (Hazard_ratings '(L M H H))
                (Status 'active))
       (remove-fact Input-Table phenol)))))

  IR2
  (($IF ((in (fact Input-Table) 'phenol)
     (same (fact Product) 'pentachlorophenol)))
   ($THEN ((material phenol
                (From 'Input-Table)
                (To 'Stirred_batch_reactor-1)
                (Phase 'liquid)
                (Hazard_ratings '(L M H H))
                (Status 'active))
       (remove-fact Input-Table phenol)))))

  IR3
  (($IF ((in (fact Input-Table) 'chlorine)))
   ($THEN ((material chlorine
                (From 'Input-Table)
                (To 'Stirred_batch_reactor-1)
                (Phase 'gas)
                (Hazard_ratings '(M L H M))
                (Status 'active))
       (remove-fact Input-Table chlorine))))
```

```
IR4
(($IF ((in (fact Input-Table) 'aluminum_chloride)
    (greater (temperature 'Stirred_batch_reactor-1) 90)))
 ($THEN ((material aluminum_chloride
              (From 'Input-Table)
              (To 'Stirred_batch_reactor-1)
              (Phase 'solid)
              (Hazard_ratings '(L L H H))
              (Status 'active))
     (remove-fact Input-Table aluminum_chloride))))


IR5
(($IF ((in (fact Input-Table) 'H2O)
    ($OR (same (fact Product) 'o-chlorophenol)
       (same (fact Product) 'p-chlorophenol)
       (same (fact Product) 'dichlorophenol_2/4)
       (same (fact Product) 'trichlorophenol_2/4/6)
       (same (fact Product) 'tetrachlorophenol_2/3/4/6))))
 ($THEN ((material H2O
              (From 'Input-Table)
              (To 'Absorption_tower-1)
              (Phase 'liquid)
              (Hazard_ratings '(L L L L))
              (Status 'active))
     (remove-fact Input-Table H2O))))))
```

R P M K B - RPM Knowledge Base:
general KB containing the Products and Input Materials.

```
(Products
  (o-chlorophenol
   p-chlorophenol
   dichlorophenol__2/4
   trichlorophenol__2/4/6
   tetrachlorophenol__2/3/4/6
   pentachlorophenol))

(Input-Material
  (phenol chlorine aluminum__chloride H2O))
```

METARULES:
contains the Metarules for the control of the rule application sequences.

```
(Rule-List (MR1 MR2 MR3 MR4))


(Metarules


(MR1
 (($IF ((not-empty Input-Table)
     (empty MIPT)))
  ($THEN ((apply* Production_Process-Rules)
       (apply* Combining-Rules)
       (apply* Production_Process-Rules)
       (apply* Input-Rules))))


MR2
 (($IF ((not-empty Input-Table)
     (not-empty MIPT)
     (empty Product-Table)))
  ($THEN ((apply* Input-Rules)
       (apply* Production_Process-Rules))))


MR3
 (($IF ((empty Input-Table)
     (not-empty MIPT)
     (empty Product-Table)))
  ($THEN ((apply* Production_Process-Rules))))


MR4
 (($IF ((empty Input-Table)
     (empty MIPT)
     (not-empty Product-Table)
     (same (fact finished) 'No)))
  ($THEN ((fact finished Yes))))))
```