Abstract—P&IDs (Piping and instrumentation diagrams) are detailed schematic diagrams of process plants. The production of P&IDs is tedious; involving continual checks of correct connectivity, correct equipment selection, etc. A fuzzy expert system has been developed to aid in PFD to P&ID conversion. The system has been constructed using a set of fuzzy rules based upon individual unit operations. It consists of two main parts: database and application program. The database has been designed using Rational Rose software, and it has been implemented using InterBase. The application program has been designed based on object oriented paradigm and UML standard, and then it has been implemented using Borland C++ Builder compiler. Performance of the developed software has been evaluated based on Tennessee Eastman benchmark. This case study shows that the software is capable of handling the PFD to P&ID conversion providing a rich knowledge base.

Index Terms—Expert System, Fuzzy Logic, process Flow Diagram, Piping and instrumentation diagram

I. INTRODUCTION

The design of complex chemical plants requires the integration of very large amounts of diverse information. Process designers make use of standardized sets of graphical representations to assist in the development and transfer of these information sets. The design of the process usually follows an ordered pattern, with the development of schematized drawings presenting slightly different but increasingly complex amounts of information, starting with simple input/output diagrams, progressing through complete material and energy flow information in Process Flow Diagrams (PFDs) and detailed process interconnections in Piping and Instrumentation Diagrams (P&IDs). Initial P&IDs possess a large amount of errors. As there are no tools available, expert engineers and draft teams spend many extensive and tedious hours checking for these types of errors by hand.

Human experts solve problems by taking into account all data from the process and use their knowledge to solve them. An expert system uses the same principle but is also shielded from many factors (i.e. stress, emotional situations, etc.) that can affect the human expert [1]. An expert system can also be designed to solve problems with incomplete data and/or non-exact solutions through the use of fuzzy logic in the inference engine [1]. Fuzzy rule-based expert systems contain such fuzzy rules in their knowledge base and derive conclusions from user inputs and through a fuzzy reasoning process [2]-[4]. These systems have been proved to be effective tools for representing vague knowledge [5]-[7] and are now widely used to perform troubleshooting and control operations because of these similarities to human reasoning as well as their simplicity [2]-[4], [8].

Fuzzy expert systems use a collection of fuzzy rules, instead of Boolean logic, to reason about data [9]. These systems can, therefore, effectively translate qualitative knowledge into numerical reasoning [10] and are primarily concerned with quantifying and reasoning using natural languages in which words have ambiguous meanings [4]. A common term, computing with words, that has been introduced by Zadeh [11] is now often used to explain the notion of reasoning linguistically rather than quantitatively [10]. Such reasoning is able to capture uncertainty and understanding in human experiences vis-a-vis the description and operation of a process [3], [12].

Many applications of fuzzy logic have appeared over the years. These include manufacturing [2], [4], [8], [13], reliability analysis [10], economics [2]-[4], [8] and even medical diagnosis [10], [14], [15]. Fuzzy logic has also been used in several petroleum-engineering-related applications including petrophysics [16], [17], reservoir characterization [18], enhanced oil recovery [19], infill drilling [20], decision-making analysis [21], and well simulation [22], [23].

The purpose of this work is to develop a fuzzy expert system to simplify the task of transforming PFDs into P&IDs. Study of current literature has revealed that there is no specific information on this topic.

The fuzzy expert system converts PFD to P&ID by following these steps: (1) The PFD is defined in the fuzzy expert system. (2) The fuzzy expert system creates objects corresponding to elements of the PFD and then sets properties of the objects. (3) The fuzzy expert system considers each object and specifies whether or not the object supports control configuration, and if it supports, the fuzzy expert system searches in the knowledge base to find all control configurations for the object. (4) Finally, a suitable control configuration for each object is selected and a P&ID report of all selected control configurations is prepared.
II. GENERAL DESCRIPTION OF FUZZY EXPERT SYSTEMS

An expert system is a computer program designed to model the problem solving ability of a human expert. Expert systems solve problems using a process that is very similar to the methods used by a human expert, using a structure shown in fig. 1.

The knowledge base contains highly specialized knowledge on the problem area as provided by the expert. It includes problem facts, rules, concepts, and relationships. How this knowledge can be coded in the knowledge base is the subject of knowledge representation. Five of the most common knowledge representation techniques are used in the development of an expert system: (1) Object-attribute-value triplets. (2) Rules: A knowledge structure that relates some known information to information that can be concluded or inferred to be known. (3) Semantic networks: A method of knowledge representation using a graph made up of nodes and arcs where the nodes represent objects and the arcs the relationships between the objects. (4) Frames: A data structure for representing stereotypical knowledge of some concept or object. (5) Logics.

The inference engine is the knowledge processor which is modeled after the expert's reasoning. The engine works with available information on a given problem, coupled with the knowledge stored in the knowledge base, to draw conclusions or recommendations. How this engine can be designed is the subject of inference techniques. There are two types of inference techniques: (1) Forward chaining: Inference strategy that begins with a set of known facts, derives new facts using rules whose premises match the known facts, and continues this process until a goal state is reached or until no further rules have premises that match the known or derived facts. (2) Backward chaining: Inference strategy that attempts to prove a hypothesis by gathering supporting information.

The working memory is the part of an expert system that contains the problem facts that are discovered during the session. During a consultation with an expert system, the user enters information on a current problem into the working memory. The system matches this information with knowledge contained in the knowledge base to infer new facts. The system then enters these new facts into the working memory and the matching process continues. Eventually, the system reaches some conclusion that is also entered into the working memory.

A fuzzy expert system is an expert system whose knowledge base rules have fuzzy premises. The inference engine of this system should be modified for working with these fuzzy rules. The fuzzy inference engine compares information of the rule fuzzy premises and working memory information to calculate rule certainty [24].

III. DEVELOPMENT OF THE EXPERT SYSTEM

A. Designing and implementation of the working memory using object oriented concepts

The working memory designing has two alternatives: (1) Designing a hierarchical structure for mapping all the PFD objects to the software environment so that all necessary object variables with their values are transferred. (2) Designing a structure for representation of the control configuration.

"PFD Based Object" class is the base of all classes that are designed for first alternative [25]. A part of this structure is shown in Fig. 2.

The two levels of derived class from PFD Based Object class

There are several approaches for control configuration representation. It is an important point that each control configuration has several pairings and each pairing has one or several measured variables (input) as well as one or several manipulated variables (output). (Pairings can be SISO or MIMO.)

One method of control configuration representation is using "Variable" class as a base of this representation [25]. This model is shown in Fig. 3.
There is one object of the "Variable" class for SISO pairing (or more than one object for MIMO pairing) that is assigned to "input" and "output" attribute of "Pairing" class. The owner of the variable objects is specified in the "Owner" attribute of the "Variable" class. "Manipulatable" and "Measurable" are two attributes of the "Variable" class that determine whether the variable object is input or output. Finally, all the pairing objects of a control configuration are added to "Pairings: TList*" attribute of the same control configuration object.

B. Designing and implementation of the inference engine using object oriented concepts

The inference engine is implemented using "Agent Manager" class, which is shown in Fig. 4.

This class has several functions. These functions use backward chaining inference technique so that for each control configuration of current "PFD based object", all premises of relevant rule are in comparison to variables of the object. If certainty of the rule is exceeded from defined threshold, the rule will be fired and the control configuration is selected. The most important function of the "Agent Manager" class is "Find Object". This function finds objects in the working memory, which is corresponded to instances in the knowledge base.

Both working memory and inference engine are designed using UML standard and Rational Rose software and then are implemented using Borland C++ Builder [26].

C. Designing and implementation of the knowledge base

Two major types of knowledge representation are used for representing knowledge about PFD to P&ID conversion. These types are "Frame" and "Rule". The knowledge about the structure of control configurations of defined unit operations is represented using "Frames". The knowledge about supporting heuristics is represented using "Rules". These heuristics define necessary conditions of selecting each control configuration.

Two collections of interrelated table are used for knowledge base implementation. The tables, shown in Fig. 5, are used for control configurations and unit operations defining.

The tables, shown in Fig. 6, are used for defining of supporting heuristics for available control configurations.

The knowledge base is designed using Rational Rose software and implemented using InterBase [27].
Figure 5. Tables of control configurations and unit operations defining
IV. ANALYZES AND RESULTS

The fuzzy expert system user interface is contained a main window and several dialog boxes to set variables of objects or select suitable control configuration among several available control configurations. The main window has a report area page that final P&ID (control configurations) report is displayed on it. The main window of the fuzzy expert system is shown in Fig. 7.

The knowledge base of the fuzzy expert system is

Figure 6. Tables for defining of the supporting heuristics
developed by integrating knowledge from many experienced
process engineers and handbooks [28].

Performance of the developed software has been evaluated
base Tennessee Eastman benchmark. This process is well
suited for a wide variety of studies including both plantwide
control and multivariable control problems.

A. Tennessee-Eastman Process Description

The process produces two products from four reactants
with an inert and a byproduct making a total of eight
components: A, B, C, D, E, F, G, and H. The reactions are:

\[
\begin{align*}
A (g) + C (g) + D (g) & \rightarrow G (liq), \quad \text{Product 1} \\
A (g) + C (g) + E (g) & \rightarrow H (liq), \quad \text{Product 2} \\
A (g) + E (g) & \rightarrow F (liq), \quad \text{Byproduct} \\
3D (g) & \rightarrow 2F (liq), \quad \text{Byproduct}
\end{align*}
\]

All the reactions are irreversible and exothermic. The
reaction rates are a function of temperature through an
Arrhenius expression.

The process has five major unit operations: the reactor, the
product condenser, a vapor-liquid separator, a recycle
compressor and a product stripper. Fig. 8 shows a diagram
of the process.

![Figure 7. The main window of the fuzzy expert system](image)

The gaseous reactants are fed to the reactor where these
components react to form liquid products. The gas phase
reactions are catalyzed by a nonvolatile catalyst dissolved in
the liquid phase. The reactor has an internal cooling bundle
for removing the heat of reaction. The products leave the
reactor as vapor along with the unreacted feeds. The catalyst
remains in the reactor.

The reactor product stream passes through a cooler for
condensing the products and from there to a vapor-liquid
separator. Noncondensed components recycle back through a
centrifugal compressor to the reactor feed. Condensed
components move to a product stripping column to remove
remaining reactants by stripping with feed stream C in figure
8. Product G and H exit the stripper base and are separated in
a downstream refining section which is not included in this
problem. The inert and byproduct are primarily purged from
the system as a vapor from the vapor-liquid separator [29].
Figure 8. Tennessee Eastman test problem
B. Generating control configurations for Tennessee-Eastman Process

First the PFD of the Tennessee Eastman plant is defined for the fuzzy expert system. After running of the fuzzy expert system, the following P&ID report is produced.

The diagram of this report is shown in Fig. 9. It is apparent from the report and Fig. 9 that there aren't suitable control configurations in the knowledge base for some unit operations.

The knowledge base can be enriched using machine learning procedure. In this procedure suitable control configurations along with supporting heuristics are added to the knowledge base gradually.

For comparing Expert system method to another method of control configuration production, another design of control configurations [30] are shown in fig. 10.

It is apparent from Fig. 9 and 10 that the control configurations of the fuzzy expert system have been simple pairing loops but most pairing loops in the control configurations of Fig. 10 are cascaded. The fuzzy expert system applies a control configuration for each unit operation independently but the control configurations of Fig. 10 are derived using plantwide ideas and this is the main criticism of the expert system method.

V. DISCUSSION AND CONCLUSIONS

A fuzzy logic-based expert system was developed for PFD to P&ID conversion. The derived system is very useful to reduce conversion time, increasing the performance and reliability of the conversion process, and avoiding usual errors and mistakes. The system was developed by integrating knowledge from many experienced process engineers and handbooks. A standard benchmark was utilized to test the developed system. The results revealed the success of the implemented system in solving real problems.

In this study, the fuzzy logic-based expert system was used for preparing control configurations for the input PFD. However, the overall approach used in this work can be of general validity. It can, for instance, be applied to a wide spectrum of industrial process units. Besides selecting control configuration, the expert system can generate complete P&ID if the knowledge base is enriched. The lack of plantwide ideas in the conversion procedure is an important drawback of this work. This will be considered in the future investigations.

<table>
<thead>
<tr>
<th>P&amp;ID Report:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected control configuration applies on Reactor is:</td>
</tr>
<tr>
<td><strong>Pairing211:</strong></td>
</tr>
<tr>
<td><strong>Pairing Component:</strong> PID_Controller211</td>
</tr>
<tr>
<td><strong>Input Variable:</strong> Temperature of Reactor</td>
</tr>
<tr>
<td><strong>Output Variable:</strong> HeatFlow of QReactor</td>
</tr>
<tr>
<td><strong>Pairing212:</strong></td>
</tr>
<tr>
<td><strong>Pairing Component:</strong> PID_Controller212</td>
</tr>
<tr>
<td><strong>Input Variable:</strong> Flow of ToReactor</td>
</tr>
<tr>
<td><strong>Output Variable:</strong> Flow of ToReactor</td>
</tr>
</tbody>
</table>

---

Selected control configuration applies on Condenser is:

**Pairing311:**
- **Pairing Component:** PID_Controller311
- **Input Variable:** Pressure of FromReactor
- **Output Variable:** Flow of CWR

---

Selected control configuration applies on Pump1 is:

**Pairing511:**
- **Pairing Component:** PID_Controller511
- **Input Variable:** Flow of TopInletStripper
- **Output Variable:** Flow of TopInletStripper

---

Selected control configuration applies on Pump2 is:

**Pairing521:**
- **Pairing Component:** PID_Controller521
- **Input Variable:** Flow of Product
- **Output Variable:** Flow of Product

---

Selected control configuration applies on Compressor.
Figure 9. The PFD with Control Configurations
Figure 10. Another design of control configurations for comparing
REFERENCES


