FINAL
TECHNICAL REPORT OF
ExpHAZOP$^+$ DEVELOPMENT

Project Funded by
Petroleum Research Atlantic Canada (PRAC)

Date: November 14, 2007
Executive Summary

Process hazard analysis is an important step to identify risk in an oil and gas processing facility. Automation of hazard identification requires efficient search techniques with the aid of a knowledge base. It also requires an easy menu-driven interface so that a common user can interact with the system with minimal intervention from an expert. One of the models to implement automation of hazard analysis is a HAZOP (hazard and operability) study. This report presents development work of an automated HAZOP tool, which relaxes the constraints of the traditional HAZOP method, and also acts as an aid for conducting real-life HAZOP studies. Fault propagation, an aspect of HAZOP analysis, defines the propagation of deviations among equipment in a process facility. This is often missing in traditional HAZOP tools. To identify all the possible hazards more rapidly, an efficient fault propagation algorithm with a knowledge-base is implemented in the present study. This report presents the outcome of the study and is comprised of three parts.

The first part of the report presents the methodology and pseudo-code for the development of an automated HAZOP tool, named ExpHAZOP+. This part also discusses the technical validation and effectiveness of the developed tool “ExpHAZOP+” with a well-known case study. The second part of the report presents an operating manual for the newly developed tool ExpHAZOP+. A step-by-step procedure on how to use the tool and troubleshoot is also presented in this section of the report. The third part of the report presents a research paper based on the current study. The manuscript is under consideration for publication in the Journal of Loss Prevention in the Process Industries. This paper illustrates the salient points of the developed tool, ExpHAZOP+, and potential applications.
# TABLE OF CONTENTS

0.0 Executive Summary ................................................................. 1  
1.0 Introduction .............................................................................. 2  
2.0 ExpHAZOP+ Development ......................................................... 3  
3.0 Algorithm of ExpHAZOP* .......................................................... 4  
4.0 Features of ExpHAZOP* ............................................................. 6  
   4.1 Graphical User Interface (GUI) ................................................. 6  
   4.2 Inference Engine ................................................................. 6  
   4.3 Report Generator ............................................................... 6  
5.0 Methodology for ExpHAZOP* .................................................... 6  
   5.1 Graphical User Interface (GUI) ................................................. 7  
   5.2 Knowledge Base ................................................................. 8  
   5.3 Inference Engine ............................................................... 10  
6.0 Implementation of ExpHAZOP* ................................................ 12  
   6.1 General Purpose Module .................................................... 14  
   6.2 Equipment and Pipe Module ............................................... 17  
   6.3 Data Module ..... ................................................................. 18  
   6.4 Connectivity Analysis Module ............................................ 21  
   6.5 Fault Propagation Module .................................................. 23  
7.0 Steps for HAZOP Analysis with ExpHAZOP* ............................ 24  
8.0 HAZOP Analysis with ExpHAZOP* ........................................... 26  
9.0 Conclusion .............................................................................. 33  
10.0 Future Recommendations ....................................................... 33  
11.0 Acknowledgement .................................................................. 34
1.0 INTRODUCTION

Process Hazard Analysis (PHA) aims to assess the possible system hazard and risk of a chemical plant or a process operation that may arise as a result of malfunctions of any equipment or an operation unit. PHA has several techniques, i.e., fault tree analysis, event tree analysis, what-if analysis, and HAZOP (hazard and operability) to find out the possible system or equipment hazard. Hazard and Operability Study, more commonly known as HAZOP analysis, is one of the more systematic and standard analysis of PHA for identifying the potential hazards and operability problems of a new or revamped plant or process operation. It uses a formal review technique of piping and instrumentation diagram (P&ID), or Engineering Flow Diagrams (EFD’S) that covers every vessel, conduit, valve and all control equipment in a typical process facility. Process parameters and Guide Words are the main features of HAZOP study. By means of combinations of these parameters and Guide Words, conceivable deviations of the proposed process design condition are systematically searched for and discussed in a team for the possible causes and consequences related to the process system. Systematically, a P&ID of a process system is divided into different sections known as study node and the relevant process parameters i.e., flow, pressure, temperature and level are identified and an intention is created for the node under consideration. Finally, the Guide Words i.e., no, more, less, reverse are combined with the process parameters to identify the deviations that occurred within the process unit or operation. For example, the guideword "High" is combined with a process parameter “temperature” to give the deviation "High temperature". HAZOP study then focuses on listing all the credible causes of “High temperature" deviation beginning with the cause that can result in the worst possible
consequence. After finding out the possible causes for the deviation then the HAZOP study records the possible consequences, safeguards and any recommendations deemed appropriate. The study is then checked for the next deviation and so on until completion of the node. The process continues until all nodes of the process system are covered by the HAZOP analysis.

An outline of the current report is as follows. Section 2 focuses on the background of ExpHAZOP+ development. The algorithm for ExpHAZOP+ development is presented in section 3. ExpHAZOP+ features are summarized in section 4. Section 5 discusses the methodology used for the ExpHAZOP+ development. The implementation of ExpHAZOP+ as an automated HAZOP analysis tools has been described in section 6. Section 7 shows the steps required to work with ExpHAZOP+. A worked out example of HAZOP analysis with ExpHAZOP+ is illustrated in Section 8.

2.0 ExpHAZOP+ DEVELOPMENT

In many process facilities, a HAZOP study is conducted manually. Due to a lack of sophisticated tools and expertise, the results of such studies remain in paperwork form and important data may be lost over time. Furthermore, the team members performing the study may become unnecessarily occupied with the system complexities and lose their focus of study. Automated HAZOP analysis overcomes these limitations and has several advantages over manual HAZOP analysis. The main advantages are:

- There is a significant saving of human effort and manpower cost by reviewing most commonly occurring fault conditions.
- The HAZOP study process becomes standardized by automatically recalling process parameters and reviewing the results.
- The study is easier to track thus making it more auditable and reliable.
The results of the automated study can be an aid for manual HAZOP analysis thus saving the overall time to conduct the study.

It can be an aid for an operator in a process facility as a decision support tool.

The results of the study can be stored for later use or printed in a report format.

An efficient fault propagation algorithm and a strong knowledge-base expert system can accelerate the performance of automated HAZOP analysis. ExpHAZOP+ has been developed to cover these two features of automated HAZOP analysis. In ExpHAZOP+, Exp means expert knowledge-base information system, which is developed based on the information derived from past HAZOP analysis related to process operation, process equipment, operating problems, failure mode, and failure frequencies; the term HAZOP+ means using an advanced fault propagation algorithm in HAZOP analysis system that will cover all downstream equipment of a process operation.

3.0 ALGORITHM OF ExpHAZOP+

A fault in a process facility is a weakness or an error within the system that may arise as a result of an equipment failure, human error or unexpected chemical reaction within the system. Certain faults may not be hazardous for certain equipment but may become fatal when they propagate to other equipment. The algorithm of ExpHAZOP+ is developed in a way that can be able to determine all credible hazard consequences for the affected downstream equipment. It consists of a Graphical User Interface (GUI), fault propagation algorithm, expert knowledge base, and inference engine system. A GUI is used for drawing P&IDs of a process system; the fault propagation algorithm along with expert knowledge base is used to detect the possible causes and consequences of the process system that arise as a result of parameter deviations in downstream equipment. The inference engine system assists the whole process to find out the related causes and
ExpHAZOP+ an automated tool for HAZOP analysis

effects of each component of the process system. Figure 1 depicts the algorithm that is used to develop the methodology for ExpHAZOP+

Figure 1: The algorithm for ExpHAZOP+
4.0 FEATURES OF ExpHAZOP+

The three main features of ExpHAZOP+ are described below.

4.1 GRAPHICAL USER INTERFACE (GUI)

The graphical user interface allows the user to perform HAZOP analysis by drawing P&IDs using existing equipment or user-defined equipment.

4.2 INFERENCE ENGINE

The inference engine acts as a search engine which searches for the occurrence of a particular user-defined deviation in the database system for all equipment.

4.3 REPORT GENERATOR

The report generator displays the output generated as a result of the user input given in the graphical user interface and the search performed in the inference engine. The output is displayed in an MS formatted document which can be printed if required.

5.0 METHODOLOGY FOR ExpHAZOP+

The architecture of ExpHAZOP+ consists of the following three components: a graphical user interface, a knowledge base, and an inference engine. The object oriented architecture of ExpHAZOP+ is presented in Figure 2. This figure shows the interconnection between the three components.
5.1 GRAPHICAL USER INTERFACE (GUI)

The graphical user interface is an open interface in which a user can draw a P&ID of a process unit via “drag and drop” of equipment icons from the toolbar menu. A database containing equipment information is connected with all equipment used in a process operation. A list of process variable deviations is automatically generated when a user is going to interact with equipment on the GUI. When a user selects a piece of equipment or a group of equipment with a corresponding process variable deviation, the causes and consequences related to the process variable deviation are displayed on a separate screen. A HAZOP analysis is carried out by the user after defining the equipment name, equipment picture, the deviations related to the equipment as a result of abnormal causes.
and adverse consequences. The graphical user interface will continually prompt the user until all data has been entered. This data is directly updated in the database in a tabular format with the other existing equipment tables. Upon completion of user data entry and P&ID drawing, the user selects the section of the P&ID in which the HAZOP analysis is required. This section is called the study node. The study node is denoted by drawing a rectangular region in the graphical user interface. The output for the analysis is the deviation with causes and consequences. The user has the option to store the analysis for later user or output it for Report Generation. The Report Generation outputs the result of the user defined HAZOP analysis in MSWord format.

5.2 KNOWLEDGE-BASE

A knowledge base is a collection of information set up in a pre-defined format which can be retrieved and used for HAZOP study. This information is gathered from past HAZOP analyses performed on the equipment. This knowledge is stored in a database which can be extracted when required. Data can be added by the user to derive meaningful results from automated HAZOP analysis. The addition of data can be specific to the process facility structure. The information is developed using practical industrial case studies of various process facilities. If the information is properly presented and maintained, it can be used for a long period of time and also for other safety-related study purposes. Based on the collected information, a computerized knowledge base is built using MS Access as the back-end and Visual Basic as the front-end.

The acquisition of knowledge used in ExpHAZOP+ encompasses process equipment as well as process operations. The knowledge base is characterized in seven different operations commonly present in a process facility. Each operation and the equipment
required to perform the operation in the ExpHAZOP+ knowledge base are categorized as follows:

- **Mass transfer operation**: units involving the transfer of one or more components from one stream to another stream (e.g. extractors, adsorbers, dust collectors, electrostatic precipitators, filters).

- **Heat transfer operation**: units involving the transfer of heat from one stream to another stream (e.g. air cooled exchangers, carbon block exchangers, shell and tube exchangers, plate and frame exchangers).

- **Mass and heat transfer combined**: units involving both chemical and heat transfer (e.g. distillation columns).

- **Reaction**: units involving chemical transformations including oxidation (e.g. continuous stirred-tank reactors, plug flow reactors, autoclaves).

- **Transportation**: units involving transfer or movement of chemicals from one place to another (e.g. piping, compressors, pumps, valves).

- **Storage**: units involving bulk quantities of chemicals (e.g. vessels).

- Other physical operations, such as mixing, compression, relief venting, purging, etc.

An operation may encompass one or more pieces of equipment. For example, transportation units consist of piping, compressors, pumps and valves. The knowledge base incorporates 19 different pieces of equipment related to the process facility. The equipment units are: adsorber, air cooled exchanger, blower, carbon block exchanger, centrifuge, compressor, cyclone, distillation column, dust collector, electrostatic
precipitator, extractor, filter heat exchanger, piping, pump, reactor, temperature sensor, temperature controller, and valve.

Figure 3 describes the hierarchy or the organization of the knowledge base. In this figure, each piece of equipment consists of a set of deviations. Each set of deviation consists of a set of causes. Each cause corresponds to a consequence. The knowledge for all equipment either general or specific is entered in the same format, i.e. the equipment name, corresponding deviations, causes and consequences. The equipment tables are connected by means of the common deviations (generic or specific) entered by the user.

![Knowledge base diagram](image)

**Figure 3:** The architecture of the knowledge base

### 5.3 INFERENCE ENGINE

The inference engine is an intermediary between the knowledge base and the graphical user interface. It acts as a search engine which searches for deviations selected by the user.
user throughout the study node selected by the user in the GUI. The output information of
the search can be updated manually according to the operator’s knowledge during the
decision making process. The fault propagation algorithm in the inference engine assists
to propagate the deviations and identify the causes and consequences for the deviations.
This algorithm uses the following sequential steps to determine the causes and
consequences for the user-defined deviations of a system.

1) Identify equipment based on unit’s operation.

2) Select the study node to perform the HAZOP analysis.

3) Identify the starting and ending equipment.

4) Select the deviation for the starting equipment.

5) Determine if the deviation is related to process generic operation.

6) Apply propagation method to identify general causes and consequences.

7) Determine if the deviation is related to process specific operation.

8) Apply propagation method to identify specific causes and consequences.

9) Move to the next equipment unit.

10) End the search if this is the last equipment. Otherwise repeat the procedure from

   step 3 – 8.

Once the user draws the process and instrumentation diagram and selects the study
type node in the graphical user interface, the inference engine is used to perform a search
within the equipment in the study node. The search is performed in the database tables for
both process general and process specific knowledge. The search is performed based on
the deviation selected by the user. An example of a deviation can be “LESS
TEMPERATURE”, “MORE PRESSURE” etc. The starting and the ending equipment are
chosen by the user along which the deviation to propagate. The deviation propagates all the way downstream, or as far as the last equipment specified by the user. The deviation for process specific operation similarly propagates from equipment to equipment, based on the process specific deviation provided by the user. If certain equipment does not contain the user chosen deviation, then that equipment does not produce an output.

However, as a result of propagation, the next equipment which might have that specific deviation in its database will generate an output. The end result is identification of all the possible causes and consequences that might result in failure of the process facility.

6.0 IMPLEMENTATION OF ExpHAZOP+

ExpHAZOP+ uses VISUAL BASIC 6.0 for coding in real life implementation. The code has been developed in such a way that all individual modules and class modules can be integrated to a common framework to enable starting from a common window.

ExpHAZOP+ consists of six main modules: Equipment Module, Pipe Module, Data Module, Connectivity Analysis Module, Fault Propagation Analysis Module and General Purpose Module. Each module performs a specific task, and is linked with the other modules. For example, the Fault Propagation Analysis Module identifies propagation of deviations in equipment provided in the form of input/output relations by the Data Module. Figure 4 shows the inter-relation between all modules in ExpHAZOP+.
Figure 4: Architecture of the message flow of the modules in ExpHAZOP+
6.1 GENERAL PURPOSE MODULE

This module deals with handling of data files, output files, and general flow of information. The general purpose module also works as a “passageway” in that it provides the necessary information to each module and sub-module to carry out desired operations, and stores the results in different files. It also provides all commonly used file operations such as copying, deleting, printing and report generation of the result of HAZOP analysis. The pseudo codes that are used to develop this module are given below.

```plaintext
Function mnuOpen_Click
    Opens a previously saved ExpHAZOP+ file with .tzr extension

Function load_FILE
    Loads an existing ExpHAZOP+ file with .tzr extension in the GUI.

Function mnuSave_Click
    Saves the current content of the GUI in a file with .tzr extension.

Function mnuExit_Click
    Exits the ExpHAZOP+ application

Function Toolbar1_ButtonClick
Repeat
    While the shapes adding are not finished
        If a button is click on the toolbar then
            theBlockCollection store the shape with an specific TagID ShapeCaptioning
        End if
    End While

*Connect two equipments
    If Click equipment connection button then
        If selected 1st shape and 2nd shape to connect then
            Store line in theLineCollection.AddLine
            Check TagID of the two shapes
```

---

ExpHAZOP+ an automated tool for HAZOP analysis
Else
    MsgBox "Two objects should be selected!"
End If
End If

'Disconnect two equipments
If Click equipment connection button then
    theLineCollection.deleteLine from the selected shape.
End If

'Remove an equipment
If Click removal equipment button then
    Remove shape, TagID, and ShapeCaption from theBlockCollection.
End If
End function

6.2 EQUIPMENT AND PIPE MODULE

The equipment and pipe module consists of the general properties of equipment and pipes. For the equipment module, the properties include the storing of the equipment name, a unique tag, the height and width, as well as the symbol of the equipment in the P&ID. The pipe module stores similar information as well as the tag id for the input and output equipment connected with the pipe. These features enable a user to draw a P&ID. The equipment module also contains features to store the user-defined equipment data. The following codes are intends to show the idea that is used to develop this module.

Function mnuCaptoLine_Click
"Adds a caption to the connecting line between two equipments"
'Adds a Caption to a line between two shapes"
    If (1st SELECTED_SHAPE ≠ -1) And (2nd SELECTED_SHAPE <> -1) Then
        Store line caption in theLineCollection.AddCaptionToLine
    Else
        MsgBox "Two objects should be selected!"
    End If
End Function
Function for CaptionToShape()
"Adds caption to an equipment"
Declare shape caption as string
If SELECTED_SHAPE ≠ -1 Then
    Repeat
        While ending caption of all shape
            Check the shape selection
            Enter the caption for a shape
        Store Caption shape cblock.cls, myBlockCollections.cls
    End while
Else
    MsgBox "Object should be selected!"
End If
End function

Class modules to control Equipment property

Class module cBlock.cls
"This class module consists of setting and storing of individual equipment properties. It consists of the following class members"

Public Property Let TagID(ByVal vData As String)
    Stores the tag assigned to equipment.
Public Property Get TagID() As String
    Retrieves the tag assigned to equipment.
Public Property Let shapeHeight(ByVal vData As Integer)
    Stores the height assigned to equipment.
Public Property Get shapeHeight() As Integer
    Retrieves the height assigned to equipment.
Public Property Let shapeWidth(ByVal vData As Integer)
    Stores the width assigned to equipment.
Public Property Get shapeWidth() As Integer
    Retrieves the width assigned to equipment.
Public Property Let sCaptionUpper(ByVal vData As String)
    Stores the caption of assigned to equipment.
Public Property Get sCaptionUpper() As String
    Retrieves the caption assigned to equipment.
Public Property Let Visible(ByVal vData As Boolean)
    Stores the visibility property assigned to equipment.
Public Property Get Visible() As Boolean
Retrieves the visibility property assigned to equipment.

**Public Property Let shapeLeft(ByVal vData As Double)**
Stores the coordinate of the left edge of the equipment.

**Public Property Get shapeLeft() As Double**
Retrieves the coordinate of the left edge of the equipment.

**Public Property Let shapeTop(ByVal vData As Double)**
Stores the coordinate of the Top edge of the equipment.

**Public Property Get shapeTop() As Double**
Retrieves the coordinate of the Top edge of the equipment.

**Public Sub updateShapeCaptionPos()**
Updates the position of the Caption of equipments.

*Class Module myBlockCollections.cls*
This class represents a collection of equipments which are entered in the Graphical User Interface. Its member functions are follows:

**Public Function AddShape(PICTYPE As Integer, sKey As String) As cBlock**
This function adds equipment in the GUI and sets the properties of the equipment such as its width, height, name, starting position etc.

**Public Property Get Item(vntIndexKey As Variant) As cBlock**
This function is used when referencing an equipment element in the collection vntIndexKey contains either the Index or Key to the collection. A collection is a data structure where equipments are added.

**Public Property Get Count() As Long**
Counts the number of elements in the collection.

**Public Sub Remove(vntIndexKey As Variant)**
Removes an element from a collection.

**Private Sub Class_Initialize()**
Creates the collection when this class is created.

**Private Sub Class_Terminate()**
Destroys collection when this class is terminated.

**Public Sub removeShape(Index As Integer)**
This function removes currently selected equipment.

**Public Function getFreeTagID() As String**
This function assigns a unique id to each equipment.

**Public Function getIndexFromTag(sTag As String) As Integer**
This functions extracts the index number from the tag of an

### 6.3 DATA MODULE
In this module all data relevant to equipment and pipes are stored in the data structure. A separate data structure is maintained for pipes and equipment. The equipment data structure stores the equipment name, a unique tag, and the label by which the user identifies the equipment. The pipe data structure stores the pipeline number, the input connecting equipment, the output connecting equipment, and the user-defined label for the input and output equipment.

At first all the equipment and pipe data for the P&ID entered in the GUI are stored in their respective data structures. This is followed by filtering the data of the equipment and pipes lying inside the study node. This filtered data is stored separately in a two-dimensional array, thus allowing for faster access of data. An example data structure with a user-selected study node is shown in Figure 5. The study node is shown in the dotted rectangle.

**Figure 5:** A typical GUI input for a set of equipment with their tag and label connected to a set of pipelines
In Figure 5, each block represents a separate piece of equipment consisting of a unique tag represented by \textit{Id1..Id9}. The user-defined label (e.g. the equipment with tag Id1 has the user-defined label Ad1) is also shown along with the tag. In an actual GUI, only the user-defined label is displayed. The tag is shown here for analysis purpose only. The pipeline is numbered \textit{1..9} to identify the pipes connecting the equipment. The representation of pipeline data which is shown in Figure 5 is stored according to Table 1.

\textbf{Table 1:} Representation of the complete pipeline data entered in the GUI

<table>
<thead>
<tr>
<th>Id1</th>
<th>Ad1</th>
<th>1</th>
<th>Id2</th>
<th>Co1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id1</td>
<td>Ad1</td>
<td>2</td>
<td>Id3</td>
<td>Bl1</td>
</tr>
<tr>
<td>Id2</td>
<td>Co1</td>
<td>3</td>
<td>Id4</td>
<td>Co2</td>
</tr>
<tr>
<td>Id3</td>
<td>Bl1</td>
<td>4</td>
<td>Id6</td>
<td>Com1</td>
</tr>
<tr>
<td>Id3</td>
<td>Bl1</td>
<td>5</td>
<td>Id5</td>
<td>Bl2</td>
</tr>
<tr>
<td>Id2</td>
<td>Col1</td>
<td>6</td>
<td>Id5</td>
<td>Bl2</td>
</tr>
<tr>
<td>Id4</td>
<td>Co2</td>
<td>7</td>
<td>Id7</td>
<td>Val2</td>
</tr>
<tr>
<td>Id5</td>
<td>Bl2</td>
<td>8</td>
<td>Id8</td>
<td>Val1</td>
</tr>
<tr>
<td>Id7</td>
<td>Val2</td>
<td>9</td>
<td>Id9</td>
<td>Rea1</td>
</tr>
<tr>
<td>Id8</td>
<td>Val1</td>
<td>10</td>
<td>Id9</td>
<td>Rea1</td>
</tr>
</tbody>
</table>

The above data structure is filtered when the user draws a study node. The filtered data is stored in a two-dimensional array as shown in Table 2. The filtered data structure is used in the \textit{Connectivity Analysis Module}.

\textbf{Table 2:} Filtered data obtained from study node in Figure 5

<table>
<thead>
<tr>
<th>Id1</th>
<th>Ad1</th>
<th>1</th>
<th>Id2</th>
<th>Co1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id1</td>
<td>Ad1</td>
<td>2</td>
<td>Id3</td>
<td>Bl1</td>
</tr>
<tr>
<td>Id3</td>
<td>Bl1</td>
<td>5</td>
<td>Id5</td>
<td>Bl2</td>
</tr>
<tr>
<td>Id2</td>
<td>Col1</td>
<td>6</td>
<td>Id5</td>
<td>Bl2</td>
</tr>
</tbody>
</table>
The class modules pseudo codes which are used to develop this module are given below.

### Class modules, control connected equipments properties in the GUI

#### Class module cLine.cls

“This class represents the contents of a connected line and its properties such as its caption, tagid of the source and destination equipment it is connected. Its member functions are defined below”

**Public Property Let sCaption(ByVal vData As String)**

Stores the caption of a connected line.

**Public Property Get sCaption() As String**

Retrieves the caption of a connected line.

**Public Property Let bShowArrow(ByVal vData As Boolean)**

Stores the contents of the arrowhead of a connected line.

**Public Property Get bShowArrow() As Boolean**

Retrieves the content of the arrowhead of a connected line.

**Public Property Let sFrom(ByVal vData As String)**

Stores the tag id of the source equipment of a connected line.

**Public Property Get sFrom() As String**

Retrieves the tag id of the source equipment of a connected line.

**Public Property Let sTo(ByVal vData As String)**

Stores the tag id of the destination equipment of a connected line.

**Public Property Get sTo() As String**

Retrieves the tag id of the destination equipment of a connected line.

**Public Property Let sFromCap(ByVal vData As String)**

Stores the caption of the source equipment of a connected line.

**Public Property Get sFromCap() As String**

Retrieves the caption of the source equipment of a connected line.

**Public Property Let sToCap(ByVal vData As String)**

Stores the caption of the destination equipment of a connected line.

**Public Property Get sToCap() As String**

Retrieves the caption of the destination equipment of a connected line.

### Class Module myLineCollections.cls

“This class represents a collection of lines connecting two equipments which are entered in the Graphical User Interface”

**Public Function AddLine(sFrom As String, sTo As String, bShowArrow As Boolean, Optional sKey As String) As cLine**

Adds a new connecting line between two equipments. Stores line properties such as its coordinates, source equipment id, destination equipment id etc.

**Public Property Get Item(vntIndexKey As Variant) As cLine**
This function is used when referencing a line element in the collection VntIndexKey contains either the Index or Key to the collection. A collection is a data structure where equipments are added.

Public Function StoreCollectionItemsInArray(itemArray() As Integer, equiptag As Integer) As Variant
This function stores the entire equipment item and the connected lines in a two dimensional array mArray from the collection object mCol.

Public Sub deleteLine(xFrom As Integer, xTo As Integer)
This function deletes a connected line.

Public Sub updateLines()
This function updates the coordinates of the connected lines every time an equipment item is moved.

Public Sub AddCaptionToLine(sFrom As String, sTo As String, sCaption As String)
This function adds a caption to a line

Public Sub showArrow(lineObj As Line, arrowTo As String)
This function is called in update line and updates the position of the arrowhead each time the arrows are moved.

6.4 CONNECTIVITY ANALYSIS MODULE
The main purpose of this module is to find all the connecting paths from the starting equipment to the ending equipment. The connectivity module uses two procedures. The InputOutput Procedure and the FindConnection procedure. The InputOutput Procedure derives the tag Id of the starting and ending equipment while the FindConnection Procedure identifies all paths from the starting equipment to the ending equipment. The pseudo code for the FindConnection procedure is given below.

In the FindConnection procedure, mArray consists of the equipment and pipe data inside the study node including the starting and ending equipment. Initially, the anArrayRow consists of only the rows in the mArray which have the starting equipment. Then temparray tracks the paths that are connected from the starting equipment inside the study node from mArray. Thus it is assured that temparray will only apply until the last equipment selected by the user in the mArray. Every time temparray stores a new path
from the \textit{mArray} it is updated in \textit{anarrayRow}. In this example \textit{mArray} has the data structure shown in Table 2.

\begin{procedure} \label{procedure_findconnections}
\textbf{Procedure \texttt{FindConnections}()}

\Repeat
\While the end of \textit{mArray} is not reached
\If items in Column1 of \textit{mArray} matches the starting equipment
\Store that row for \textit{mArray} in \textit{anarrayRow}.
\Store the equipment connected with the starting equipment in \textit{aString}.
\EndIf
\EndWhile
\Repeat for each item in \textit{aString}
\While the end of \textit{mArray} is not reached
\If items in Column1 of \textit{mArray} matches the item in \textit{aString}
\Store this item in \textit{temparray}
\If this is the first occurrence of the item in \textit{mArray}
\Find the Row in \textit{anarrayRow} in which the item be added.
\Add the item in \textit{anarrayRow} (Row)
\Else
\Find the Row in \textit{anarrayRow} in which the item be added.
\Copy the \textit{anarrayRow}(Row)
\Add the item in the copied \textit{anarrayRow}(Row).
\EndIf
\EndIf
\EndWhile
\If there are elements in \textit{temparray}
\Erase \textit{aString}
\textit{aString} = \textit{temparray}
\Erase \textit{temparray}
\EndIf
\Until there are elements in the \textit{temparray},
\EndProcedure

\end{procedure}

Thus it is assured that all the connected paths from the starting equipment to the ending equipment are stored in \textit{anarrayRow}. This is represented in Table 3.

\textbf{Table 3:} All possible paths between starting and ending equipment
The user performing the HAZOP analysis will have the option to perform the analysis on any of the two paths from \textit{Id1} to \textit{Id5}. Once the user selects the path the fault propagation algorithm is applied using the \textit{Fault Propagation Module}.

\textbf{6.5 FAULT PROPAGATION MODULE}

This module applies the fault propagation algorithm in order to identify the causes and consequences of the deviation provided by the starting equipment. The module uses the function \textit{cmdPerformAnalysis} to find the propagation of deviation of the path chosen by the user. This function first distinguishes between process general and process specific knowledge. For process general knowledge it outputs the data directly from the equipment database. For process specific knowledge, the causes and consequences come from the user input which is stored in a temporary database. The data structure which gives the user an option to choose a path for HAZOP analysis is in the connectivity analysis module is used to access the equipment database. The pseudo code for \textit{cmdPerformAnalysis} is shown below:

\begin{verbatim}
Function cmdPerformAnalysis
    For each equipment in an array\texttt{Row} for the user chosen row
        If the equipment is user defined
            Access the user defined temporary database
            Output Deviation, Connecting input equipment, starting deviation, Causes and Consequences.
        Else
            Access the equipment database
            Output Deviation, Connecting input equipment, starting deviation, Causes and Consequences.
        End If
End Function
\end{verbatim}
7.0 STEPS FOR HAZOP ANALYSIS WITH ExpHAZOP+

ExpHAZOP+ follows the following consecutive steps to identify relevant hazards in a process system:

1) Development of P&ID: The piping and instrumentation diagram (P&ID) varies from one process facility to another. In order to perform the HAZOP analysis the user must have all the data for equipment entered in the P&ID. If the user changes equipment at a certain point of analysis, then ExpHAZOP+ identifies that point, such that the analysis restarts from that point. The HAZOP analysis may be performed on connected equipment as well as on single individual pieces of equipment.

2) Selection of process general and process specific equipment: The process general equipment is already available as menu buttons on the GUI. The database for this equipment is presented in Access Database. The current database contains data for 19 different pieces of equipment related to the process facility. The equipment units are: absorber, air cooled exchanger, blower, carbon block exchanger, centrifuge, compressor, cyclone, distillation column, dust collector, electrostatic precipitator, extractor, filter heat exchanger, piping, pump, reactor, temperature sensor, temperature controller and valve.

3) Selection of a study node: The study node is selected by the user to perform the HAZOP analysis. All the equipment and pipes are entered in a data structure in order to identify the connectivity of the equipment and pipes inside the study node. Only the equipment units inside the study nodes are considered as the starting and ending points of analysis.
4) **Performing HAZOP analysis:** This step involves selecting the starting and ending equipment on which the HAZOP analysis is to be performed. The process variable deviation applied to all equipment during the deviation propagation is also selected in this step. The deviation propagation algorithm described in Figure 7 is applied at this step.

COMMENT: Is Figure 7 the correct figure number? The last figure referred to is Figure 5.

5) **Creation of data structure:** All the equipment and pipes along with their connectivity are stored in a collection data structure. However, only the equipment and the connecting pipes inside the study node are stored in a two dimensional array. The names of the equipment and pipes are also stored inside this array for identification purposes.

6) **Selection of path for HAZOP analysis:** There can be more than one possible path from the starting equipment to the ending equipment. This path is sorted out once the user enters the starting and ending equipment for HAZOP analysis. If there is more than one path then all the paths are shown to the user so that the user can choose the specific path for HAZOP analysis. The equipment database is accessed for all equipment present in the selected path. The user performs the desired analysis on the selected path.

7) **Report Generation:** This step assists in generating the output report of the entire HAZOP analysis in a MS file. An expert can then further analyze this output report for final auditing and reviewing purposes. This would enable the HAZOP analysis to be more focused and less time consuming.
8.0 HAZOP ANALYSIS WITH ExpHAZOP+

To facilitate the HAZOP analysis, a system with four components i.e., blower, air cooled exchanger, adsorber, and centrifuge, has been considered to examine the blower exerting a high pressure effect on downstream equipment. The four components in this example have been captioned with blower as BL, adsorber as AD, centrifuge as CF and air cooled exchanger as ACE.

Step-1: P&ID drawing

The equipment ToolbarMenu is used for drawing a P&ID. To add a blower on the GUI, the user needs to click on the blower button and add its caption. The user can drag and move any item on the GUI by holding the mouse left button on the item and dragging it where it needs to be placed. The desired two items need to be clicked first, followed by clicking Connect equipment under Equipment operation in the Menubar to connect two items on the GUI. Figure 6 shows the P&ID for the current example.

![Figure 6: P&ID for the example.](image-url)
Step-2: Selection of process general and process specific equipment:

For this example, the equipment and related data are already pre-loaded to perform the analysis. Hence the approach taken here is process generic.

Step-3: Study node selection

Once the drawing of the P&ID is completed, the next step is to draw a study node. The study node is drawn using “Mark Study Node” under “Study Envelope” Menu. This allows the user to perform HAZOP analysis once the study node is selected. Figure 7 shows the study node selection for the example.

![Study node for the example](image)

Figure 7: Study node for the example.

Step-4: Creation of data structure:
All the equipment and their connecting paths are loaded in the data structure.

**Step-5: Performing HAZOP analysis**

The user can now begin the HAZOP analysis by selecting the “Start HAZOP analysis” under the “Analysis” Menu. A new window will open with the heading “HAZOP ANALYSIS”. The user needs to define the starting and ending equipment along with deviations for performing HAZOP analysis for a specific problem. In the example above for the given P&ID and the study node (dotted rectangular line) the starting equipment is “blower” (with caption “BL”) and the ending equipment is “air cooled exchanger” (with caption “ACE”). All input must be entered in order to conduct a valid HAZOP analysis. An error message will otherwise be generated. After defining the starting and ending equipment, the user clicks “Perform Hazop Analysis” button to start the HAZOP analysis (Figure 8).

![HAZOP ANALYSIS](image)

**Figure 8:** HAZOP analysis for the example.

**Step-6: Selection of path for HAZOP analysis**
Once the input and output equipment are selected, another window appears. Based on the P&ID and the starting and ending equipment, this window finds all the different possible paths of connectivity between the starting and ending point, and presents the user with different options for connection between the input equipment and output equipment. For the above example the different possible paths to perform HAZOP analysis are shown in Figure 9.

![Path Selection for HAZOP analysis](image)

**Figure 9:** Path selection for the example.

In the example above for the given input, the user has to select one of the two paths to perform the HAZOP analysis. The connection paths are derived from the P&ID drawn by the user. In the above analysis the user selects the first path ("BL 1 AD 4 ACE"). The numbers 1 and 4 represent the pipelines connecting the equipment. The user then performs the HAZOP analysis on the selected path.

**Step-7: Report Generation**

If the user clicks on “Performed HAZOP on selected track” under the “Path Selection for HAZOP analysis” window, then a report is automatically generated based on the
given user information for the defined problem. The output report for the current example is given in Table 4.

**Table 4**: Report of the HAZOP analysis with ExpHAZOP+.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>CONNECTING EQUIPMENT</th>
<th>ORIGINAL DEVIATION</th>
<th>DEVIATION</th>
<th>CAUSES</th>
<th>CONSEQUENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>blower</td>
<td></td>
<td>Original deviation of More Pressure from BL</td>
<td>More Pressure</td>
<td>Leakage on suction side of compressor pulls air into system</td>
<td>flammable atmosphere</td>
</tr>
<tr>
<td>adsorber</td>
<td>From equipment BL Pipe 1</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>Migration of internals into lines</td>
<td>blockages / rupture</td>
</tr>
<tr>
<td>adsorber</td>
<td>From equipment BL Pipe 1</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>Blockage of packing / trays</td>
<td>rupture</td>
</tr>
<tr>
<td>adsorber</td>
<td>From equipment</td>
<td>Original</td>
<td>More</td>
<td>Process</td>
<td>vaporization of the</td>
</tr>
<tr>
<td>ITEM CONNECTING EQUIPMENT</td>
<td>ORIGINAL DEVIATION</td>
<td>CAUSES</td>
<td>CONSEQUENCES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------</td>
<td>--------</td>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air cooled exchangers</td>
<td>From equipment</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>Corrosion/erosion of exchanger internals</td>
<td>heat transfer surface leak or rupture and possible overpressure of the low pressure side</td>
</tr>
<tr>
<td>Air cooled exchangers</td>
<td>From equipment</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>Excessive heat input</td>
<td>vaporization of the cold-side fluid (e.g., control system failure, cold-side</td>
</tr>
<tr>
<td>BL Pipe 1 deviation of More Pressure from BL</td>
<td>pressure</td>
<td>liquid reintroduced into improperly cooled adsorber</td>
<td>process liquid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air cooled exchangers</td>
<td>From equipment AD Pipe 4</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>fouling, accumulation of non-condensables, or loss of cooling medium from condensing side</td>
<td>Loss of heat transfer capability / higher tube wall temperatures / overheating of reactive material / loss of tube strength / excessive differential thermal expansion</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Air cooled exchangers</td>
<td>From equipment AD Pipe 4</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>Cold-side fluid blocked in while heating medium continues to flow</td>
<td>overheating</td>
</tr>
<tr>
<td>Air cooled exchangers</td>
<td>From equipment AD Pipe 4</td>
<td>Original deviation of More Pressure from BL</td>
<td>More pressure</td>
<td>Ambient temperature increase</td>
<td>higher vaporization rate in air heated exchanger</td>
</tr>
</tbody>
</table>
9.0 CONCLUSION

To facilitate use of a knowledge base expert system in HAZOP analysis in a more systematic way, an expert tool ExpHAZOP+ has been developed in this work. The aim of the ExpHAZOP+ tool is to integrate the expert knowledge base system with an efficient fault propagation algorithm, which can significantly reduce the time and effort, manpower cost, and repetitive work often associated with a manual HAZOP analysis system. A systematic algorithm and methodology has been followed to interface the HAZOP analysis procedure with the knowledge base expert system. To implement the developed framework of ExpHAZOP+, this work uses software engineering methodologies in every stage of its implementation, including design of the architecture, development of the code base, and testing of the software. The ExpHAZOP+ tool has the following properties:

(1) Drawing of a user-defined P&ID.

(2) Automatic generation of the report for HAZOP analysis.

(3) Robust and flexible knowledge base structure.

(4) Flexibility for users to implement their specific knowledge and ideas.

(5) Ability to propagate deviations, thus enabling identification of possible causes and consequences for downstream equipment.

10.0 FUTURE RECOMMENDATIONS
ExpHAZOP+ tool is an automated software for analyzing the HAZOP study of a process system. The addition of the following features with the existing ExpHAZOP+ tool can make this tool more interactive and user friendly:

- Improve the knowledge base system through updating its access database system.
- Add an interface to update the knowledge system and the equipment list.
- Develop a procedure to find the propagation of different kinds of deviations among equipment (e.g., How “MORE TEMPERATURE” in one equipment causes “MORE PRESSURE” in another equipment and identify the deviations of such).

11.0 ACKNOWLEDGMENT

The research team gratefully acknowledges the Petroleum Research Atlantic Canada (PRAC) for providing the necessary financial support to carry out this project successfully.