Is AI Really Doing HAZOP — or Something More Limited?

The Issue

There is a growing claim, both in the literature and on commercial websites, that artificial intelligence can now read a P&ID and automatically produce a HAZOP record. At first sight, this sounds remarkable. HAZOP has always been a labour-intensive, expert-led process. It requires engineers, operators, control specialists and process safety professionals to sit together, examine a system node by node, apply guidewords, challenge assumptions, identify credible deviations, and test whether safeguards are real, effective and sufficient. If an AI system could simply read a P&ID and fill in the HAZOP table, that would appear to be a major step forward.

But the claim needs unpacking very carefully. The question is not whether an LLM can produce something that looks like a HAZOP table. It clearly can. The real question is whether it has understood enough about the system to produce a defensible HAZOP record. That is a much higher bar.

Clarification

The first distinction is between a smart P&ID and a dumb P&ID. A smart P&ID is not just a drawing. It is an engineering database presented visually. Before the drawing is produced, the project environment normally defines tag formats, line-numbering conventions, class libraries, approved symbols, attribute sets, instrument naming rules, validation rules and data structures. When the engineer places a pump, valve, vessel, instrument bubble or line on the drawing, they are not merely drawing a symbol. They are instantiating a tagged engineering object with properties and relationships. The diagram is therefore a view of a structured model.

A dumb P&ID is different. It may be a PDF, scan, image, or non-intelligent CAD drawing. It may contain all the human-readable information that an engineer needs, but it does not contain reliable machine-readable semantics. The computer does not inherently know that a particular mark on the drawing is a centrifugal pump, that a nearby text label is its tag, that a line is connected to its discharge nozzle, or that an instrument bubble is measuring pressure on that line. Those meanings have to be inferred, extracted, digitised and checked.

This is where DEXPI becomes important. DEXPI provides a neutral, standardised way of representing process-industry engineering information. It gives a disciplined structure for equipment, piping, instrumentation, properties, tags and topology. In effect, it provides a semantic target. A smart P&ID can potentially be exported into a DEXPI-style representation. A dumb P&ID cannot. It must first be interpreted: symbols recognised, tags read, lines traced, topology reconstructed, and relationships mapped into a structured form. Only then can it become a credible basis for a knowledge graph.

Improvements

Prompting an LLM to "use DEXPI" can help, but it does not solve the underlying problem. It gives the model a better organising language. Instead of producing a loose visual description, the LLM can be asked to classify visible items as candidate equipment, piping components, instruments, process lines, signal lines, annotations and topology. That is useful. It can produce a candidate extraction table. It can say, "this appears to be a pump," "this appears to be a line," or "this instrument appears to be associated with that vessel." But the word "appears" matters. Without validated digitisation, these are not engineering facts. They are hypotheses.

A sensible workflow would therefore treat the LLM as a triage assistant rather than as the source of truth. From a dumb P&ID, it can produce a draft inventory of likely equipment, tags, valves, instruments, line routes, off-page connectors and apparent relationships. It can propose candidate triples for a knowledge graph. But each node and edge should carry provenance, confidence and review status. A relation such as "V-101 candidate-connected-to P-101" is very different from "V-101 connected-to P-101." The former is a claim awaiting validation. The latter is a model fact.

The situation improves when the P&ID is not the only source. A mixed dataset may include process descriptions, operating procedures, line lists, equipment lists, control narratives, cause-and-effect charts and previous HAZOP records. These documents can compensate for one another. The P&ID suggests visible structure. The process description explains what the process is intended to do. Procedures describe sequences, line-ups, checks and operator actions. Equipment and line lists stabilise tags and attributes. Control narratives reveal the logic that is often only partly visible on the drawing. Together, these sources can form the basis of a much richer evidence graph.

Even then, the graph should be understood as a claim graph before it becomes a truth graph. The LLM can reconcile evidence, highlight conflicts and propose relationships. It can say that a procedure mentions opening a valve before starting a pump, that a line list confirms the line number, and that the P&ID appears to show the pump discharging into that line. That is useful. But it is still not the same as a validated process model.

Automatic scribes

This matters directly when we examine claims about automatic HAZOP generation. HAZOP has a regular worksheet structure, which makes it attractive for automation. Once a system has identified a node, it can apply guidewords to relevant parameters. "No flow," "more pressure," "less temperature," "reverse flow," "other than composition" and similar deviations can be generated very quickly. The system can then use prior HAZOP examples, component failure libraries, process descriptions and LLM drafting to suggest causes, consequences, safeguards and actions.

That is how the apparent magic happens. The tool converts the P&ID and associated documents into some form of structured plant representation. It divides the process into HAZOP nodes. It infers or retrieves the design intent. It applies a guideword-parameter matrix. It filters for plausible deviations. It proposes causes based on known failure modes, visible neighbouring equipment and operating procedures. It suggests consequences by looking downstream or upstream. It identifies visible or documented safeguards. Finally, the LLM turns this into fluent HAZOP worksheet language.

This can be extremely useful. It may save time, reduce blank-page effort, improve consistency, and help teams prepare for a HAZOP more efficiently. It may also expose missing tags, unclear node boundaries, unlinked instruments or gaps in the documentation. Used well, it could provide a valuable preparation layer.

But it is not the same as doing the HAZOP.

The core limitation is that a P&ID-derived knowledge graph is usually local and topological. It contains components, tags, instruments, piping connections and immediate upstream or downstream relationships. It can tell the AI what is near what. It can show that a pump discharges into a line, that a valve is on that line, that a transmitter measures pressure, or that a vessel is downstream. What it usually does not contain is a whole-system process model. It does not necessarily know the process dynamics, operating modes, material behaviour, control-loop response, abnormal line-ups, operator recovery actions, start-up conditions, shutdown conditions, maintenance states or the functional dependencies that make a system behave as a system.

This means that the “effects” generated by such systems are often local effects. They are effects around the node, or effects on immediate neighbours. A pump deviation may produce a downstream starvation warning. A valve failure may produce a local flow restriction. A pressure deviation may flag a nearby pressure instrument or relief device. These are useful observations, but they are not necessarily system-level consequence analysis.

For that reason, it may be more accurate to describe many of these systems not as fully automated HAZOP, but as AI-assisted Deviation or Variability Modes and Local Effects Analysis. In classical HAZOP language, the guideword-parameter combination produces a deviation from design intent. In a broader systems language, it identifies a mode of variability: no flow, more pressure, less temperature, reverse flow, wrong material, late action, failed indication, or inappropriate control response. The AI then uses the graph to infer local effects and visible safeguards.

That is a narrower and more defensible claim. The AI is not necessarily understanding the whole plant. It is identifying candidate variability modes at a node and flagging likely local effects based on nearby structural relationships and retrieved text. It may be very good at that. But the scope of the claim should match the scope of the model.

Transparency matters

This distinction also explains why a simpler, more honest tool may sometimes be safer than a more elaborate one. A system that openly says, “I can pre-populate candidate HAZOP rows from available evidence, but every row is unverified,” creates the right kind of caution. A system that presents a polished, confident, fully populated HAZOP worksheet may create a false sense of trust. The danger is not that the AI generates rows. The danger is that plausible rows are mistaken for validated process-safety judgement.

Every generated HAZOP row should therefore carry a health warning. What is the source? Was the P&ID smart or dumb? Was the topology validated? Was DEXPI used as an actual structured representation or merely as a prompting schema? Was the cause drawn from the local graph, a generic library, a previous HAZOP, a procedure or an LLM inference? Was the safeguard visible, documented, independent, tested and effective? Is the consequence local, downstream, systemic, dynamic or merely generic?

These questions matter because HAZOP is not just table filling. It is structured expert challenge. It is a social and technical process for discovering how a design, procedure or operation may fail to achieve its intent. The table is the record of that process, not the process itself.

Conclusion

So the more cautious conclusion is this: AI can help pre-fill HAZOP records, especially when smart P&IDs, DEXPI, knowledge graphs and supporting documents are available. It can classify components, identify candidate nodes, apply guidewords, suggest deviations, retrieve similar cases and draft local causes and consequences. But unless it is supported by a much richer process model, it remains limited to local topology and evidence-based inference. It is better understood as AI-assisted Variability Modes and Local Effects Analysis than as autonomous HAZOP.

That is not a criticism of the technology. It is a clarification of what it is actually doing. Used honestly, it could be very valuable. Used carelessly, it could turn HAZOP into a confidence-generating paperwork exercise. The right direction is not to reject AI assistance, but to make its boundaries visible: candidate not confirmed, local not systemic, graph-based not dynamic, pre-filled not validated, and drafted not signed off.

The safer claim is not “AI does HAZOP,” but “AI can support Variability Modes and Local Effects Analysis as a disciplined pre-HAZOP drafting layer.”

David Slater

References

1. **Center for Chemical Process Safety (CCPS)** (2022) *Guidelines for Hazard Evaluation Procedures*. 4th edn. Hoboken, NJ: John Wiley & Sons.International

2. **DEXPI e.V.** (n.d.) *Specifications*. Available at: DEXPI specifications page (Accessed: 31 May 2026).

3. **Kim, B.C., Kim, H., Moon, Y., Lee, G. and Mun, D.** (2022) 'End-to-end digitization of image format piping and instrumentation diagrams at an industrially applicable level', *Journal of Computational Design and Engineering*, 9(4), pp. 1298–1326. doi: 10.1093/jcde/qwac056.

4. **Goldstein, D.P., Schulze Balhorn, L., Alimin, A.A. and Schweidtmann, A.M.** (2025) 'pyDEXPI: A Python framework for piping and instrumentation diagrams (P&IDs) using the DEXPI information model', *Systems and Control Transactions*, 4, pp. 1365–1370. doi: 10.69997/sct.139043.

5. **Alimin, A.A., Goldstein, D.P., Schulze Balhorn, L. and Schweidtmann, A.M.** (2025) 'Talking like Piping and Instrumentation Diagrams (P&IDs)', *Systems and Control Transactions*. doi: 10.69997/sct.159477.

6. **Alimin, A.A. and Schweidtmann, A.M.** (2026) 'GraphRAG for Engineering Diagrams: ChatP&ID enables LLM interaction with P&IDs', *arXiv preprint*, arXiv:2603.22528.

7. **Oeing, J., Holtermann, T., Welscher, W., Severins, C., Vogel, M. and Kockmann, N.** (2023) 'preHAZOP: Graph-based safety analysis for early integration into automated engineering workflows', *Chemie Ingenieur Technik*, 95(7), pp. 1083–1095. doi: 10.1002/cite.202200222.

8. **Elhosary, E., Moselhi, O. and Bucur, C.** (2024) 'Evaluation of AI-assisted HAZOP software tools', *IChemE Hazards 34 Symposium Series*, 171.