Knowledge Representation Standards and Interchange Formats for Causal Graphs

David R. Throop	Jane T. Malin	Land Fleming
The Boeing Company	NASA Johnson Space Center	Hernandez Engineering
2100 Space Park Drive	2101 NASA Road 1	17625 El Camino Real
Houston, TX 77058	Houston, TX 77058	Houston, TX 77058
281 460-8415	281 483-2046	281 483-2055
david.r.throop@boeing.com	jane.t.malin@nasa.gov	land.d.fleming1@jsc.nasa.gov

Abstract[1,2]—In many domains, automated reasoning tools must represent graphs of causally linked events. These include fault-tree analysis, probabilistic risk assessment (PRA), planning, procedures, medical reasoning about disease progression, and functional architectures. Each of these fields has its own requirements for the representation of causation, events, actors and conditions. The representations include ontologies of function and cause, data dictionaries for causal dependency, failure and hazard, and interchange formats between some existing tools. In none of the domains has a generally accepted interchange format emerged. The paper makes progress towards interoperability across the wide range of causal analysis methodologies. We survey existing practice and emerging interchange formats in each of these fields. Setting forth a set of terms and concepts that are broadly shared across the domains, we examine the several ways in which current practice represents them. Some phenomena are difficult to represent or to analyze in several domains. These include mode transitions, reachability analysis, positive and negative feedback loops, conditions correlated but not causally linked and bimodal probability distributions. We work through examples and contrast the differing methods for addressing them. We detail recent work in knowledge interchange formats for causal trees in aerospace analysis applications in early design, safety and reliability. Several examples are discussed, with a particular focus on reachability analysis and mode transitions. We generalize the aerospace analysis work across the several other domains. We also recommend features and capabilities for the next generation of causal knowledge representation standards.

Table of Contents

1. Introduction

Many disciplines reason with causal networks and must represent them. Within a discipline, multiple tasks and tools must share the information in causal. There are several efforts underway in different disciplines, all aimed at defining an interchange standard for causal networks. The current effort surveys emerging standards in Systems Engineering, medical reasoning, Bayesian networks and the representation of procedures.

All these efforts are generally moving towards XML representations of causality, but choosing XML (or RDF or OWL) is but a first step in defining a representation. Some basics are central to all the disciplines and will certainly be part of any emergent standard –
- There are events and conditions;
- They have causal links between them;
- Events, conditions and causes have associated probabilities;
- Causes combine via logical operators (AND, OR and NOT.)

The thrust of this paper is toward describing the types of causal phenomena that could be captured in a comprehensive interchange format. We have had discussions with practitioners in each of these disciplines. We elicited descriptions of the causal phenomena that are useful and common in reasoning tasks, but which present representational difficulties. It is our hope that discussing these hard representational problems, with examples drawn from multiple domains, will drive the emergence of interchange formats that can represent the most complex causal phenomena and that can be shared across very different disciplines.

Standard formats for interchange, autogeneration and comprehensibility

Users need standard formats so that they can model systems without being tied to a single tool or vendor. Moreover, users need standard formats so that they can automatically generate inputs from electronic sources. Fault-tree analysis (and the related PRA) illustrate this. There are several available tools and vendors for these analyses, each with its own virtues. Models built in one tool are not generally readable by the others. But there's a larger problem. Fault trees built in *any* of these tools are manually entered. The knowledge source for this data entry has mostly come from some other electronic data source. E.g., it is taken from functional allocations, FMEA reports, the utility connectivity (power, data, thermal, lube) and from schematics (CAD, wiring diagrams, P&ID.) This entry is incomplete – carrying all the relevant data from the sources is too laborious for a manual effort, so available information (part numbers, locations codes, the design engineer's name) is omitted. Worse, as soon as the fault tree is built, it is severed from its sources. The system changes; its design changes; its parts are substituted, renamed and renumbered; its documentation is corrected. These updates do not flow into the fault trees.

Standard interchange formats for causal modeling tools will greatly reduce the manual data entry and resulting errors and omissions. Users will automatically generate and update causal models from verified and version-controlled sources. Users will be able to import causal information from libraries – libraries whose use can span multiple tools. This will also enable the analysis tools to add features that manual data entry renders impractical – features such as hyperlinks to manuals, photography, or viewing the source files (functional allocations, schematics FMEA sheets etc.)

Causal graphs have two very different major uses. They are employed so that automated inference engines can read them and perform analysis – automated diagnosis or PRA. But they also aid human comprehension – people look at (sometimes huge) causal graphs, asking questions. "What went wrong? What would have to fail before we lost this capability?" An interchange standard must not only address whether a format is machine-readable. It must also ensure that the encoded knowledge is human-inspectable. It's not enough that machines understand these models. We have to, too.

2. Causal Models and Interchange Formats

In researching this work, we have been in contact with several groups in different disciplines that are working on interchange format standards for causal graphs or things akin to causal graphs. The groups may learn from each other and might borrow other group's best ideas. Accordingly, this section and the next will review briefly

- Knowledge representations in different disciplines, highlighting any unusual or especially useful concepts, capabilities and representations.
- Vocabularies as aids to making the semantics of a node explicit, available both to the tool and the user.
- Our own work interchanging event sequences between an ontology and a simulation engine
- Several emerging standards for interchange formats
- Representational challenges – causal relationships that are difficult to represent; attention to these during standards formulation might extend the representational scope.

We've mentioned several fields that use causal models. These represent scenarios as directed graphs (often trees) of causally linked events. Some features are common across all the domains and most of the tools. Nodes in the graphs generally represent events or conditions; edges represent causality. There are special nodes in the graph, called gates, which combine causation through simple logical operators (and, or, not) or higher-order operators (xor, nand, k-of-m or arbitrary truth tables.) The models are usually built with nodes uniformly either being successes or failures; there are methods for transforming between success-space representations and failure-space representations. Nodes with only incoming edges (such as the root of a tree) represent some final state of interest. Nodes with only outgoing edges (the leaves of a tree) are 'basic events' for which no causal explanation is necessary or available. Probabilities – representing the likelihood or frequency of a condition being true – are associated with the basic events and can be computed for other nodes.

Causal models in various fields

Let us review the prominent causal-network representations from different fields, and any associated interchange formats. Our interest in interchange formats for causal models was sparked by our risk-analysis work in functional and physical models. [10] Our software generates event sequences for potential accident scenarios. We needed to communicate these scenarios between software tools. We sought any existing interchange formats, rather than inventing our own. *[Jane, do I need to say more about RTSAD and ECS here?]* Practitioners in many fields perform causal modeling; our work is most akin to the work in fault trees.

In fault trees, [17] the root of the tree is some 'top event' – a failure that has occurred and is being investigated, or a potential failure to guard against. Analysis generates 'cut sets' – sets of basic events that are sufficient to cause the top event. Fault trees are composable – fault trees for individual components can be produced separately, and a system level fault tree can reference them. Some causal

propagations depend upon a system's state or configuration; Dynamic Fault trees conditionalize subtrees accordingly. In some situations, causes are mutually reinforcing. The fault tree methodology does not allow circular causality; there are techniques for breaking these feedback loops.
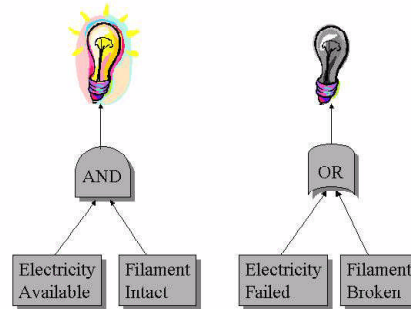
Figure 1: Success-space and failure-space models can be intercoverted.

Probabilistic Risk/Reliability Assessment (PRA) [15] is an extension to fault trees wherein probabilities are assigned to the nodes and the likelihood of the top event is calculated. It is customarily used to evaluate likelihood of success of an overall project or mission, and to focus attention on areas where loss is most likely to occur. The extension noted for Dynamic Fault trees allows PRA to span missions where the configuration and the risks repeatedly change; e.g. the launch, mission and return of a spacecraft

Two software suites, saphire [14] and cafta, are the most-used fault-tree analysis and PRA tools in aerospace. Both are descended from set, a 1980's era fault tree tool. Its file format became a *de facto* standard interchange format. Saphire, cafta and some other tools can save models in this format. But the fault tree field has advanced. Set's format is inadequate today; for instance, it can represent the combination of two conditions (and or or,) but not the negation of a condition (not.) It does not encode events' probabilities (e.g. Mean Time Between Failure, MTBF.) There is no common interchange format between saphire and cafta, but there is a NASA-commissioned software that translates between saphire's MAR-D format and cafta.

Root cause analysis [3] represents scenarios similarly to fault trees. It concentrates mainly on analyzing undesirable events that have already occurred. The trees are built in failure space. Nodes, gates and edges have essentially the same semantics as in fault tree. Probabilities are not represented and cut sets are not generated. Root cause analysis allows inclusion of causes that were not present in

the actual sequence – alternative conditions which could have caused or prevented the loss, if they had been present. It has an explicit representation of when causal chains have crossed organizational boundaries (and are thus out of the direct control of the investigating team.)

Bayesian networks employ causal networks similar to PRA trees. Nodes in both techniques store probabilities, but the semantics of a probability varies between the approaches. Edges in a Bayesian network can describe correlation, rather than causation *per se.* Some Bayesian networks associate a probability with causal edges as well as with the nodes. Bayesian networks are used both for diagnostics and prognostics. A proposed Bayesian Network Interchange Format [12] includes representations for probability distributions (including multi-modal distributions) not just fixed-point probabilities.

Procedures are event sequences and there is normally at least an implicit causal linkage between procedure steps. Copma iii [2] has developed an Extensible Procedure Action, or XPA, as an XML representation of procedures. XPA does not represent probabilities. It does not use logical gates; it has Condition blocks that evaluate comparative relationships (equality, precedence) that supply similar functionality. Copma iii holds much of its procedure descriptions as blocks of natural-language text, related graphically as predecessor/successor and parent/child relations. It explicitly represents whether subtasks may be performed sequentially or in parallel [1]

Qualitative Reasoning [6] is not, strictly speaking, a causal network model; its networks represent simultaneous non-directional constraints. It bears mentioning because its M^+ constraint captures a causal relationship between continuous variables. M^+(Voltage, Current) means the voltage and current must increase together. In Qualitative Process Theory [5] this relation implies an influence rather than a fixed constraint – increasing voltage tends to increase current, if other influences do not change. Neither methodology uses an interchange format. The M^- constraint is analogous. The *d/dt* constraint expresses a time-derivative relation, e.g. *d/dt*(Position, Velocity).

Controlled Vocabularies

In medicine, the HL7 (an ANSI Standards Development Organization, www.hl7.org) is developing the Reference Information Model, RIM [19]. RIM is not an interchange format; it is an approach to specifying data formats. The RIM orders how medical terms can be composed (how to specify a concept like "a benign fibrous tumor partially removed from the anterior of the liver.") This is done using controlled vocabularies (organized in taxonomies, or ontologies) and compositional rules. RIM is a meta-standard for specifications; several sets of RIM-compliant terminologies are being advanced, including SNOMED [20], GLIF [21] and GELLO [22].

Neither RIM nor these approaches specifies a causal-network interchange format *per se.* But RIM's use of controlled vocabularies is applicable to causal networks. In causal networks, nodes have names, which represent conditions, events, actions etc. In the other methodologies surveyed so far, the node names are free-form natural language text. The user can put any arbitrary text string in the name; automated inference engines analyzing the graph do not use the string except to report results.

Controlled vocabularies offer several advantages. A node name would be a well-formed composition of defined entities and actions. This would allow authoring tools to prompt users for default causes, automation of model-checking (spotting causal links that don't make sense, or inferring missing links) and to automate the linking of causal-graph tools to other online resources (databases, webpages, illustrations and photography…) Controlled vocabularies make information more comprehensible to users, too. Medical terminology suffers from the "Shakespeare Syndrome" – a profusion of different (long, Latinized) terms, all meaning the same thing. (A heart attack is a myocardial infarction is acute coronary heart disease..) Practitioners frequently do not understand each others' jargon. This is not just a problem across national boundaries; it's an issue when sending medical records between departments within a hospital. Controlled vocabularies allow practitioners to be sure they are understanding each other. We know this is also an issue in the aerospace community; we believe it bedevils other communities as well.

Interchange between specification and simulation

Our own work [8,9,10] provides an interchange format between two software tools. HIT (Hazard Identification Toolkit, implemented in Protégé) holds a functional model of a system (including its normal and abnormal operation.) CONFIG is a hybrid discrete/continuous simulator.

The two applications are interfaced to assist in identifying hazards in the operation of a system. A user of HIT constructs the system model from components selected from the HIT library of component classes and component connections. The description of a HIT component class may specify the entities that are the products and side effects generated by the component in different modes of operation and also the component's vulnerabilities to external entities while in a given mode.

A rule-based HIT tool can then perform a reachability analysis that finds paths over which an entity generated by one component in the system may reach other components for which the entity generated is specified as a vulnerability. Such a completed path may be viewed as a potential hazard to the system's functioning.

The user may then construct an operational scenario that could produce the hazardous situation. The HIT model can then be translated to a CONFIG model, and the operational scenario can similarly be translated to a script to be executed by the CONFIG simulator.

While CONFIG has no formal representations of hazards and vulnerabilities, its numerical simulation capabilities may be used to further assess the plausibility and severity of potentially hazardous scenarios identified by HIT. HIT creates a specification for a CONFIG system model, and also sends a hazard scenario (an accident or potential accident) to CONFIG. HIT has reasoned qualitatively that a scenario is of concern; CONFIG confirms it in a numerical simulation. The HIT/CONFIG scenario interchange format specifies features applicable to any causal-model interchange format. It distinguishes between different item-classes: functions, entities, hazards, mitigations and behaviors. It locates these items in computational ontologies. That is, there are taxonomies (inheritance hierarchies) of item-classes, with attributes for the classes. Events in the scenario are built by using composing these items and assigning attribute values.

Emerging standards for interchange formats

There is widespread recognition of the usefulness of a standard interchange format, especially for fault trees and PRA. Several groups are proposing standards.

Standard for the Exchange of Product model data, or STEP (ISO-10303) [13] is an existing family of International Standard for the computer-interpretable representation and exchange of product data. It is in use in the Systems Engineering community. A team is working on its AP233 and AP239, which will include a standard for representing fault trees. STEP currently models in EXPRESS, an 80's-era format. AP239 will move to a Uniform Modeling Language (UML) representation. Working in tandem with the AP239 effort, SysML Partners, in its Systems Modeling Language effort [16], is developing modeling standards for fault trees.

ISO-17666 [18] *Space Systems – Risk Management* is an emerging standard, originating from the European Space Agency. While it doesn't set forth representational standards for causal graphs *per se*, its documentation

standards for risks call out relevant features and a standard terminology for risks. Its risks are similar to fault tree's top events. Its definition and procedures for determining event criticality could provide the semantics for criticality measures in an interchange format.

3. Representational Challenges

There is much commonality in all these methodologies. Presumably, that which is common is sure to be included in *any* new representational standard. But today, complex causal explanations are not usually captured in any computer readable format, be it proprietary or open. For the most part, though, these causal explanations are captured in English text, diagrams and animated PowerPoint charts. For an excellent example of this, see [4]. Any emerging representational standard should anticipate its most demanding applications. It should be capable of expressing complex causal relationships – relationships that exceed the expressiveness of any of today's software tools. As we researched the domains discussed in the last section, we questioned practitioners. We asked, "What critical causal relationships are most taxing? Most difficult to explain, to represent, to reason about? What critical thing is likely to get left out of a standard?" This section synthesizes their answers, along with our own experiences in automating reasoning about causation.

Good computing practice—The standard should encode for each node and each edge:

- Its name
- An author
- The creation and modification dates (with an explicit standard for date format)
- A unique ID (unique within the causal model)
- A package (so that multiple causal models can be composed without ID conflict)
- An optional reference ID (pointing back to the informational source, such as an identifier in a wiring diagram)
- A comment field (with an explicit standard about what characters are allowed)
- The condition or variable which the node represents, with Boolean values. enumerated values, or numeric ranges with units
- A list of labeled URLs, pointing to associated or supplemental information.

Complex relationships—There are relationships that tax the expressive ability of today's tools. We do not know the best ways to encode them in a standard, but any group producing a standard interchange format should address

them.

- Exploded/Subsumption relations – generally, each causal step in an event sequence can potentially be expanded into its own causal network. The lower level network may take inputs from conditions and events not shown at the higher level.
- Feedback loops – fault-tree analysis and PRA prohibits causal feedback loops. The representation should store the location where the loop is broken. It should also be possible to specify information about the loop itself – whether it is positive or negative feedback, the gain around the loop, or the time-constants associated with it.
- Most cyclic behaviors are representable as closed causal loops. Cyclic behavior is abstracted to parameters (frequency, amplitude, gain, decay rate) at a higher level. Causal inputs at the lower cyclic level must flow up to the parameter level.
- Time to effect – causal links may have latencies
- Alternative testimonies of events – Accident reports are used to build a fault tree. Witnesses may give contradictory testimony. Fault trees don't currently capture this.
- Divergent/Convergent causation – Cause C1 increases intermediate condition I1 and decreases I2. I1 and I2 both promote effect E1, so C1 has a mixed effect on E1. Represent the information that one leg of the causal chain is stronger than the other, and that increasing C1 tends to increase E1.
- Discretization – Causal reasoning involve converting ranges of continuous variables into discrete-valued variables. A numeric temperature value gets simplified into Hot/Normal/Cold. The numeric endpoints of the qualitative ranges may be conditionalized on the system state.
- Represent mode transitions.
- Represent events that are correlated but where causality is unknown.
- Systematic conditions can affect the probabilities of many nodes at once. Poor maintenance, corrosive environments, inadequate training, financial pressure, low morale, startup/turnaround… These systemic effects should be representable without having causal edges drawn from each systemic cause to every single affected node.
- A causal subgraph may be repeated in a large causal graph, representing multiple instances of the same general type, but with different inputs. E.g. in a fault tree, a loss may have been caused by a series of operational errors caused by inexperienced personnel. Joe was out with the flu Wednesday; Jane was out with the flu Friday. The 'employee sick' subgraph should be instantiated twice, with different values.
- Race conditions occur when there are two causal chains with opposite effects; whichever finishes first controls. Again, these need to explicitly representable and

abstractable to a higher level.

- Software fails differently than do components. In fault tree/PRA, it is conventional to treat software failure similarly to either a component failure or to a human error. But software failure is unique. Any representational standard must pay particularly close attention to the special representational demands of software, and the particular ways it fails. A discussion of this exceeds the scope of this article, but interested readers should consult [7].

Presentation—Graphs are viewed graphically; the graphical presentation of a causal network should also be preserved across tools. This presentation information need not be part of node and edge definitions. But somehow the standard should associate with nodes and edges:

- Size, shape, shape orientation (rotation)
- Line and border thickness, dash-pattern, arrowhead size
- Color, transparency, shading
- Text font, font size, centering and justification
- Superscripting, subscripting, bold, italic
- X,Y center-position information for nodes
- X,Y anchor points (vertices) and curvature for edges
- Back-to-Front ordering (what items are 'in front of' other items)
- Sequence ordering (time-ordering nodes' appearance in an animated view)
- Groupings (sets of items that can be selected and moved as a unit)
- Icons (e.g., different .gif images for conditions, functions, mitigations)
- A page, or sheet (specify that some subsection of the overall graph appears in a single view)
- Boxes that enclose multiple nodes locate events w.r.t. equipment or organizations, and emphasize when causal chains cross system boundaries.
- Free text and graphics that can be placed in the graphic view, not associated with particular nodes and edges (e.g., a key for the graphic conventions.)

Certain causal relationships should have standard graphical representation

- Links to nodes distant or in other subgraphs.
- Inhibition, representing 'When A is true, it breaks the causal link from B to C.' (Logically, this can be composed with NOT and AND gates, but a more compact graphic representation is desirable. See [4] for illustration.)
- M^+, M^- and d/dt relations
- Binary or enumerated variables which each variable-value enables a different causal path.
- Higher-order logical operators (xor, nand, k-of-m, iff or arbitrary truth tables)
- Top nodes and basic events

4. Conclusions

Teams in several fields are creating standard interchange formats. These formats will keep modelers from being tethered to a single proprietary tool, will allow autogeneration of causal models, and will allow causal analysis tools to be integrated to auxiliary services. The several teams should consider coordinating their efforts and borrowing the best ideas across disciplines. They should also ensure that the forthcoming standards can encode the most difficult modeling challenges – causal relationships that exceed the scope of any of today's tools. The standards should address the graphical presentation of causal models, so that both machines and people can read the causal knowledge.

5. Acknowledgements

References

[1] Oivind Berg, Svein Nilsen, "The Computerised Procedure System COPMA-III use in Human Factors Experiments." http:// www.ife.no/media/841_Kyoto-copma.pdf

[2] R. Bisio, J. E. Hulsund, S. Nilsen "Brief Introductions to the COPMA-III Tool," http://www.ife.no/media/433_Copma-1.pdf

[3] Dean L. Gano, Vicki E. Lee, Wendy C. Mitchell , *Apollo Root Cause Analysis - A New Way Of Thinking,* Apollonian Pubns, Sept 1999, ISBN 1883677017

[4] EFA Education, National Institute of Health, "Diet-Induced Dyslipemias and Disease" http://efaeducation.nih.gov/sig/chainofevents.ppt

[5] Kenneth D. Forbus "Qualitative process theory", Artificial Intelligence. Volume 24 , Issue 1-3 December 1984, Pages: 85 – 168, ISSN:0004-3702, Elsevier Science Publishers Ltd. Essex, UK

[6] B. J. Kuipers. 1994. Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. Cambridge, MA: MIT Press.

[7] Leveson, Nancy. 1995. Safeware. Addison-Wesley Publishing Company. ISBN 0-201-11972-2.

[8] J. T. Malin, L. Fleming and D. R. Throop, "Hybrid Modeling for Scenario-Based Evaluation of Failure Effects in Advanced Hardware-Software Designs," Model-Based Validation of Intelligence, Technical Report SS-01-04, AAAI Press, Menlo Park, CA, 2001.

[9] J. T. Malin, L. Fleming and D. R. Throop, "Predicting System Accidents with Model Analysis during Hybrid Simulation," Proceedings of Business and Industry Symposium, Advanced Simulation Technologies Conference, Simulation Councils, Inc., pp. 155-160. April 2002.

[10] J. T. Malin, D. R. Throop, L. Fleming and L. Flores, "Computer-Aided Identification of System Vulnerabilities and Safeguards during Conceptual Design," 2004 IEEE Aerospace Conference Proceedings, March 6-13, 2004.

[11] J. T. Malin, D. R. Throop, L. Fleming and L. Flores, "Transforming Functional Requirements and Risk Information into Models for Analysis and Simulation" 2005 IEEE Aerospace Conference Proceedings, March 2005 (forthcoming.)

[12] Microsoft Decision Theory and Adaptive Systems Group, "XML Belief Network File Format" http://www.research.microsoft.com/ research/dtg/bnformat/default.htm

[13] Michael J. Pratt, "Technical Note: Introduction to ISO 10303 – the STEP Standard for Product Data Exchange", http://www.mel.nist.gov/msidlibrary/doc/jcise1.pdf

[14] Kenneth Russel, "The 10 Year Anniversary of SAPHIRE", http://saphire.inel.gov/10_years.cfm

[15] Michael Stamatelatos *et al,* "Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners, Version 1.1", NASA Office of Safety and Mission Assurance, NASA HQ, August 2002.

[16] SysML Partners, "Systems Modeling Language Overview", October 2003, http://neptune.irit.fr/Biblio/03-10-07.pdf, www.SysML-Partners.org.

[17] William Vesely, "Fault tree Handbook with Aerospace Applications, Version 1.1", NASA Office of Safety and Mission Assurance, NASA HQ, August 2002.

[18] European Cooperation for Space Standardization, Space Systems – Risk Management, ISO-17666, www.iso.org

[19] HL7 Reference Information Model (http://www.hl7.org/library/data-model/RIM/modelpage_non.htm)

[20] SNOMED Clinical Terms, College of American Pathologists, Jan 2002

[21] Guideline Interchange Format, (www.openclinical.org/gmm_glif.html)

[22] M Sordo, A Boxwala, O Ogunyemi, R Greenes, "Description and Status Update on GELLO: a Proposed Standardized Object-Oriented Expression Language for Clinical Decision Support," (http://tinyurl.com/6psfg or http://cmbi.bjmu.edu.cn/news/report/2004/medinfo2004/pdf files/papers/5558Sordo.pdf)

Biography

David R. Throop *has been an Artificial Intelligence Specialist with The Boeing Company since 1992. He provides engineering software support in the Intelligent Systems Branch in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center. He oversaw development of FMEA modeling software and its use for the International Space Station. His 1979 Bachelors of Chemical Engineering is from Georgia Tech. His 1992 Ph.D. in Computer Science is from the University of Texas, with a dissertation on Model Based Diagnosis.*

Jane T. Malin *is Senior Technical Assistant in the Intelligent Systems Branch, Automation, Robotics and Simulation Division, Engineering Directorate, NASA Johnson Space Center, where she has led intelligent systems research and development since 1984. She has led development of the CONFIG hybrid simulation tool. She has led research on intelligent user interface and intelligent agents for control of space systems, and on teamwork tools for anomaly response teams. Her 1973 Ph.D. in Experimental Psychology is from the University of Michigan.*

Land D. Fleming *is a Computer Systems Specialist supporting the NASA Johnson Space Center Automation, Robotics, and Simulation Division since 1990. He has been involved in both the development of computer simulation tools and their application to space systems. His 1987 M. S. in Computer Science is from De Paul University.*
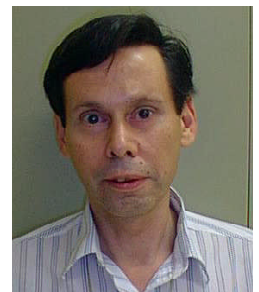
Knowledge Representation Standards and Interchange Formats for Causal Graphs

David R. Throop
The Boeing Company

Jane T. Malin
NASA Johnson Space Center

Land Fleming
Hernandez Engineering

In many domains, automated reasoning tools represent graphs of causally linked events. These include fault-tree analysis, probabilistic risk assessment (PRA), planning, procedures, medical reasoning about disease progression, and functional architectures. Each of these fields has its own requirements for the representation of causation, events, actors and conditions. The representations include ontologies of function and cause, data dictionaries for causal dependency, failure and hazard, and interchange formats between some existing tools. In none of the domains has a generally accepted interchange format emerged. The paper surveys existing practice and emerging interchange formats in each of these fields. Setting forth a set of terms and concepts that are broadly shared across the domains, we examine the several ways in which current practice represents them. Some phenomena are difficult to represent or to analyze in several domains. These include mode transitions, reachability analysis, positive and negative feedback loops, conditions correlated but not causally linked and bimodal probability distributions. We work through examples and contrast the differing methods for addressing them. We detail recent work in knowledge interchange formats for causal trees in aerospace analysis applications in early design, safety and reliability. We generalize the aerospace analysis work across the several other domains. We also recommend features and capabilities for the next generation of causal knowledge representation standards.