Cognitive
Engineering
Laboratory

Integrated Abstraction Hierarchy and Plan-Goal Graph Model for the DURESS II System: A Test Case for Unified System- and Task-based Modeling and Interface Design

Christopher A. Miller
And
Kim J. Vicente

CEL 98-07

Cognitive Engineering Laboratory Department of Mechanical & Industrial Engineering University of Toronto
5 King's College Rd. Toronto, Ontario, Canada M5S 3G8
Phone: +1 (416) 978-7399 Fax: +1 (416) 978-3453
Email: benfica@mie.utoronto.ca URL: www.mie.utoronto.ca/labs/cel/

Cognitive
Engineering
Laboratory

Director: Kim J. Vicente, B.A.Sc., M.S., Ph.D.

The Cognitive Engineering Laboratory (CEL) at the University of Toronto (U of T) is located in the Department of Mechanical & Industrial Engineering, and is one of three laboratories that comprise the U of T Human Factors Research Group. CEL began in 1992 and is primarily concerned with conducting basic and applied research on how to introduce information technology into complex work environments, with a particular emphasis on power plant control rooms. Professor Vicente's areas of expertise include advanced interface design principles, the study of expertise, and cognitive work analysis. Thus, the general mission of CEL is to conduct principled investigations of the impact of information technology on human work so as to develop research findings that are both relevant and useful to industries in which such issues arise.

Current CEL Research Topics

CEL has been funded by Atomic Energy Control Board of Canada, AECL Research, Alias|Wavefront, Asea Brown Boveri Corporate Research - Heidelberg, Defense and Civil Institute for Environmental Medicine, Honeywell Technology Center, Japan Atomic Energy Research Institute, Natural Sciences and Engineering Research Council of Canada, Nova Chemicals, Rotoflex International, and Westinghouse Science & Technology Center. CEL also has collaborations and close contacts with the Mitsubishi Heavy Industries and Toshiba Nuclear Energy Laboratory. Recent CEL projects include:

- Studying the interaction between interface design and adaptation in process control systems.
- Understanding control strategy differences between people of various levels of expertise within the context of process control systems.
- Developing safer and more efficient interfaces for computer-based medical devices.
- Designing novel computer interfaces to display the status of aircraft engineering systems.
- Developing and evaluating advanced user interfaces (in particular, transparent UI tools) for 3-D modelling, animation and painting systems.

CEL Technical Reports

For more information about CEL, CEL technical reports, or graduate school at the University of Toronto, please contact Dr. Kim J. Vicente at the address printed on the front of this technical report.

Unified Modeling Project

UT/NCL/HTC/NSERC

Integrated Abstraction Hierarchy and Plan-Goal Graph Model for the DURESS II System; A Test Case for Unified System- and Task-based Modeling and Interface Design

An Interim Report of work under Task 5: "Develop Model Integration Approach" of the NOVA/UofT Task Breakdown (Jan. 13, 1998)

Release Date: 18 March, 1998
Document Version: 1.01
Filename: PGG-DURESS.doc

Submitted to:
Jamie Errington, NOVA Chemicals, Ltd.

Prepared by: Chris Miller and Kim Vicente
Honeywell Technology Center &
University of Toronto Cognitive Engineering Laboratory

1. Document History

Versions .01, .02 and ..03 were internal, working drafts developed at the University of Toronto.

Version 1.00 is a draft circulated for review and comment to NOVA and HTC.
Version 1.01 incorporates minor suggested revisions from Jamie Errington (NOVA) and Peter Bullemer (HTC) and adds the acknowledgements section. This version was released as a UT Cognitive Engineering Lab Technical Report.

2. Summary

As a test case for the development and use of a unified task- and system- (or work domain)based modeling and interface design approach, we have been looking at Dr. Kim Vicente's DURESS II simulation. DURESS II (Vicente, 1996) is a thermal-hydraulic process system simulation, which has been extensively modeled using Abstraction Hierarchy (AH) techniques. DURESS II is a relatively small, yet representative, system and there is substantial expertise in it here in Kim's lab. Furthermore, a diverse set of interfaces have already been designed for it, many based on the work-domain analysis methodology, but a task-based approach has not yet been applied. Thus, it serves as a good domain for developing and testing initial theories and techniques before moving with them to the more realistic domains which NOVA offers.

The work described in this report represents the first time we have attempted to model a system using both a system-based technique (AH) and a task-based technique. The task-based technique we chose to use was a variation on the Plan-Goal Graph (PGG) methodology developed by Norm Geddes (Geddes, 1989; Sewell and Geddes, 1990). PGG was chosen because we suspected that its decomposition of the tasks in the domain into a hierarchy of causally related plans and goals either paralleled, or could be made to parallel the functional decomposition hierarchy which the AH provides. Overall, the attempt to merge these two analytic techniques was done primarily as an exercise to investigate integrative modeling techniques and to examine what potential strengths and weaknesses one such technique offered.

On the one hand, the construction of a PGG for DURESS II proved surprisingly easy to do—undoubtedly because the plan-goal relationships which are explicitly modeled in PGG are closely related to the means-ends relations which are modeled in AH. The AH model provided a more rigorous, system-valid basis for building a PGG than has frequently been available in the past. I n contrast, the PGG provided a suggestion of alternate, task-based ways to organize information in an interface. On the other hand, the benefits provided by the PGG model might be less than optimal. We may be missing some of the power of task-modeling techniques to compliment AH by representing known procedures or task sequences for accomplishing goals, since these are not modeled explicitly in PGG. Nevertheless, the PGG model is useful as it is, and could be made much more so in the context of additional information or dynamic software support for operator intent tracking and/or system simulation—both of which have been used with PGG in the past.

3. The Modeling Technique

I have used a modified form of Geddes (1989) and Sewell and Geddes (1990) Plan/Goal Graph (PGG) modeling technique. This technique is not, as far as I can tell, well documented in the literature. It shares common theoretical underpinnings with other goal-oriented modeling techniques—especially GOMS (Kieras, 1988). PGG was used extensively in the Lockheed's Pilot's Associate program where it was used to construct the central plan representation which served as a 'map' to coordinate all of the smart subsystems with each other and with the pilot. Thus, AI planners would recommend plans in the language of the PGG (that is, they would recommend a 'plan' which the PGG represented), the Intent Inferencing component would observe pilot behavior and the situation and attempt to map them into the known tasks represented in the PGG, the interface management component would configure cockpit information displays and some automation behavior on the basis of the set of plans it was told were or should be active, etc.

Vocabulary note: I will provide a more detailed set of definitions below. For the moment, however, in order to prevent confusion, I will note that the "plans" in the PGG are more commonly referred to as "tasks" in the rest of the task modeling and task analysis literature. For this reason, I will use the term "task" in this document even though the PGG modeling technique would refer to these entities as "plans".

3.1 Why PGG?

This representation technique has a number of significant advantages, especially if our goal is the integration with an Abstraction Hierarchy (AH) representation, as well as a few disadvantages. The advantages include:

1. It supports an explicit distinction between tasks (plans) and goals. Many other task representations blur this distinction, either representing only one or the other, or more commonly, folding them together by assuming that the accomplishment of the task <u>is</u> the goal. Separating goals out provides more flexibility and more stability in the representation (since goals are generally more stable than the methods of achieving them, representing goals increases a model's stability while simultaneously facilitating folding in new methods when they arise), the ability to maintain the tracking of an operator's activities even when they do not correspond to any known task method, the ability to recognize serendipitous circumstances, etc. Of critical importance for the purpose of integrating with an AH representation, we believe that goals are the point of contact between human intentional behavior and system and equipment design and capabilities—the operator intends to achieve a certain goal and the machine, when operated in a certain fashion, was designed with the capabilities of achieving that goal.
2. In fact, the goal representation used in PGG is richer than that in most other task representations because it is repeated at each level of the hierarchy. Every task in PGG must be immediately preceded by a goal at the next higher level; every goal must have at least one task for accomplishing it at the level immediately below. This interleaving of goals with tasks that accomplish them (but only do so through the posing of new goals), repeats the power of an explicit goal representation (as described in point 1 above) at each level of the hierarchy.
3. As with many task representations, it is hierarchical. This fact combined with the explicit Goal representation, offers the potential for a natural mapping between a PGG and the means-ends dimension of an AH. The PGG explicitly embodies the notion that tasks are means to accomplish higher level goals, while imposing subgoals of their own which must have additional sub-tasks as the means to accomplish them.

Disadvantages include:

1. The explicit separation of goals and tasks means that building a PGG may entail more work than some other task representation techniques. The interleaving of task and goal levels exacerbates this.
2. While the PGG does embody a sort of means-ends decomposition, this decomposition differs in two important ways from the means-ends decomposition of the Abstraction Hierarchy. First, it is a decomposition of actions, not of the work domain structure itself. Second, the means and ends represented have rarely been tied explicitly to the capabilities and constraints of the equipment and work domain. Instead, they have been tied to the operator's mental model of tasks in his or her work domain. Whether or not a PGG can be usefully constructed on the basis of system-centered means and ends is an open question.
3. The PGG does not represent some of the task knowledge which is commonly represented in other task representations. In this sense, it may be more similar than complimentary to the AH to provide maximum benefit. In most cases, there are ways of including this information in a PGG model, but these ways are generally not as thorough or not as convenient as other methods make them. Among the task knowledge missing or poorly represented in PGG is:
    - sequential dependencies among tasks
    - known strings of tasks which compose a "procedure"
    - Conditions under the task/method is applicable

- Personal or organizations preferences for one among multiple alternate methods when these do not have a sound basis in the goals of the domain.

Ultimately, the PGG was designed and, to date, has been primarily used (cf. Geddes, 1989; Hoshstrasser and Geddes, 1989; Webb, Geddes and Neste, 1989, Geddes, 1994; Geddes, et al., 1996 to serve as the 'map' of the possible activities available in a work domain against which an intent inferencing system evaluates observed operator actions in an attempt to determine what goals the operator is (or, to some extent, should be) currently pursuing. Thus, it lays out ways in which actions can be strung together to achieve progressively higher and more abstract work goals—i.e., action capabilities supported by the work domain. Since its intent is to support *tracking* operator activities, it is comparatively weak (compared to some other task representations) at recommending or enforcing known ways of doing things (e.g., procedures and strategies). At the end of this paper, we will discuss some ways of augmenting the PGG to serve these ends and some ways of using the PGG representation without these additions, along with our review of the strengths and weaknesses of the approach for the purposes of integration with a work domain model such as Rasumusen's Abstraction Hierarchy for the goal of improved interface design and/or management.

3.2 PGG Entities and Representation Formalism

Below are the terms, representational formalisms, and modeling heuristics I intend to use for this analysis.

Goal = A desired state of the world. Goals must be described as states, not actions. Inevitably, these will be partial state descriptions. They are represented by ovals.

Task = A method of accomplishing a goal. Tasks are actions, not states. They are represented by rectangles, except when they are actions (see below). A 'pure' use of the PGG technique would label these "plans".

Action = The lowest level of task represented in the hierarchy. A typical stopping criteria for PGG is to have actions be reliably operational—that is, where the actor needs no further information to guide him/her/it to successful performance of the task. Actions are represented by triangles.

Plans = Equivalent to "Tasks" as defined above in a 'pure' use of the PGG technique. I will use this term instead to refer to sequential information about the relationship of multiple tasks. I believe this usage is more common in the task modeling literature. When multiple tasks must be strung together to accomplish a goal, the instructions/rules/heuristics about how to string them is a plan, by this definition. Thus, plans are not explicitly represented in this PGG model (though they could probably be added).

Rule 1: Goals and tasks are both hierarchically structured and interleaved. Goals decompose into subgoals. Tasks decompose into subtasks.

Rule 2: Although you can view either the goal hierarchy or the task hierarchy independently, the full graph should be built with them interleaved. Goals are accomplished by one or more tasks. Accomplishing tasks generally means asserting new subgoals which must be met by subtasks, etc.

Rule 3: Subgoals used to decompose a task answer the question 'What does it <u>mean</u> to accomplish the task acceptably?' Thus, they are a description of the state which will hold if the task is successful. For example, a high-level DURESS II task is "Maintain Reservoir Temperatures" (note that this is a task because it is an action which some actor must perform). The subgoals which decompose this task are "Min 1 < Reservoir 1

Temperature < Max 1" and "Min 2 < Reservoir 2 < Max 2"—that is, being in a state where temperatures in reservoirs 1 and 2 is between the minimum and maximum acceptable values for each reservoir. These subgoals answer the question "What does it mean to 'Maintain Reservoir Temperatures'?"

Rule 4: Subtasks used to decompose a parent goal answer the question "How can you achieve the parent goal?" Thus, they are a description of the various methods known to work (perhaps only in certain contexts) to accomplish the goal.

Although full plan information is not represented in the PGG, the following relationships are shown:

AND = AND relationships mean that all of the child nodes must be satisfied in order for the parent node to be accomplished. These are represented by a solid line between branches.

OR = OR relationships mean that one or more of the child nodes must be satisfied in order for the parent node to be accomplished in different circumstances. These are represented by the absence of a line between branches.

XOR = XOR relationships mean that one and only one of the child nodes can be satisfied in order for the parent node to be accomplished. These are represented by the a solid line with a cross hatch between branches.

These entities and relationships are illustrated in Figure 1, along with a sample PGG to illustrate the modeling formalism.

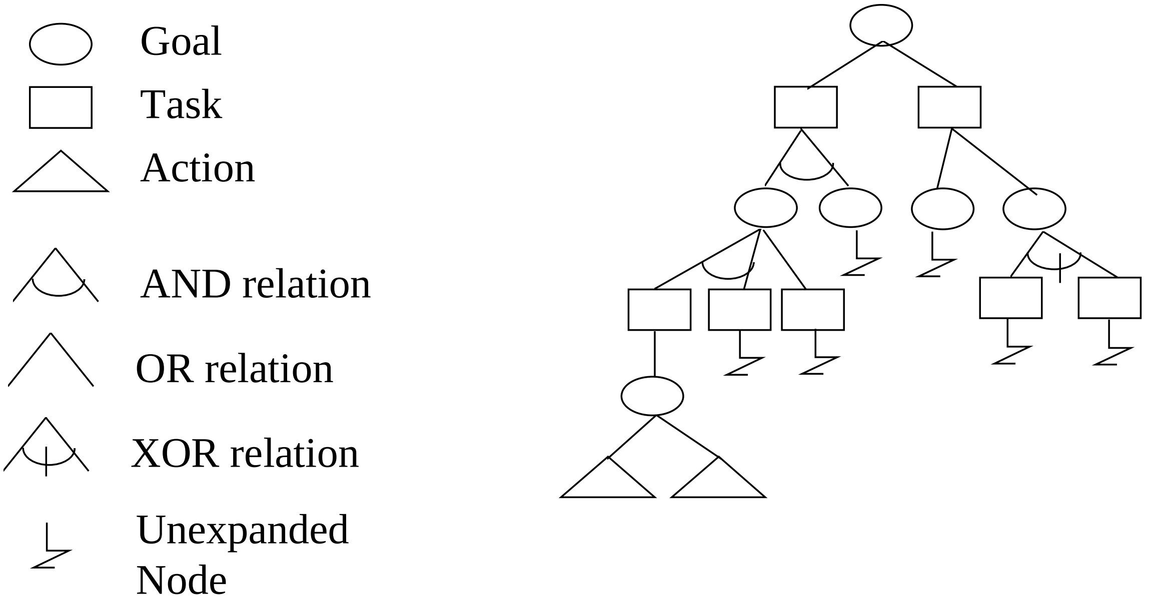

Figure 1. Representational formalism for PGG.

4. The Analysis

4.1 DURESS II Description

The following description is from Vicente, forthcoming:

DURESS (DUal REservoir System Simulation) II is a thermal-hydraulic process control microworld that was designed to be representative of industrial process control systems, thereby promoting generalizability of research results to operational settings. The physical structure of DURESS II consists of two redundant feedwater streams (FWSs) that can be configured to supply water to either, both, or neither of two reservoirs. Each reservoir has associated with it an externally determined demand for water that can change over time. The work domain purposes are twofold: to keep each of the reservoirs at a prescribed temperature (40° C and 20° C), and to satisfy the current mass (water) output demand rates. To accomplish these goals, workers have control over eight valves (VA, VA1, VA2, VO1, VB, VB1, VB2, and VO2), two pumps (PA and PB), and two heaters (HTR1 and HTR2). All of these components are governed by first order lag dynamics, with a time constant of 15 seconds for the heaters and 5 seconds for the remaining components. DURESS II is described in more detail in Vicente, 1996.

4.2 AH for DURESS II Description

Abstraction Hierarchies for the DURESS II system have been completed multiple times and are, in fact, used as a training exercise in classes taught at the University of Toronto. One account of the analysis and the resulting interface design can be found in Vicente and Rasmussen, 1990. Vicente (forthcoming) includes a didactic chapter on how to use the AH to perform work domain analyses and works through a detailed analysis of DURESS II. Finally, a Cognitive Engineering Lab Technical Report (CEL-95-08—Hunter, Janzen and Vicente, 1995) includes detailed training materials for the DURESS II AH. These were the primary materials I used in performing the analysis below.

While we frequently, and casually, talk of the AH analysis of DURESS II, it would be more appropriate to refer to it as an abstraction-decomposition analysis. This is because the analysis typically proceeds along two dimensions: the abstraction dimension which traces means-ends links and the decomposition dimension which traces part-whole links. The typical abstraction-decomposition analysis for DURESS II consists of 3 layers in the part-whole dimension corresponding to the system as a whole, decomposition of the system into subsystems, and further decomposition into individual components. The analysis also contains 5 layers in the means-ends dimension corresponding, from higher to lower levels of abstraction, to Functional Purpose (the overall goals of DURESS II operation), Abstract Function (mass and energy balances), Generalized Function (heat and water flows), Physical Function (component values and settings), and Physical Form (the actual appearance and location of DURESS II components).

These two dimensions form a matrix as illustrated in Figure 2 and the typical abstraction-decomposition analysis of the DURESS II work domain examines models of the domain only within certain cells of that matrix—those on or near the diagonal (illustrated by shading in Figure 2). The reasons for this will be described in more detail below. Within each cell of the matrix, however, separate but linked models of the DURESS II system are constructed. For example, within the Functional Purpose x System cell, a simple “model” of the purposes of the entire system is constructed: to maintain output temperature and flow goals. This model is linked, however to all the other models, however—thus, for example, using the analysis we can trace how the work domain is constructed to enable the achievement of these purposes by the generalized functions of individual components.

	System	Sub-system	Component
Functional Purpose	▨		
Abstract Function		▨	
Generalized Function		▨	▨
Physical Function			▨
Physical Form			▨

Figure 2.. Dimensions of the abstraction-decomposition analysis of DURESS II. Shaded cells are those for which models are typically constructed.

4.3 PGG for DURESS II

When I undertook the effort to construct a PGG for DURESS II, my intuition was that the Means-Ends decomposition axis of the DURESS II AH would provide a convenient structure for informing the Task-Goal decomposition of the PGG. Thus, I strove to stay as close as reasonably possible to the DURESS II AH decomposition, while at the same time maintain the actor/task orientation of the PGG. As will be discussed below, this proved both very convenient and, for the most part, a powerful structuring guide for the PGG analysis. The results of this effort are shown in Figures 3-6: a partial PGG for DURESS II. I have attempted to identify the mapping between the means-ends layers of the DURESS II AH and the levels of the PGG along the side of each figure.

While the part-whole decomposition used in a typical Abstraction Hierarchy analysis is also important and useful, typical AH analyses "walk the diagonal" of the abstraction-decomposition matrix. This means that they tend to analyze higher abstraction levels in terms of more aggregated views of the system on the part-whole dimension (e.g., functional purposes analyzed at the whole system level), and lower abstraction levels in terms of less aggregated views of the system (e.g., physical function at the component level). While it is entirely possible to perform analyses in other cells in the abstraction-decomposition matrix (e.g, functional purpose of components), these cells are frequently skipped as providing less benefit than those which lie along the diagonal. In constructing a means-ends analysis of actions and goals available in the work domain, I found it useful to 'walk the diagonal' too. That is, the highest levels in the PGG analysis correspond to system-wide goals or purposes, while the lowest levels correspond to achieving and managing specific functional behaviors from physical components.

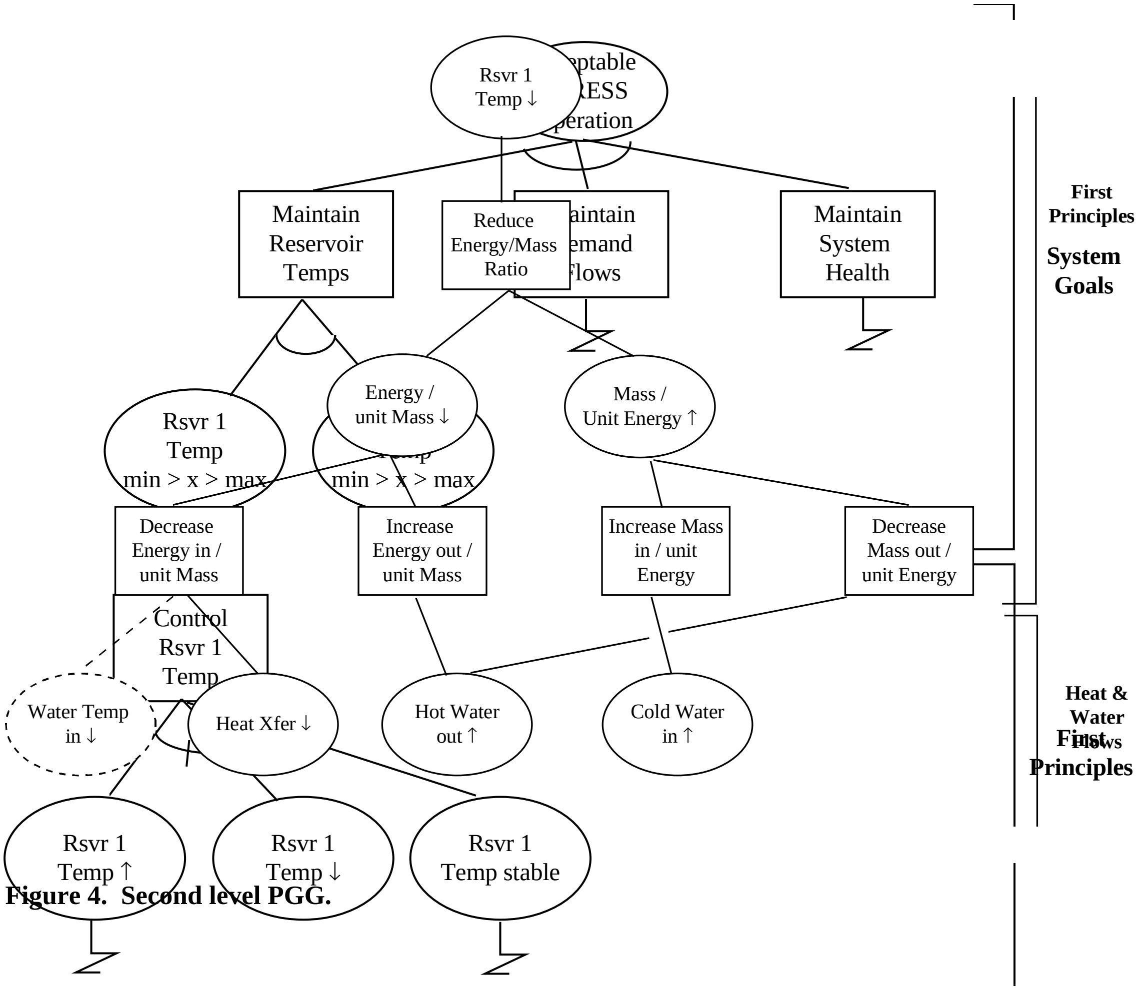

Figure 4. Second level PGG.

Figure 3.. Top level DURESS II PGG.

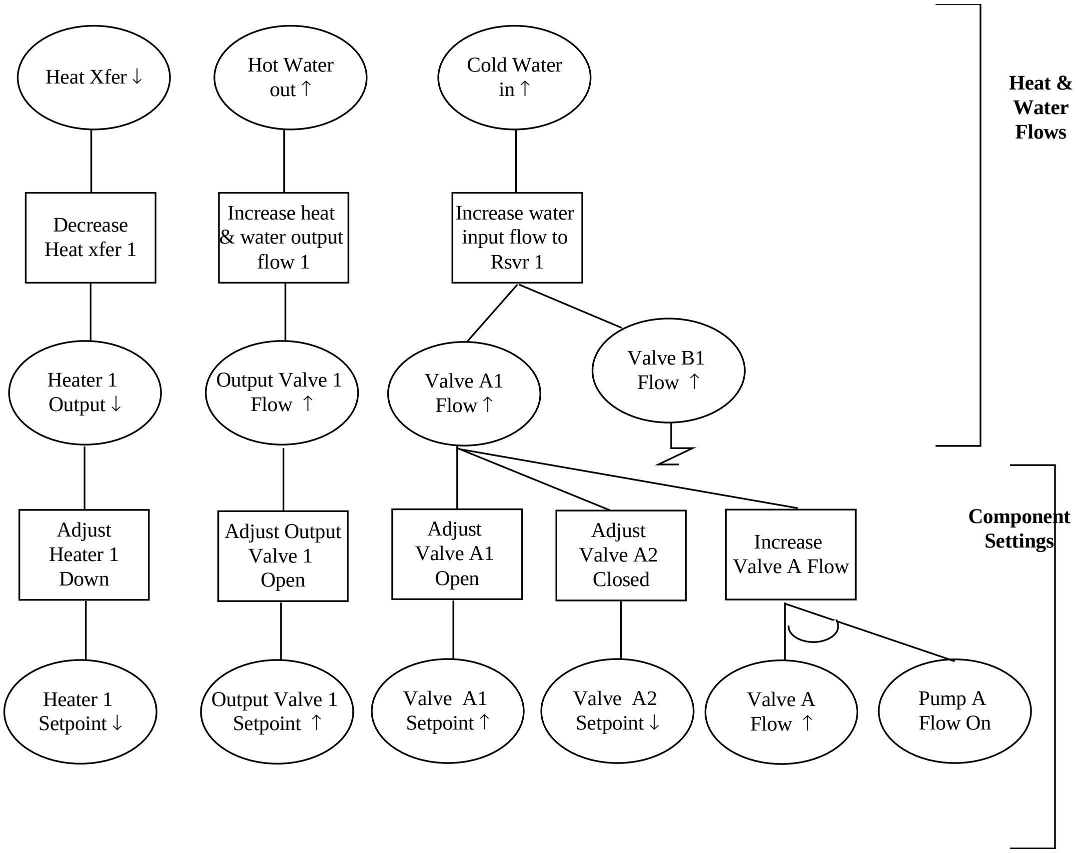

Figure 6. Third level PGG.

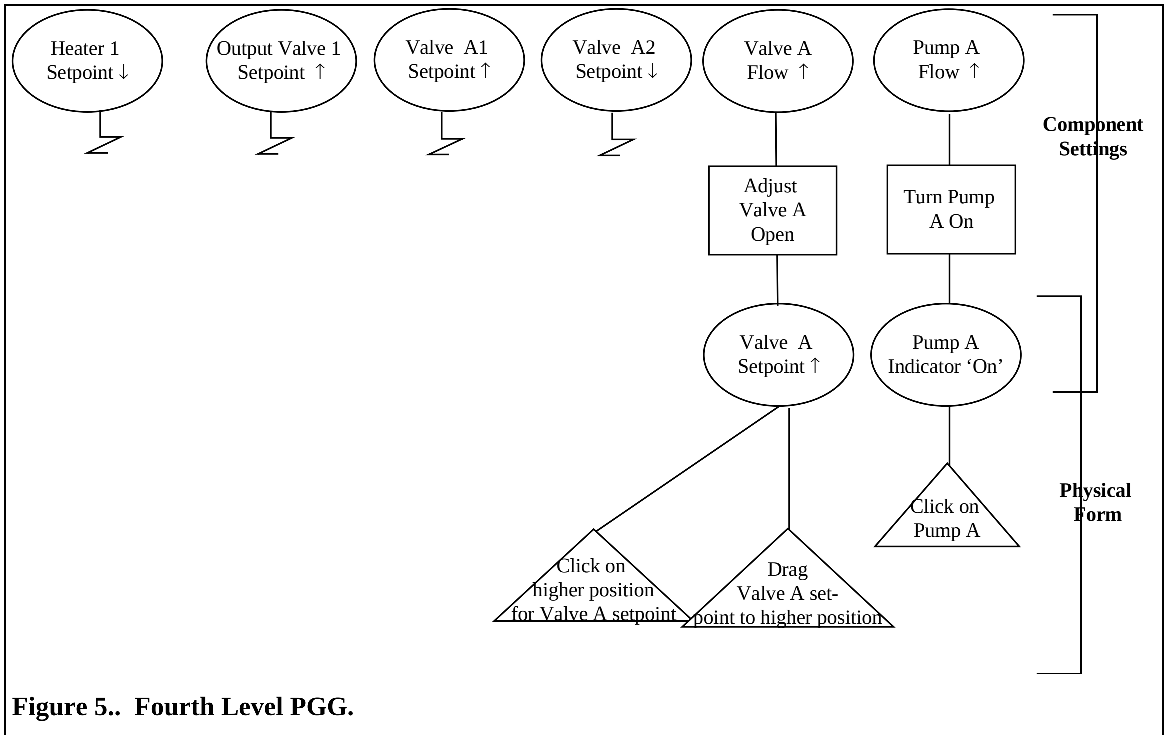

Figure 5.. Fourth Level PGG.

4.4 Notes on Performing the PGG analysis for DURESS II

In this section, I report observations made during the PGG analysis.

1. The top level goal in a PGG is usually unary. I don't believe this is a requirement, but it is very typically something equivalent to 'Be successful in this work'.
2. The top level 'goals' in the AH decomposition of DURESS II, I have listed as tasks in this analysis. This is in keeping with the definition of goals as states and tasks as actions. I don't believe there is anything particularly deep about this—just differences in semantics. Notice that the system state description which defines success for this task is provided by moving one level down in the PGG.
3. Various information is typically stored with each task or goal in the PGG depending on the use to which the model is to be put. Typical types of information are acceptability conditions for each node (a description of the conditions under which this goal or task is appropriate or useful), information and/or resource requirements (given that the operator is or should be involved in this task or goal, what information or resources does s/he need?), duration information, etc.
4. It is also typical for a computer implementation of the model to maintain links, dependencies and constraints between all nodes (which gets very messy on paper), thus while there are implications for the overall safety goal in how far you can reduce the level in a reservoir, and these end up imposing constraints on the acceptability conditions of the task/method "Increase heat & water output flow 1", these constraints are not, and probably cannot be, modeled in a simple graphical presentation. What the graphical model would show, if fully expanded, is that there are "avoidance goals" underneath "Maintain System Health" which might be threatened if certain combinations of methods of achieving the other system goals are attempted. This is not a particularly convenient representation for a priori identification of hazard states, but it works reasonably well for computer monitoring and plan recommendation, as well as doing a reasonable job of reflecting the causal/functional reasons for not doing some things which are possible within the system.
5. Information requirements are typically associated with tasks and not with goals, though we have thought about doing both. Goals might also be a good place to put general preference information—otherwise whether to satisfy one goal versus another in an OR or XOR relationship will be entirely dependent on the successfulness of the lower level tasks in context.
6. In attempting to closely follow the AH means-ends decomposition, I found myself introducing levels into the PGG that I probably would not have done otherwise. For example, I was tempted to jump immediately from "Control Rsvr1 Temp" to methods of controlling flows and heaters. That is because I feel that that representation is more 'natural' or user-centered. Of course, I may feel that way because I'm the closest thing I've got to a user and I do not typically or naturally think about this domain in terms of mass and energy balances. I think this illustrates the power of building the PGG from a work domain model focused on the system first. If mass and energy balances are the 'right' or most appropriate way to think about the domain, then by building the PGG around them, I think I come up with a better way to steer the user in how to use those concepts. By contrast, if I just did the PGG on the basis of user interviews (which is typical), I probably wouldn't come up with the mass/energy balance concepts, and if I just built an AH on the basis of knowledge about the work domain, I might not provide a good fit for those concepts into the user's goals and operations within the domain.
7. On the other hand, I think I came across a place where the operator task-focus imposed by the PGG might actually offer an improvement over the AH. The DURESS II training manual I was working from shows the part-whole decomposition of DURESS II as representing the feedwater subsystems as physically collocated: Pump A, Valve A, Valve A1 and Valve A2 comprise one such subsystem, and Pump B, Valve B, Valve B1 and Valve B2, the other. My analysis indicates that, at least for the tasks I modeled, the operator is more interested in that set of components which effects the goal of getting feedwater into each reservoir, than s/he is in the physical collocation of components. This would argue for defining the feedwater subsystems as Pump A, Valve A, Valve A1 and Valve B1 (representing the set of components which control flow into reservoir 1) and Pump B, Valve B, Valve A2 and Valve B2 (which control flow into reservoir 2). Other task organizations are possible and might be useful for different tasks I did not model in my analysis. The point is, though, that having the tasks and goals represented provides us with a means of selecting the most appropriate organization for the current goal focus. This organization would necessarily be dynamic and based on the needs of the task being

performed rather than solely on the physical or functional structure of the work domain itself—but if such organization could help the operator perform some tasks, it would be worth making it available.

8. Figure 4 provides a small example of the power of a functional decomposition. The goal node in dashed lines ("Water Temp In ↓") illustrates a possible goal which is not supported by the system. If the system provided access to a method for reducing the temperature of the feedwater, it would be an alternate method of performing the task "Decrease Energy in / Unit Mass". Thus, we know exactly where to put it in our model and what parent goals and tasks it will affect. Furthermore, we know that the tasks/methods required to accomplish it will possibly occur at the same time as reducing heater 1 output, but will be detrimental to the goal "Rsvr1 Temp ↓", etc.
9. The linking of two parent tasks to the same sub goal in Figure 4is primarily done for the ease of the model creator, but it also illustrates the symmetry between the two tasks. Raising the rate at which hot water leaves the reservoir will accomplish both increasing the energy out per unit mass and decreasing the mass out per unit energy. PGG permits multiple parents for a goal, task or action and thus, is a directed acyclic graph rather than a true tree structure hierarchy.
10. In Figure 5, an example of an acceptability criterion for the task "Increase heat and water output flow 1" *in this task and goal context* might be "if the water in reservoir 1 is hotter than the feed water and if the reservoir is not at capacity." This illustrates the utility of having multiple instances of the nodes which occur in different task and goal contexts in the PGG.
11. Plans, although not explicitly represented in the PGG, may be viewed as a sequenced traversal through the tasks in the PGG. Thus one plan for "Increase Water Flow to Rsvr 1" might be "Adjust Valve A1 open by clicking and dragging the setpoint higher and then ensuring that Pump A is on by clicking on the pump."
12. The strategy/preference for obtaining appropriate mass balances before addressing temperature balances might be encoded in the highest level task nodes in Figure 3. Perhaps more useful would be to develop a plan representation which describes the sequence in which tasks should/must be done and store this separately from the PGG representation itself.
13. There are very few action nodes in the PGG because I have had little interaction with DURESS II as yet. These nodes would (potentially) be different depending on the interface used, however, though the higher level goals and tasks in the system might not change.
14. Hmm . . . It is interesting that I have no monitoring or decision making tasks in the PGG. I suspect this is an artifact of using the AH as a starting point. How, for example, is the operator to know when s/he should raise, lower or keep stable the reservoir temperatures? This could be built into the acceptability conditions for each of those tasks, but it might be more appropriate (and more in keeping with leaving the design goal of leaving the operator in charge) to model an explicit monitoring and decision making task.

5. Utility

In the NSERC proposal, we said that, generally, the utility of task representations was that:

1. they enabled context-sensitive (in fact, task-sensitive) information management--that is, enables taking advantage of task constraints to prioritize and filter information,
2. They enable task tracking for automated information management, and/or task reminding/critiquing,
3. They can enable task recommendation (a good next course of action), task reminding (a task which should have been accomplished given context, but was not), and task critiquing (given knowledge about task methods, and knowledge of current intent and/or context, critique current operator performance),
4. They can embody direction-like aspects of task knowledge—that is, convey known ways of achieving goals to operators.

The simple PGG representation used above is somewhat aberrant as a task modeling technique precisely because it sacrifices the representation of known procedures for a representation of general capabilities. In this sense, it is perhaps *too* much like the Abstraction Hierarchy to be complimentary to its strengths and weaknesses and, therefore, to provide maximal utility. While specific procedures can be linked to the PGG (indeed, they must be, if it is to be used to track tasks or serve intent inferencing purposes), this is an optional extension to the technique which we did not pursue above. Future work should explore the benefits from adding such procedural knowledge..

On the other hand, the PGG has a number of advantages. First is its fundamental compatibility with AH; the fact that both employ a means-ends hierarchy (albeit, parallel ones) makes it extremely easy to build a PGG from an AH. Second is its emphasis on goal representations. The fact that goals are represented explicitly and repeatedly at each level of the hierarchy makes it easy to link the relationships between equipment constraints and behaviors and the intentional activities of human operators in directing or managing equipment in the performance of those behaviors. Third, PGG's demonstrated use in systems where a primary design goal was for the human to remain in charge (e.g., Pilot's Associate) provides both a body of experience and a philosophy which is compatible with that of CWA to draw on in attempting to design interfaces which make use of both task- and work-domain models.

I will now examine the PGG representation's ability to support each of the task representation advantages listed above.

1. *Can PGG be used to support task-sensitive information management?* Certainly (but not by itself, see point 2 below). It has done so extensively on the Pilot's Associate project. Furthermore, the multi-tiered aspect of the PGG hierarchy provides a highly flexible structure on which to "hang" information needs. While we have typically associated information requirements with the leaf nodes in the hierarchy, this needn't be the case. For example, while many task representations would be completely lost (and therefore, would provide no information management assistance) when a user fails to follow defined procedures exactly or, worse yet, performs a novel task, the PGG retains the ability to represent the higher level tasks and goals which the user is engaged in and, therefore, to provide less specific but still useful assistance at those levels.

On the other hand, the representation itself (or at least those parts of it which I have worked through in this paper) cannot perform this function. The PGG serves as a roadmap; some other technique or system must be used to tell an information management system where on that roadmap the user currently is. Such techniques will be discussed in the next section below.

2. *Can the PGG be used to support task tracking or intent inferencing? Can it be used for task reminding or task critiquing?* Again, certainly. It was used for his purpose on the Pilot's Associate and Rotorcraft Pilot's Associate projects. As mentioned above, since the PGG lays out a "roadmap" of possible actions the pilot might take and links them to goals and higher order tasks, it provides a natural and powerful method of infering operator intent based on observed actions. It also facilitates constraining intent recognition in a top-down fashion; if the operator's high-level goals are known (as is frequently the case), then this knowledge can be used to make the interpretation of low level actions easier. Two points should be made, however. First, intent inferencing software is a separate system which operates on the PGG; the PGG does not provide this capability in and of itself. Second, in order to map observable operator actions to higher level goals and tasks, a library of task schemas is typically used (see point 4 below). While this is not theoretically necessary, it may be practically necessary in order to obtain satisfactory speed and accuracy in an intent recognition system. These task schemata are, again, not included in the basic PGG representation as outlined above.

One thing the PGG by itself could be useful for is the construction of a task selection mechanism. This is the simplest (one might claim, "degenerate") form of intent inferencing device—a method for the operator to explicitly declare his or her intent to the system in order obtain system support in that intention. Since the PGG decomposes operator goals into a hierarchy of their own, the operator could use this hierarchy to declare intent at any level—e.g., Lower Reservoir 1 Temp vs. Increase Cold Water In to Rsrv 1 vs. Adjust Valve A Open. S/he could then receive information or automation support in keeping with that intent and that level of specification—e.g., general information and access (though perhaps not immediate access) to controls if "Lower Rsvr 1 temp" is the intent; much more specific information and controls for Valve A if “Adjust Valve A Open” is the intent. This is, at root, the approach we are taking in our Tasking Interface for Unmanned Air Vehicles work at Honeywell.

Sewell and Geddes (1990) discuss the utility of the PGG by itself to support the design of an information retrieval system. Their points, derived from a design project for an information retrieval and analysis system, are that (a) the PGG provides a list of the functionality which must be included in the system in order to achieve operator goals, and (b) that the PGG itself serves as a reasonable and psychologically valid decomposition of this functionality and thus, that the PGG provides added information about desirable groupings of functionality and, therefore could be used as the basis for a menu driven interface to the system.

3. *Can the PGG representation be used to support task recommendation, reminding or critiquing?* With regards to task reminding or critiquing, the PGG is, again, of only marginal use by itself. Since it is intended to represent task capabilities and to serve as a ‘roadmap’ for recognizing operator actions and mapping them into known goals and intents, substantial additional mechanisms are required to transition from a recognition of what the operator is attempting to do to an ability to recommend what s/he should be doing. On the other hand, the ‘roadmap’ which the PGG provides is a powerful and useful one to use for these purposes.

4. *Can the PGG representation be used to convey known ways of achieving goals to operators?* I believe the answer to this question is ‘by itself, only with slightly added value beyond that which the AH provides by itself.’ Just as AH shows what components and subsystems may be manipulated in order to achieve higher level effects, so the PGG can be used to show what task activities (i.e., methods) *may* be combined in order to achieve a given goal. I believe that there is a small advantage to the task- or method-based information conveyed by the PGG over the strictly functional, system-based information conveyed by the AH. This can be seen in the above example by the inclusion of directional information in the goal states and tasks included in the PGG over and above the simple functional relational information conveyed by the representational form of the AH. Using the PGG (at least the one I have created above), one might more readily realize the type of manipulation required to achieve middle level goals. For example, in order to adjust the temperature of the water in Reservoir 1 down, one might want to increase the setpoint of Valve A1. Using the AH representation alone (as opposed to the equations which underlie the DURESS II simulation, or by study and experience using the DURESS II system), one can most readily realize only that manipulating Valve A1 can have an effect on the energy in Reservoir 1.

Beyond that, more explicit aiding in the conducting of activities to achieve goals in DURESS II can only be obtained through one or both of two methods: (1) the use of more traditional task or procedure representation techniques, or (2) the inclusion of a sophisticated, probably simulation-based decision-aiding system which would work in parallel with the PGG. The first approach would use traditional task analysis and representation techniques to capture, represent and then communicate known strategies for operating DURESS II. These could range from (or perhaps, could combine) highly specific methods for opening and closing valves, to more general techniques for raising or lowering temperature or flow, to general overall strategies such as achieving appropriate flow ratess before achieving appropriate temperatures. The second approach would use either rules or, ideally, a simulation to actively track system state and dynamically provide either specific recommended actions (“open valve A1 to 7”) or, probably more usefully, dynamically “close” task options in the PGG which are inappropriate in the current circumstances (e.g., not presenting “Open Valve A1” as an option for increasing cold water flow in until there is adequate flow from valve A). Either approach would involve substantial augmentation to the existing PGG.

6. Acknowledgements

This work was supported by a research grant from NOVA Chemicals Ltd. to the University of Toronto, and by internal research and development funds from the Honeywell Technology Center. The authors would like to thank Jamie Errington of NOVA and Peter Bullemer and Ian Nimmo of Honeywell for their support and feedback on earlier versions of this document.

7. References

Geddes, N. (1989). *Understanding human operators' intentions in complex systems*. Ph.D. Dissertation, Georgia Institute of Technology.

Geddes, N.D. (1994) A model for intent interpretation for multiple agents with conflicts. *Proceedings of the 1994 IEEE International Conference on Systems Man and Cybernetics*. San Antonio.

Geddes, N.D., Brown, J.L., Beamon, W.S. & Wise, J.A. (1996) A shared model of intentions for Free Flight. *Proceedings of the 14th IEEE/AIAA Digital Avionics Systems Conference*. Atlanta. October 1996.

Hoshstrasser, B. and Geddes, N. OPAL: Operator intent inferencing for intelligent operator support systems. In V. Shalin and G. Boy (Eds.) *Proceedings of the IJCAI-89 Workshop on Integrated Human-Machine Intelligence in Aerospace Systems*, 53-70.

Hunter, C., Janzen, M. and Vicente, K. (1995). *Research on Factors Influencing Human Cognitive Behavior II*. Cognitive Engineering Laboratory, University of Toronto. Technical Report CEL-95-08.

Kieras, D. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.) *Handbook of human computer interaction*. pp. 135-157.

Sewell, D. and Geddes, N. (1990). A plan and goal based method for computer-human system design. In D. Diaper, et. al. (Eds). Human-Computer Interaction—INTERACT '90. Elsevier Science Publishers; North-Holland. pp. 283-288.

Vicente, (forthcoming). To be titled book on Cognitive Work Analysis, in progress, Spring, 1998.

Vicente, K. J. (1996). Improving dynamic decision making in complex systems through ecological interface design: A research overview. System Dynamics Review, 12, 251-279.

Vicente, K. and Rasmussen, J. (1990). The ecology of human-machine systems II: Mediating 'direct perception' in complex work domains. *Ecological Psychology, 2*(3), pp. 207-249.

Web, B., Geddes, N., and Neste, L., (1989). Information management with a hierarchical display generator. In *Proceedings of the '89 NCGA Conference, 1*, 52-62.