Cluster-based Representation of Hydraulic Systems

Arthur M. Farley

CIS-TR-87-11
September 28, 1987

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

Cluster-based Representation of Hydraulic Systems

by

Arthur M. Farley *
Computer and Information Science
University of Oregon
Eugene, OR 97403

ABSTRACT

Traditional qualitative modeling techniques have focused on means for abstracting the value spaces of variables that are used to represent system states and for simplifying the constraints that hold among those variable values. In this paper we present a technique for structural abstraction applicable to the domain of pressurized hydraulic systems. Valves, when closed, functionally isolate clusters of components. A cluster can only be in one of two qualitative states -- static, where pressures are equal throughout and no flow occurs, or dynamic, where flow from a high pressure source to a low pressure sink occurs. Reasoning in terms of clusters is shown to facilitate planning for and explaining about the operation and repair of hydraulic systems.

Subjects: Knowledge representation, qualitative reasoning

* Research supported in part by grant # NAG-323 from the National Aeronautics and Space Administration (NASA), Ames Research Center, Moffett Field, CA 94035.

Introduction

To conquer new frontiers, such as manned space flight, we continue to develop new tools of high complexity based on technologies of ever increasing sophistication. As a consequence, it can often be the case that operators using these tools and troubleshooters repairing them do not understand how the technology functions. They can not correctly rationalize steps of standard operation or repair procedures. They can not tell why manipulating some component of a device is part of a standard operating procedure, why observing some aspect of a device's behavior may be relevant to a current diagnosis, or which device failures could possibly give rise to an observed functional abnormality. Limited understanding results in limited performance. Operators are reduced to following procedures by rote from standard checklists and, as a result, make obvious errors; they are not able to demonstrate flexibility in unexpected circumstances and can not successfully troubleshoot a device when a component fails.

Qualitative mechanism modeling has been proposed as basis for the development of useful methods for reasoning about complex devices and for rationalizing steps in operational and troubleshooting procedures. A qualitative mechanism model is an abstract, symbolic model of a device's structure and component behaviors [Bobrow, 1985]. A complex device is modeled as a structure of interconnected components. Value domains of the descriptive variables that represent relevant aspects of each component's behavior are restricted to symbolic value sets of small size. Constraints among variable values represent the behaviors of the various components. An allowable set of descriptive variable values for a component corresponds to a possible qualitative state of the component.

Changes in the qualitative state of a component may arise from the normal evolution of an ongoing process, from operator manipulation of device controls, or from failure of particular aspects of the component. Qualitative mechanism models can be used to explain why a given operational sequence will produce a desired outcome by describing the propagation of a component's state change through other connected components. Similarly, qualitative models of complex systems can serve as bases for predicting what abnormalities will be observed when a given component failure occurs or, reasoning in the other direction, for suggesting possible failures that could account for observed abnormalities.

The primary goal of our research has been to determine a means for simplifying the generation and explanation of plans for the operation and diagnosis of complex hydraulic systems. While traditional techniques of qualitative modeling have focussed on means for simplifying the values and constraints involved in reasoning about physical systems, we have concentrated on finding a method that would reduce the conceptual complexity of the structure of a hydraulic system by aggregating components into isolated subsystems where possible. The result has been the cluster-based approach to the modeling of hydraulic system structure which we report in this paper.

We consider the Orbital Refueling System (ORS), designed for use by space shuttle crews in the refueling of communication and other earth satellites, as a typical example of a hydraulic system that would be useful for space application. We begin our presentation with a description of the Orbital Refueling System, which provides examples for later discussion. We then introduce our notion of cluster-based modeling. Finally, we discuss the application of cluster-based modeling to the generation and explanation of standard operating and troubleshooting procedures.

The Orbital Refueling System

The Orbital Refueling System (ORS) is an experimental system designed for fuel management and transfer to accomplish the replenishment of propellant and other liquid consumables on earth-orbiting satellites. The notion is that a space shuttle (or other space tanker) could rendezvous with an orbiting satellite; an astronaut could go out, connect the ORS to the satellite, and transfer liquid fuel to it, allowing the satellite to remain in orbit and to function much longer than at present.

The architectural design of the hydraulic subsystem of the ORS, presented in Figure 1, itself consists of two primary subsystems: pressurization and fuel transfer. Nitrogen gas, managed under high pressure within the pressurization subsystem, provides the propellant force for the fuel transfer subsystem. The two subsystems interact at two fuel tanks, each containing an internal bladder whereby pressure from the nitrogen gas is transmitted to the hydrazine fuel found in the fuel transfer subsystem. The pressure provides the potential force to move the fuel between the tanks or to external devices, e.g., satellites.

The pressurization subsystem consists of a tank GTK holding nitrogen gas under high pressure sensed by guage @P6, connected through a pipe and T-junction to two valves in parallel, V1 and V2 (where V2 is preceded by an orifice O1). These valves are then connected through parallel pipes and another T-junction to an orifice O5 and pressure regulator REG. Following REG is another T-junction, one branch leading to a relief valve RV and the other branch to a check valve CV. The check valve branch then has a pressure gauge @P5 on the pipe that leads to another T-junction which results in two paths, each eventually leading to a fuel tank. One path has two valves in succession, V3 and V7, prior to fuel tank TK1, while the other has valves V11 and V10 prior to tank TK2. Prior to the branches leading to the fuel tanks is another junction where a side path leads, through two valves V13 and V17, to sump0. Conceptually, sump0 is infinite size tank having pressure lower than any component of the pressurization system (i.e., atmospheric pressure).

The order in which components were given above reflects the order that components would be encountered by a gaseous nitrogen flow from GTK during pressurization of the fuel tanks. Flow is always from a high pressure source, in this case GTK, to a lower pressure sink, here the fuel tanks (TK1 or TK2). Each fuel tank is split by an internal bladder which transmits the pressure of the gaseous nitrogen to the liquid hydrazine fuel, allowing the fuel to be transferred between fuel tanks or to external satellites. We will focus on the pressurization subsystem of the ORS in our discussions that follow. The approach we consider for the representation of the pressurization subsystem can be applied to the fuel transfer subsystem as well.

Hydraulic System Representation

In our description of the pressurization subsystem above, we note that a hydraulic system can be characterized as a set of interconnected components. Each component is represented by a set of descriptive variables, that capture aspects of the component's internal state, a set of ports, each represented by a set of variables that capture tha component's interface with other components, and a set of constraints among these variables that capture the behavior of the component. A set of constraints that identify port variables from various components represent their interconnections.

A hydraulic system model can be described by a **component graph**, with vertices representing components (including pipes) and edges representing their port-to-port interconnections. The graph representing the design of the ORS which was presented in Figure 1 differs from a component graph in that pipes and junctions are drawn as edges rather than vertices. System behavior can be generated from models consisting solely of representations of components and their port-to-port interconnections. STEAMER took a quantiative, interactive modeling approach to the representation of hydraulic systems based solely on component-level models [Hollan et al, 1984]. However, this modeling approach may not necessarily capture the most appropriate level of abstraction for describing system behaviors of interest. Our concerns are with the clarification and explanation of system control and troubleshooting procedures. Since hydraulic systems can be architecturally complex, we wish to find a level of structural abstraction above the component level, if possible.

The key element in our approach to the higher-level representation of hydraulic systems will be the notion of a cluster. A **cluster** consists of a maximal connected subset of system components that does not contain a closed valve. Thus, clusters are separated by closed valves. A closed valve effectively isolates the behaviors of its neighbors, partitioning a hydraulic system into functionally independent subsystems. We define a **cluster graph**, which can be thought of as an overlay to the component graph, where vertices represent clusters and edges represent boundary elements (i.e., closed valves). Two clusters are adjacent in the cluster graph if they have a common boundary element; there may be more than one boundary element between a pair of clusters. The cluster graph for the pressurization subsystem of the ORS representing its proposed launch configuration is shown in Figure 2.

From a behavioral perspective, a cluster can be in only one of two qualitative states: **stable**, where flow equals 0 and pressures are equal and constant throughout the cluster; and **unstable**, where flow is greater than 0, directed along flowpath(s) from a source of higher pressure to a sink of lower pressure within the cluster. In each unstable cluster there exists one or more flowpaths. A **flowpath** is a sequence of components and component ports, beginning with a tank that is a source of high pressure and ending with a tank (or sump0) that is a sink of low pressure. In an unstable cluster, pressure values decrease from a source to a sink as resistances are encountered along the flowpath; over time, pressure decreases at the source and increases at the sink, with flow thus decreasing (assuming non-infinite sources and sinks). Quantitative values for pressures and flow rates depend upon the pressures and resistances associated with components along the complete flowpath; for example, flow rate depends upon the pressure difference between source and sink and the sum of resistances along the flow path.

An unstable qualitative state tends toward stability (in the absence of external addition of material or of thermal energy that would raise pressures within the system); pressures move toward equality and flows decrease. Components such as pressure regulators, check valves and relief valves will automatically change the clustering when pressures reach critical thresholds during the evolution of an unstable qualitative state. For example, a pressure regulator will not allow the pressure (relative to sump0) at its output side to rise above a specified pressure rating. As the output pressure reaches this limit, the regulator valve will close, resulting in two clusters that are stable at different pressures. If the cluster at the output side has its pressure subsequently lowered, the pressure regulator will open automatically, creating flow into the output side of the reunited cluster (assuming the pressure on the input side has remained higher than the output pressure rating of the pressure regulator).

By opening and closing valves, one can change the cluster configuration of a system. By opening a valve, one will most often create an unstable cluster, resulting in the movement of gas or fluid within the system. By closing a valve, one can eliminate an unstable cluster, cutting a flowpath. It is the creation and elimination of unstable clusters that is the primary goal of many standard operating procedures for such systems. Through instability, work is accomplished (i.e., material is moved within the system); through stability, potential for work is maintained. In our research, we will investigate the use of cluster-based modeling of hydraulic systems as a basis for planning and explaining actions taken during standard operating and diagnostic procedures.

Cluster-based Reasoning About Hydraulic Systems

We begin with some useful definitions. A hydraulic system is always in some **current configuration**, being the set of clusters and boundary connections, as determined by the valves that are currently closed. Each cluster is in a **current cluster state**, being one of the two qualitative states described above. Detailed specifications for each of the qualitative states are assumed to be included, such as some, possibly qualitative, measure of the pressure relative to sump0 for a stable cluster and the flowpaths that exist for an unstable cluster. The **current** system state is simply the union of current cluster states.

Given a current system state, we can reason about the future states of the system by a form of qualitative simulation according to the following two rules:

Rule 1: If all clusters are stable, then the current system state will continue indefinitely, until an external change in the current configuration is made.

Rule 2: If a cluster is unstable, it will become stable, either through culmination of change in the qualitative state or through a commanded or automatic change made to the current configuration.

When all clusters are stable, no change occurs without external influence. All pressures are equal throughout each cluster so no flow can occur. Operator-commanded changes to valve settings or faults occurring in system components may change the current configuration and disrupt the overall stability. During instability, the directions of change for pressure and flow at various locations within a cluster tend toward reestablishment of stability, as discussed in the qualitative state descriptions given above. The time it takes to reach stability and the quantitative values of pressures relative to sump0 in the resultant stable state depend upon properties of components in the cluster and upon the initial pressures and capacities involved. If more than one cluster is unstable, it is generally unclear, when reasoning at a qualitative level, which cluster will reach stability first.

Cluster-based Planning for Standard Operations

For standard operating procedures, the usual goals are to raise or lower the pressure (ΔP^+ or ΔP^-) at some location or to create a flow (ΔF^+ or ΔF^-) past a component along a certain flowpath in a certain direction or to stop a flow (ΔF^0) past a certain location. How can our cluster-based approach to hydraulic system representation assist us in generating standard operating procedures that satisfy these types of goals?

To raise (lower) the pressure at some location L, one must merge the cluster containing location L with another cluster having higher (lower) pressure; this results in a new cluster with pressure approaching or stabilizing at some value between the two initial values. By *merging clusters*, we mean opening a valve on the boundary between two clusters that are adjacent, thereby forming a composite, or merged, cluster. If no adjacent cluster has higher (lower) pressure, a subgoal of increasing (lowering) the pressure of an adjacent cluster is generated and problem solving continues in a recursive fashion. To achieve more precise, quantitative pressure changes will require reference to initial pressures, capacities and component specifications (such as ratings for orifices or pressure regulators) that lie on the connecting flow paths or active monitoring of the pressure changes in real time during the unstable state.

To create a flow (ΔF^+ or ΔF^-) past a certain location in a certain direction (+ or -) requires that one put that location in an unstable cluster along a flowpath with a pressure source and a pressure sink on appropriate sides of the location in question. This requires reasoning at both the level of clusters and at the level of component topology. The plan will again involve opening valves on the boundaries of appropriate clusters, thereby merging them into a single unstable cluster containing a flowpath of the desired form. To stop a flow (ΔF^0) past some location, one can cut off the source of higher pressure by closing a valve on the flowpath between the location and the source, thereby containing the high pressure source in a neighboring cluster, or one can disconnect the sink from the location, containing the higher pressure in a cluster that includes the location of interest.

We hypothesize that with an understanding of the two basic qualitative states of clusters and the usual goals for standard operations as discussed above, we will be able to explain most actions performed during standard operation. To demonstrate our approach, consider the operating goal of raising the pressure in fuel tank TK1 of the ORS, given the launch configuration, as shown in cluster-based form by Figure 3. We assume that the pressure in the pressurization tank GTK is greater than that in all other clusters (which are assumed to have approximately equal pressures). Our cluster-based reasoning, given in a means-ends manner, would be as follows:

1. In order to raise the pressure at TK1, we must merge its cluster (C2) with a neighboring cluster having higher pressure.

2. Cluster C2 has three neighboring clusters (C1, C3, and C5), only one (C1) of which has pressure higher than C2.

3. Thus, we have the goal of merging C1 and C2, by opening an valve on its boundary.

4. In this case, there are two valves (V1 and V2) on the boundary between C1 and C2.

5. We open V1 to create an unstable cluster, raising the pressure at TK1.

It is important to note that our search for an operating procedure proceeds through the cluster graph representation of the system. In this case, we did not consider the other components in cluster C2 during our search. By design, the cluster graph is typically much simpler than the component graph of a hydraulic system at any time. Any specification of the final quantitative pressure desired in TK1 can now be addressed by further refinement of the qualitatively determined solution. This could involve determining the length of time to leave the valve open by consideration of parameters characterizing system components or simply monitoring the actual pressure increase in TK1 until the goal is reached. When the specified pressure is reached, the goal of halting the flow into TK1 can be satisfied by closing one of the open valves (V1, V3, or V7) on the flowpath within the unstable cluster.

Cluster-based Reasoning in Troubleshooting

Troubleshooting is the problem solving process whereby aspects of a system's behavior that have become inconsistent with system specifications are again made to satisfy those specifications. We have previously discussed troubleshooting as a problem solving process that occurs across three interrelated problem spaces: Observation Space, Diagnosis Space, and Repair Space [Farley, 1985]. Problem solving in Observation Space is concerned with the design of test procedures and the interpretation of their results in the form of fault-related symptoms. Problem solving in Diagnosis Space is concerned with reasoning about the relationships between observed symptoms and possible faults, managing the current set of possible faults, and proposing symptoms of interest for further observation. Problem solving in Repair Space is concerned with generating plans for the replacement or adjustment of likely faulted components or with the changing of standard operating procedures to avoid ill effects of likely faults (i.e., generating work-arounds).

Faults refer to incorrect operating states of system components. An incorrect operating state results in component behaviors that differ from those specified in a component's definitional description. The outputs of these incorrect component behaviors are then propagated through other components of the system, eventually resulting in changes to observable aspects of system behavior. These noticeable changes in system behavior are *symptoms* of the component fault. Diagnostic reasoning focuses on the problem of associating observed symptoms with possible incorrect operating states of system components or, in other words, determining those faulted components the results of whose altered behaviors could have propagated through the system to produce the observed symptoms.

Troubleshooting must deal with component-level models and these component-level models must include fault models, i.e., descriptions of how a component's behavior is altered when the component is faulted in various possible ways. Below we first describe component models for three basic hydraulic system components. We then discuss how the cluster-based approach to hydraulic system modeling will allow us to generate fault hypotheses in Diagnosis Space that account for observed symptoms. Finally, we note how a cluster-based approach to hydraulic system modeling can assist us in generating plans in Observation Space to determine the presence of relevant symptoms. Symptoms themselves will be defined relative to the cluster in which they appear, the *affected cluster*.

The two simplest components in hydraulic systems are the *pipe* and the *orifice*. A pipe offers no resistance to material flow, and thus transmits flow and pressure from its input port to its output port without change. An orifice does offer resistance to flow, and thus reduces overall cluster flow and produces a drop in pressure P_{drop} between its input and output ports according to the constraint $P_{drop} = \text{Flow} * \text{Resistance}$. The *valve* is an important component type in a hydraulic system. It is a commandable component and thus is a primary focus of attention during the creation and execution of operational procedures. When open, a valve acts like a pipe, transmitting flow and pressure without resistance. When a valve is closed, it does not transmit pressure or flow; its two ports act as if they were part of two separate worlds (note the ports may be connected through other paths, but the closed valve does not contribute to their interaction). When closed, a valve can be part of the boundary between neighboring clusters.

To complete a model sufficient for diagnostic reasoning, we must represent the possible faults for each component type. The qualitative values of interest at any time for a given location within a hydraulic system are flow and pressure and their derivatives with respect to time (i.e., whether they are increasing, decreasing or steady). When reasoning about a system for diagnostic purposes, we are concerned with how observed values differ from expected, or normal, levels [Farley, 1987]. Thus, we will model component faults in terms of how they affect pressure and flow values with respect to normal levels during cluster instability and as well as how they can alter cluster topology.

A pipe may be faulted in one of several ways: leaking, clogged, or blocked. If a pipe is clogged, it acts as like an orifice, introducing resistance and lowering expected flow and pressures along the flowpath(s) of which it is part within an unstable cluster. If it is blocked, it acts like a closed valve permitting no flow or pressure transmission, possibly creating an unexpected cluster boundary within the system. A leaking pipe also changes the architecture of the system, creating an orifice connection to sump0. An orifice may fail in one of two ways: blocked or open. When an orifice is blocked, it acts as a closed valve. If an orifice has failed open, it acts as a pipe, transmitting flow and pressure without resistance and, therefore, at higher than expected levels. A valve can fail in several ways: failed-open, failed-closed, clogged, or leaking. Valve faults are dependent upon the commanded state of the valve. If a valve is open, it can fail as failed-closed or clogged. In the first case, it acts like a closed valve, changing the commanded configuration; while in the second, it acts like an orifice, reducing expected flows. If a valve is closed, it can fail as failed-open or leaking, acting as a pipe or an orifice, respectively; in both cases, the fault alters the commanded configuration, most likely producing flows where none were expected.

How can these component models, together with a cluster-based model of a system, assist us in hypothesizing faulted components from observed symptoms? In a stable state, there should be no flow and static, equal pressures are expected throughout the cluster. Symptoms that could arise in this context are a pressure above or below the expected value or a pressure that is changing, thereby indicating flow when none is expected. In all of these cases, leaking or failed open faults for valves on cluster boundaries are possible faults; the commanded configuration differs from the current system configuration in these cases. If pressure is low and decreasing, the suspected valves would be on boundaries between the affected cluster and neighboring clusters of lower pressure. A leak in a pipe within the cluster, merging the cluster with sump0, is also a possibility. If, on the other hand, the pressure is abnormally high and/or increasing, then valves on boundaries with neighboring clusters of higher pressure are suspect. Externally generated heat, adding energy and thus increasing internal pressure throughout the system, may be considered as well (as is system cooling a possibility with low pressure symptoms). The cluster-based representation focusses our attention on valves at boundaries satisfying certain constraints as being the most likely faulted components.

If the symptom arises while the affected cluster is in the unstable qualitative state, components on the flowpath(s) are candidates for being faulted. If flow or pressure is lower than expected at some location, consider either clogged valves, orifices or pipes along the flowpath in the unstable cluster. If pressures are too high, consider orifices (or pressure regulators) that have failed open (or are biased high). Again the cluster-based representation focusses our attention; here on particular components along a flowpath within the unstable cluster. The cluster-based representation also serves as a basis for explaining why the components are suspected as being faulted. The faulted components are playing important roles in the system's operation, either forming part of a cluster's boundary or lying on the path of an intended flow within the system.

The cluster-based representation can also assist us in generating plans in Observation Space to evaluate the presence of symptoms that would distinguish between possible faults. For performing system monitoring and diagnosis, we assume that the hydraulic system is instrumented with sensors at particular points of interest that allow us to measure flow and pressure values. As an example of problem detection, an initial symptom of higher than expected pressure could be reported by a sensor. If a pressure sensor in a stable cluster is reading high, we can locate another pressure sensor in the same cluster (or open a valve to place another pressure sensor in the same cluster) and check if it reports the same high reading, verifying that the initial sensor is reading correctly. If we suspect a certain valve is leaking and there is an open valve between the sensor reporting the unexpected, changing pressure and the possibly faulted valve, we can close that other valve and check to see whether the pressure stabilizes. If it does, then the suspected valve is the faulted component; otherwise, it is not. These plans in Observation Space clearly depend upon a cluster-based representation of the system and upon fault models of the various components for their generation and rationalization.

Conclusion

In this paper, we have described a cluster-based approach to the representation of hydraulic systems and demonstrated its potential usefulness in generating and explaining procedures. We are currently continuing to formalize our ideas and implement them as part of an interactive, computer-based system for designing, displaying, and reasoning about hydraulic systems. Our interactive system has an interface consisting of three windows: a design window, a cluster window, and a plan window. A typical screen is shown in Figure 3. The design window allows a user to construct component-level models of hydraulic systems. The design window is menu-based and mouse-driven; a user selects a component type from a menu and places the instance on the window, interconnecting various elements by pipe segments as desired. In addition, valves can be open or closed by selection with the mouse in the design window. The plan window allows a user to present standard operational goals to the system and receive in return operating plans that are explained according to a cluster-based representation of the system.

The cluster window displays the current configuration for the system as a graph if interconnected cluster nodes. The cluster graph bears some resemblence to the component graph as a cluster's node is positioned at the center of mass of the cluster's components as they are displayed in the design window. Edges of the cluster graph are labelled by the valves occurring on the boundaries. The cluster graph display is automatically updated as valves are opened or closed. A cluster node can be selected with the mouse and its components are highlighted in the design window.

We believe the concurrent display of the component and cluster graphs of a hydraulic system during system control by human operators is potentially a very important application of our ideas. The simplified view of the system presented by the cluster graph can assist the operator in generating actions, as previously discussed, as well as in understanding the effects that actions have on system configurations. Mistakes in the interpretation of action effects are often primary factors in the occurrence of system failures involving human error. Many of these errors probably can be traced to misconceptions as to the current cluster configuration of a system. For instance, at Three Mile Island, cooling pumps were turned on but, due to closed valves, the pumps were not part of the cluster involving reactor cooling pipes; thus, the expected flow and cooling were not realized [Rubenstein, 1979]. A cluster-based display could have made this error apparent to the operators.

References

Bobrow, D. (ed.) *Qualitative Reasoning About Physical Systems*, MIT Press: Cambridge, Ma., 1985.

Farley, A.M. "A general model of troubleshooting and its application to computer support", in *Proceeedings of the Fifth Conference on Expert Systems and their Application*, Avignon, Fr., p489-503 , May, 1985.

Farley, A.M. "Diagnostic mechanism modeling", Technical Report # CS-TR-87-08, Computer and Information Science Department, University of Oregon, submitted for publication.

Hollan, J., Hutchins, E.L. and Weitzman, L. "STEAMER: An interactive inspectable simulation-based training system", AI Magazine, p15-27, Summer, 1984.

Rubenstein, E. "The accident that shouldn't have happened", *IEEE Spectrum*, 16 (11), p33-42, November, 1979.

ORS PRESSURIZATION SYSTEM

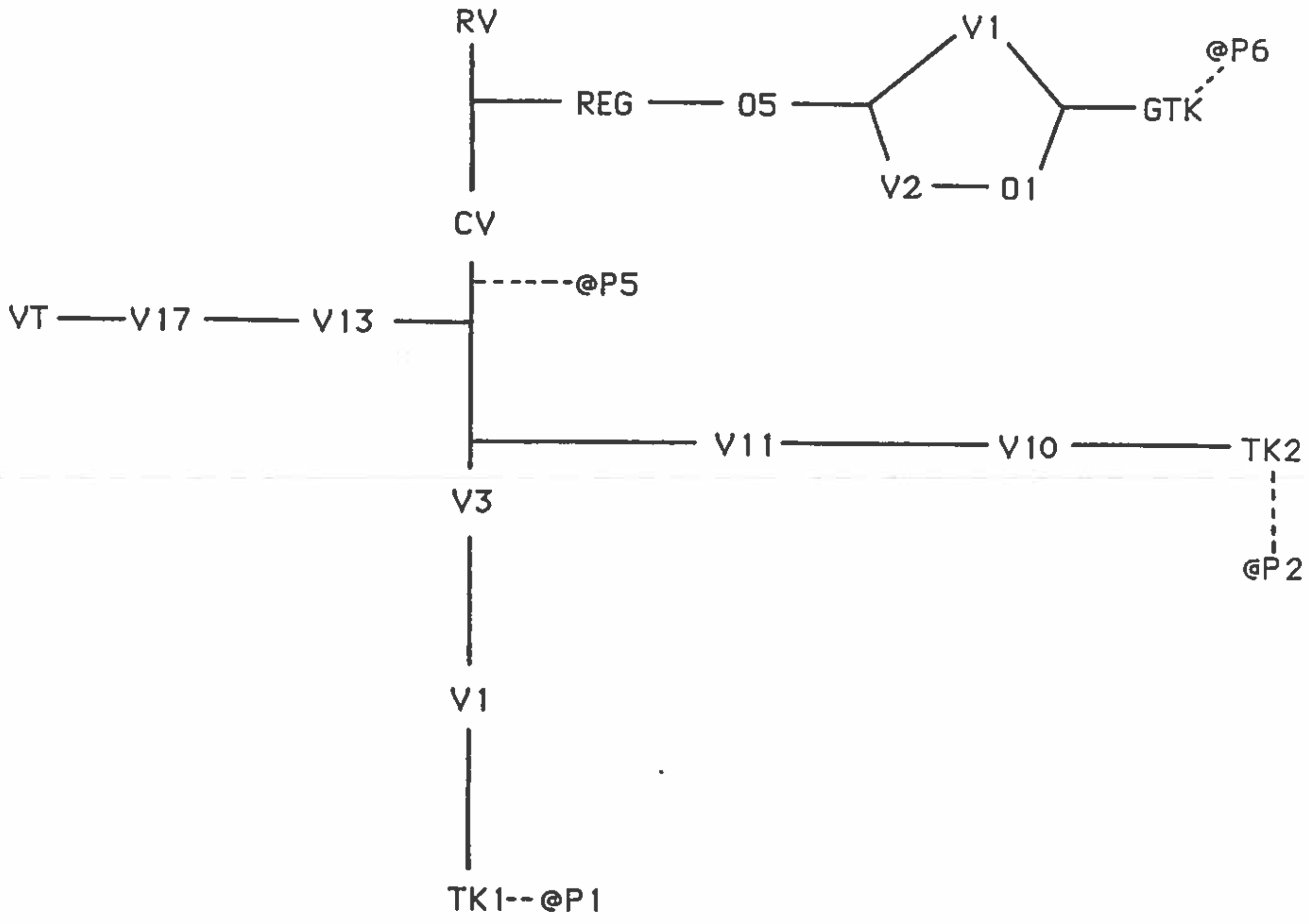


Figure 1. Component Representation

LAUNCH CONFIGURATION

C1: GTK, 01, @P6

C2: 05, REG, CV, @P5
V3, V7, TK1, @P1

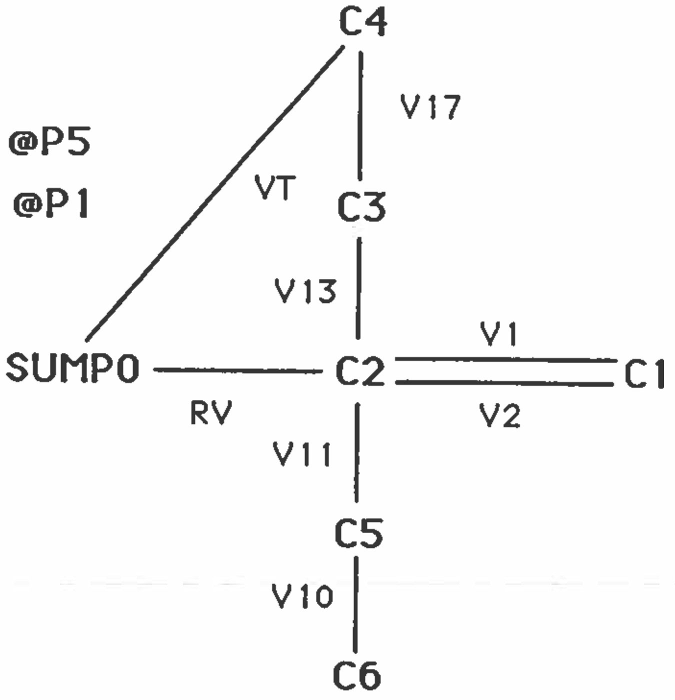
C3:

C4:

C5:

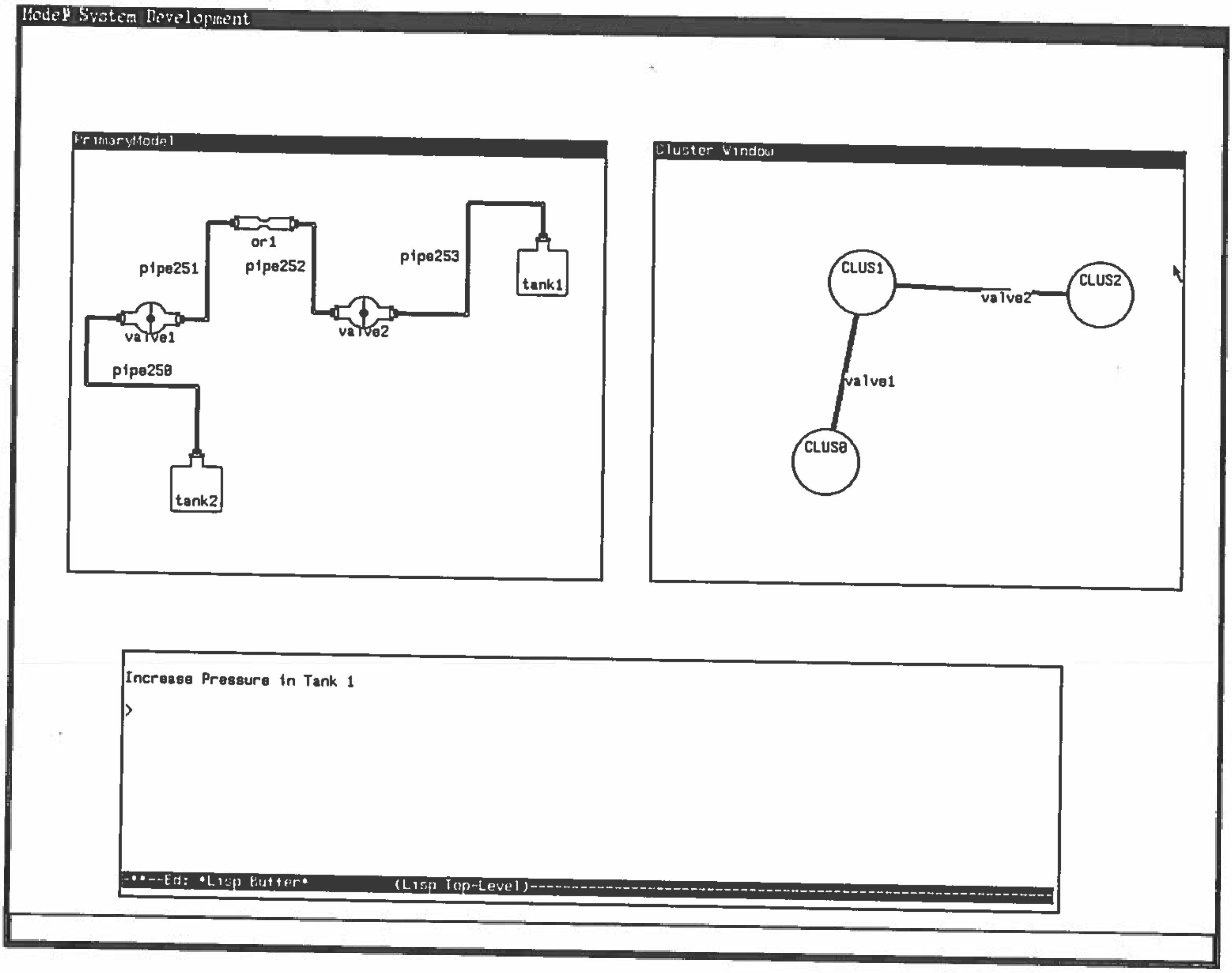
C6: TK2 , @P2

SUMP0:



Pipes and Junctions not included in cluster component listings.

Figure 2. Cluster Configuration



```
Increase Pressure in Tank 1
>
-**--Ed: *Lisp Buffer*          (Lisp Top-Level)------------------------------------------------
```

Figure 3. Simple Screen from Interactive System