A Theory of Mapping from Structure to Function Applied to Engineering Thermodynamics

John O. Everett

Qualitative Reasoning Group at The Institute for the Learning Sciences
Northwestern University 1890 Maple Avenue Evanston, IL 60201 USA
phone 708-491-4790 fax 708-491-5258
email: everett@ils.nwu.edu

Abstract: This paper presents a theoretical framework for mapping from structure to function in engineering domains. We argue that a generative approach grounded in Qualitative Process Theory produces useful functional explanations. Useful explanations are articulate, in that they enable the user to explore their theoretical justifications and they support counterfactual reasoning. These explanations stem from a teleological representation based on goals, plans, roles, and views. We also show that an ontology based on aggregated processes provides a powerful means for recognizing recurring thermodynamic structures. We describe an implementation of this theory, a system called CARNOT, that currently explains steady-flow thermodynamic cycles ranging in complexity from four to twenty-four components.

1. Introduction

Thermodynamic cycles (e.g., power plants, refrigerators) form an important class of artifacts. Devices based on them are typically complex and often costly to operate, which provides several motivations for reasoning about them. Engineers and students need to verify that their designs will behave as desired, and plant operators need to generate and test hypotheses concerning system functions from schematics.

Each of these cases calls for reasoning about function given a structural description. This paper describes a theory of structure-to-function mapping that supports these tasks in the domain of thermodynamic cycles. We have implemented this theory in a system called CARNOT that takes as input a schematic depicting the structural configuration of a system such as a refrigerator and produces a description of the system's function.

de Kleer (1984) was the first to investigate the mapping from structure to function. He proposed, for the domain of electronic circuits, a methodology using qualitative physics to map from structure (what the artifact is) to behavior (what the artifact does) and a separate, teleological reasoning process to map from behavior to function (what the artifact is for). Thus the behaviors of a working turbine include expansion of the working fluid, cooling of the fluid and creation of shaft work. Its function, however, may be either to produce work or to cool the working fluid, and depends on the context in which it is embedded.

We observe that the causality of electronic circuits is particularly ill-defined, especially when compared with other engineering domains, such as thermodynamics. For example, thermodynamic cycles have a readily-established direction of flow, and their constituent components typically have few or no degrees of freedom in their potential behaviors. We initially believed that a simpler approach than de Kleer's would suffice for mapping from structure to function in domains other than electronics.

This turns out not to be the case; we encountered significant ambiguities in mapping from the structure of thermodynamic cycles to their function. This paper describes how our theory resolves these ambiguities to produce a single functional description of a schematic. Section 2 presents an overview of the domain, Section 3 discusses our theory, Section 4 describes our representations, Section 5 outlines the algorithm, and Section 6 presents in detail one example and summarizes some of the more interesting results from other cycles CARNOT currently solves. We conclude with a discussion of related and future work.

2. Domain Overview

Artifacts incorporating thermodynamic cycles are pervasive. Virtually all electrical power generated today relies on a thermodynamic cycle in which massive boilers generate steam to turn turbines that drive generators. Refrigerators rely on the same cycle, albeit running in reverse and supplied with a different working fluid that enables operation at safer pressures. Automobile and jet engines operate in a so-called "open" cycle that takes in air from, and expels exhaust gases to the environment. Industry relies on thermodynamic cycles for power, for liquefying gases (e.g., natural gas), and for process steam.

A Simple Heat Engine The defining characteristic of a thermodynamic cycle is that it operates between two reservoirs of different temperatures, typically by passing a working fluid through a system of pipes and components. Figure 1 depicts a simple heat engine.

Simple Vapor-Cycle Heat Engine

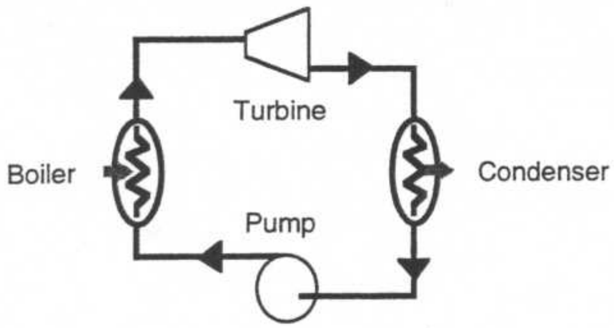

Figure 1

This basic cycle (with modifications to increase efficiency) is commonly used to generate electricity. Heat energy from combustion or nuclear reaction converts the working fluid into vapor in the boiler. This fluid leaves the boiler at high pressure and expands in the turbine, converting its pressure into velocity. The kinetic energy of the steam striking the turbine blades forces the turbine shaft to rotate, producing work. The condenser returns the working fluid to its original state by ejecting heat to the environment. The pump ensures that the boiler gets a steady supply of working fluid at a pressure high enough to maintain the system's direction of flow.

The constituent devices of this and other thermodynamic systems are complex artifacts designed to accomplish specific functions, which led us to believe that teleological recognition in this domain would be less ambiguous than de Kleer found the electronics domain. However there are significant ambiguities in mapping from behavior to function in this domain. For example, a turbine may function as either a work-producer or a cooler, and in cryogenic cycles the latter is the desired function. Reaching human-like conclusions with little information despite such ambiguity has been the primary motivation for the development of our theory.

3. Teleological Theory

The goal of this research is to automate the process of making functional inferences from structural information. To gauge our success, we need criteria for what constitutes a good functional explanation. However, a system that produces "good" explanations will be useful only insofar as it scales up to solve real-world problems in a reasonable amount of time.

We define a good functional explanation to be one that:

- Provides a single, internally-consistent explanation for the system
- Takes into account all available information
- Relates each device to at least one design goal
- Provides an indication of the certainty of each inference
- Enables counterfactual reasoning about the function of the system
- Provides a chain of inference that grounds in a qualitative theory of behavior

The value of functional explanations lies not in the explanations per se but in the inferences they sanction. A simple pattern-matching approach could produce canned descriptions in great detail. However, such explanations would be unresponsive to user queries. A student might not understand a particular statement, and should therefore be able to backtrack through that statement's inferential chain all the way to domain theory givens. An engineer might want to know how a design modification would affect the system's functioning.

A generative approach based on qualitative physics is necessary to address these issues. To achieve generativity, we minimize the size of knowledge fragments and rely on inference to assemble a set of fragments into an explanation. This minimizes the redundant encoding of information (which would occur in a template-matching system) and enables the explanation of novel cycles. Modularizing the representation also facilitates the task of maintaining a knowledge-base large enough to support a practical teleological reasoner. Our representation is consistent with Qualitative Process Theory (Forbus, 1984), and therefore produces explanations grounded in qualitative behavioral inferences. However, it also enables the use of quantitative information for making more precise functional inferences.

Finally, to prevent explosive inferencing, CARNOT adopts what de Kleer (1984) calls the *teleological perspective*, in that we assume by default that each device contributes to the function of the system. This enables us to avoid extensive qualitative and/or quantitative simulation because we assume that components operate within their normal parametric ranges. Because this *rational-designer premise* is assumed explicitly, retracting it enables us to consider alternative situations, such as temperatures exceeding component tolerances.

4. Knowledge Representations

Perusal of thermodynamic texts and reference materials reveals no universal standards for schematics, although informal conventions do exist (e.g., turbines are generally represented as trapezoids with vertical parallels). We use the schematic representation we designed for CyclePad (Forbus & Whalley, 1994), a system that enables students to design and experiment with thermodynamic cycles.

This representation, reflects the pedagogical considerations underlying CyclePad. To encourage students to consider modeling issues, only basic devices are explicitly represented. For example, there is no jet-ejector (a pump utilizing a high-velocity jet) because a mixer can function in this capacity. Likewise, mixers may also function as open heat-exchangers (heating via mixing) and splitters may function as flash-chambers (devices that cause the working fluid to rapidly evaporate due to a pressure drop).

CARNOT'S functional descriptions are composed of *plans* and *roles*. Plans summarize common structural configurations that have particular functional import. Roles specify which behavior of a particular component is its intended function. Intermediate *view* and *process* constructs enable the instantiation of the proper plans and roles. Views describe possible behaviors of particular devices, while processes ground explanations in a qualitative model of thermodynamics and provide a useful definition of locality. Figure 2 provides an overview of these representations. Arrows indicate constraining relationships. For example, the topology of the schematic determines which processes, roles, and views are instantiated. We describe these constructs in more detail in the balance of this section.

Goals The rational-designer premise allows us to restrict the number of goals we impute to a system. From this point of view, thermodynamic cycles have three possible design goals, (1) achieving a change of state in the environment, (2) doing so with a minimal input of energy, and (3) preserving the integrity of the system. In the case of a heat engine, the first goal is to convert heat energy into shaft-work, whereas for a refrigerator it is to move heat from one location to another. The second goal, of maximizing efficiency, follows from the teleological perspective; each device is assumed to contribute to the function of the whole because the designer faces tight economic constraints. Finally, because configurations that

CARNOT'S Representations

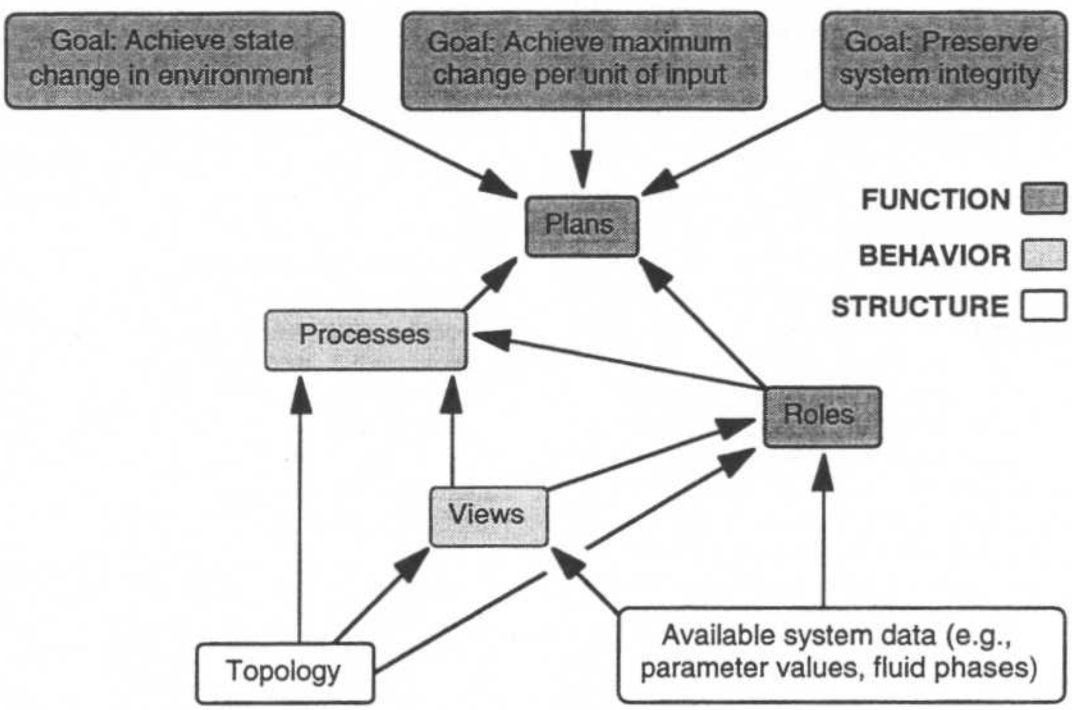

Figure 2

achieve the first two goals also create potentially damaging conditions, some devices may be present solely to prevent the occurrence of such states. For example, most pumps cannot handle fluids composed of a mixture of liquid and gas, so an upstream pump may ensure that preheating of the working fluid does not cause it to flash into vapor.

Views Views are device-specific behavioral descriptions. For example, a pump's views include *default*, *coasting*, *cavitating*, and *losing*. By default, CARNOT considers pumps to compress liquids (compressors compress gasses), so a default pump view sanctions inferences that both input and output stuffs are liquid and that the input stuff pressure is less than that of the output.

Views are biconditionals of the form "if and only if this particular conjunct of facts is found to be true, then this view is active." For example, the conjunct of facts for a liquid-heater view includes (1) flow is positive, (2) phase of stuff in is liquid, (3) phase of stuff out is liquid and (4) temperature of stuff in is less than temperature of stuff out. Views propagate phase information, which can resolve ambiguities and prevent devices known to be behaving abnormally from participating in process and plan inferences.

Roles Roles are the functional counterpart of views. For example, the potential roles of a pump include (1) flow-producer and (2) flash-preventer. The behavior of a default-view pump is to compress liquid; its function is to produce a flow. The difference is a presumption that this flow is essential achieving one or more of the three design goals. A view is insufficient to support this presumption, because it is possible that the actual function is to act as a work-sink rather than to create a flow.

Although roles are device-specific, they generally require consideration of the structural context for a device, and hence more reasoning. Unlike views, roles are not always mutually exclusive; indeed, achieving multiple functions via a single device is often desirable from a design standpoint, for potential cost-savings and/or efficiency improvements. For example, a pump may act as both a flash-preventer and a flow-producer. In the flash-preventer role a pump functions to prevent the fluid downstream from suddenly vaporizing.

Processes Although the fixed flow direction and component-oriented schematics for thermodynamic cycles would appear to make them suited to a device-centered ontology, we have found a process-oriented ontology more useful. Processes are central to thermodynamics; the components of a particular cycle exist solely to create and control them. Moreover, processes often span several devices, which may or may not be immediately adjacent. Reifying such processes provides CARNOT with a powerful definition of locality.

CARNOT distinguishes three types of process: (1) local, (2) boundary, and (3) aggregate. Devices create one or more local processes across their fluid-paths. For example, a pump creates a local fluid-flow process from its inlet to its outlet.

We adopt the thermodynamic convention of establishing *control volumes* around systems and subsystems of interest. Control volumes require an accounting of all mass and energy crossing their boundaries. CARNOT explicitly labels all boundary-crossing processes. For example, the heat-flow to a boiler must cross the system boundary, so all heaters give rise to boundary heat-flow processes.

Aggregate processes provide a flexible means for matching canonical plans to cycles, because they capture critical aspects of a system without being overly sensitive to its particular topology. For example, inserting a pump and a heater immediately after the cooler in Figure 1 would improve the cycle's efficiency yet not alter its function nor its identity as a Rankine cycle. The example presented in Figure 9 is, despite its complexity, still a Rankine cycle.

Plans Certain thermodynamic configurations recur so often that their idealized abstractions have been reified. For example, most electrical power generating systems use the Rankine cycle, around which a working fluid is vaporized and condensed. We refer to such named configurations as plans because they are in effect strategies for the realization of design goals. Figure 3 shows our representation of the Rankine cycle plan. Other common plans include the Carnot cycle, a theo-

Rankine Cycle Plan

- Vaporize working fluid at constant pressure
- Create a constant-entropy resisted expansion to produce shaft work
- Fully condense working fluid at constant pressure
- Pump liquid working-fluid at constant entropy to maintain flow direction

Figure 3

retical ideal, and the Brayton cycle, used for jet engines.

Idealization simplifies analyses by assuming certain state parameters remain constant across the plan's processes. In the ideal, the Rankine cycle is comprised of constant-pressure (isobaric) heating and cooling processes and constant-entropy (isentropic) expansion and compression processes.

CARNOT distinguishes truly ideal from stepwise-ideal processes. The latter occur when the creation of two processes is interleaved. For example, pumps and heaters may be interleaved, obviously preventing the aggregate heating process from occurring at constant pressure. However, if each constituent local process is ideal, then CARNOT labels the aggregate as stepwise-ideal. This distinction enables CARNOT to differentiate practical from ideal cycles. An ideal cycle maximizes efficiency even at the cost of destroying system integrity. Such cycles are useful for pedagogical reasons and as benchmarks for assessing the efficiency of practical cycles.

CARNOT'S plans vary in generality. The most general are the heat-engine and refrigerator plans, which have no ideal-process requirements. The more information CARNOT is given, the more specific the plans instantiated. For possible plans, CARNOT backchains on their antecedents and includes them in the final description with a caveat that the antecedents must be true. In addition to cycle plans, CARNOT recognizes inter-cycle plans, such as CASCADE-CYCLES and USE-WORK-INTERNALLY. In cascaded systems one cycle uses the heat ejected by the other, while in systems that combine heat-engine and refrigerator cycles, the heat-engine's work drives the refrigerator.

CARNOT'S most specific plans arise from particular device roles. For example, a mixer serving as a jet-ejector (a type of pump in which a high-velocity jet entrains the working-fluid being pumped) will cause the instantiation of a plan to preserve system integrity by reducing complexity, because jet-ejectors replace turbine-compressor combinations.

5. CARNOT's Algorithm

CARNOT uses a logic-based truth maintenance system (Forbus & de Kleer, 1993) coupled to a pattern-directed rule engine. CARNOT'S knowledge base is encoded as a set of rules. The underlying TMS caches the resulting chains of inference, enabling CARNOT to perform counterfactual reasoning and to construct causal explanations on demand. Figure 4 shows the steps of the algorithm.

Instantiating Domain Knowledge CARNOT first instantiates a set of device models that describe the structure of the input system and result in the instantiation of views. In some cases there isn't enough initial information for a particular device to have an active view. For these devices CARNOT instantiates the most specific view consistent with the known information.

For example, the default view for a heater makes no commitment about the phase of the stuff at inlet or outlet. However, if CARNOT detects that there are only compressors (which can only compress gasses) present, it will assume a Gas Heater view, which implies that the phase of the stuffs at inlet and outlet is gas.

Identifying Topological Structures CARNOT next parses the cycle topologically into *floops* (short for "fluid loops"). These are directed cycles in which neither arcs nor vertices are duplicated.[1] CARNOT breaks floops immediately upstream of the first compressing device to be found after the last expansion device. This is the most "natural" point at which to break a cycle, because the working fluid is closest to ambient conditions here. For example, the automobile engine, which operates in a so-called "open" cycle, breaks the cycle at this point, because it takes in its working fluid (i.e., air) immediately prior to compressing it, and exhausts it immediately after the power stroke.

Floops do not necessarily correspond to meaningful substructures in the input cycle, but merely represent routes that a piece of working fluid could traverse. Therefore CARNOT next generates a hypothesis concerning the function of each floop. Each floop is potentially a heat-engine, a refrigerator, or a topological artifact. Initially we thought it would be possible to identify floops by the order of device occurrence because the canonical heat-engine has a pump, heater, turbine, and cooler in that order while the canonical refrigerator swaps the heater and cooler, but this turns out not to be the case, because splitters and mixers have several possible functions.

---

[1] To avoid terminological confusion, we reserve the term "cycle" for thermodynamically meaningful closed paths.

Algorithm Step	Example of Result
1. Assert propositions describing system	(device turbine tur-1 s10 s20) (tur-1 HAS-INLET s10)
2. Run rules to instantiate immediate consequences of description	(tur-1 HAS-VIEWS (default eroding stalling))
3. Identify fluid loops in cycle	(FLOOP fl-1 (pmp-1 htr-1 tur-1 clr-1) (s10 s20 s30 s40))
4. Create a consistent view structure for system	(VIEW default HEATER htr-1)
5. Do dependency-directed search for roles	(ROLE mxr-1 JET-EJECTOR)
6. Identify routes in system	(SUBCYCLE subc-1 (pmp-1 htr-1 tur-1 clr-1) (s10 s20 s30 s40))
7. Refine view structure in light of new information	(VIEW EVAPORATING HEATER htr-1)
8. Aggregate local processes	(PROCESS AGGR EXPANSION (tur-1 tur-2 tur-3 tur-4 tur-5))
9. Run rules to identify plans	(PLAN RANKINE-CYCLE subc-1)

Figure 4

A mixer can act as either a simple route-joiner, a heat-exchanger (if its two inputs are of different temperature) or a pump (if its two inputs are of different pressure). A splitter may either act as a route-divider or a flash-chamber, in which the working fluid evaporates, the gas leaving by one exit and the remaining liquid by the other. We first confronted this issue in breaking floops that had no apparent compressor. In such situations, CARNOT considers mixers to be compressors, the only remaining thermodynamic possibility if the floop is in reality a subcycle.

This functional ambiguity means that valid subcycles may lack apparent pumps, expansion devices, heaters or coolers. To identify such floops, CARNOT uses the constraints shown in Figure 5 to conduct a dependency-directed search for a consistent set of views of each floop's devices. These constraints follow from the rational-designer premise; there is no thermodynamically sound reason to heat and immediately cool or compress and immediately expand a working fluid. For this search, CARNOT generates sets of potential roles for each mixer and splitter, ordered such that any solutions that allow the default mixing and splitting roles of those devices will be found first, so that simpler solutions are preferred.

Functional Labeling Constraints

- Processes are considered neighbors if they are consecutive on a particular route or if they are connected by one or more splitting and/or mixing processes.
- Heating and cooling processes cannot be neighbors.
- Expansion and compression processes cannot be neighbors.

Figure 5

On completion of the search, CARNOT generates a refrigerator, heat-engine, or topological-artifact hypothesis based on either the order of devices in the floop or the presence of devices which could accomplish the essential compression, heating, expansion, and cooling processes. Device order, although more persuasive evidence than mere presence, is not a certain indicator of floop type. CARNOT therefore asserts hypothesis statements that contain the inferred floop type, the justification for the inference (ORDERED or ALL-PRESENT) and the set of role assumptions required for that floop type to pertain. CARNOT postpones committing to a hypothesis, however, because that requires non-local reasoning and can be made with greater certainty later in the processing.

Resolving Roles via Qualitative Inference Roles depend on the context in which the device is embedded. For the jet-ejector and open heat-exchanger roles, this context is limited to the state of the mixer's inputs; a temperature difference across the inputs indicates an open heat-exchanger, while a pressure difference implies a jet-ejector. When CARNOT instantiates its knowledge of mixers, it also expresses interest in finding inequalities in either pressure or temperature across the mixer's inputs, in order to avoid aimless transitivity calculations.

Once CARNOT has identified the system's floops, whatever information necessary to find these ine-

Identifying an Open Heat-Exchanger via Inequalities

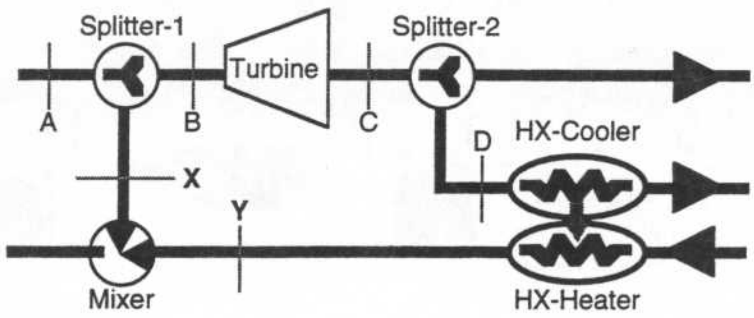

Figure 6

qualities, should it exist at all, will be present in the database. At this point, CARNOT attempts to assert an inequality statement for the identified stuff parameters via transitive reasoning. For example, in the cycle fragment of Figure 6 the mixer can be identified as an open heat-exchanger, given CARNOT'S domain knowledge. The transitivity reasoning proceeds as follows:

1. No temperature drop across a default-view splitter gives T(A) = T(B) = T(X).
2. Temperature drop across a default-view turbine gives T(B) > T(C).
3. No temperature drop across a default-view splitter gives T(C) = T(D).
4. T(D) ≥ T(Y) because, at best, perfect heat transfer in the default-view heat-exchanger would make T(D) = T(Y).
5. By transitivity, T(Y) ≤ T(C) and therefore T(Y) < T(B).
6. Because T(X) = T(B), we can deduce that T(Y) < T(X), thus satisfying the conditions for an open heat-exchanger.

Identifying Subcycles and Paths CARNOT now attempts to re-parse the input system into a set of mutually exclusive and collectively exhaustive subcycles and paths, the latter originating at splitters and terminating at mixers. The only case in which CARNOT relaxes the "no-gaps/no-overlaps" constraint is when a hypothesized heat-engine shares structure with a hypothesized refrigerator, because such shared structure can improve efficiency and reliability.

CARNOT identifies subcycles by applying the following heuristics:

1. Should a floop exactly subsume two or more floops, consider only the subsumed floops.
2. Floops that have no structure in common with other floops are considered subcycles, as are lone heat-engine and refrigerator floops.
3. Of a set of floops sharing structure, choose the putative heat engine with the greatest number of worksources, or choose the putative refrigerator with the greatest number of fluid heaters (i.e., refrigerator coils).

CARNOT now accepts or rejects the type hypothesis for each identified subcycle. If all the views required for the hypothesis to hold are true, then CARNOT simply records the relevant type statement, while if a single view is false, CARNOT asserts the implication that the subcycle is not of purported type. When one or more view statements are unknown, CARNOT assumes in turn that each unknown is true and looks for any resulting contradictions in its knowledge of the system. For example, CARNOT may know that the temperature across a mixer is constant, so assuming that the mixer is an open heat-exchanger would cause a contradiction. Should such a contradiction occur, CARNOT retracts the view and asserts that both the view and the hypothesized type cannot mutually pertain. Otherwise, CARNOT assumes the views are valid and accepts the hypothesized floop type.

Aggregating Processes and Inferring Plans The set of active views determines what processes are considered to be active. For example, a boiling process is only active if its associated heater is viewed as a Boiling-Heater.

Aggregate processes arise from two or more devices operating in conjunction to produce a single effect. CARNOT aggregates local processes according to the set of heuristics shown in Figure 7, which are based on the rational-designer premise; there is no physical law enforcing these constraints, but violating them would serve no thermodynamic purpose, and in fact be at odds with one or more of the three teleological goals CARNOT imputes to an input system. Figure 9 il-

Rules For Composing Aggregate Processes

- Heating processes
  - May have an arbitrary number of intervening pumping, mixing, and splitting processes
  - Last local heating process must be downstream of last local pumping process
  - No intervening cooling, expansion, or throttling processes
- Cooling processes
  - May have an arbitrary number of intervening mixing, splitting, and expansion processes
  - No intervening heating processes
- Compression processes
  - May have an arbitrary number of intervening heating, mixing, and cooling processes
  - First local compression process must be upstream of the first heating process
  - No intervening expansion processes
- Expansion processes
  - May have any number of intervening heating, cooling, splitting, and mixing processes
  - Last local expansion process must be downstream of last heating process
  - No intervening throttling or compression processes

Figure 7

lustrates the aggregation resulting from an application of these rules to a cycle.

The assertion of plans is simply a local propagation based on the current set of active aggregate processes and other information cached in the database. Figure 8 shows the rules for instantiating an ideal Rankine cycle plan.

6. Example

We present here a cycle typical of those CARNOT can explain. The cycle shown in Figure 9 is a practical Rankine cycle for electrical power generation. Steam bleeds from the turbine (represented by the five turbines in series across the top of the schematic) preheat the feedwater to increase the efficiency of the cycle. CARNOT'S explanation of this system is, translated from the predicate calculus:

- The system is a heat engine. Given stepwise isentropic expansion in the turbines and stepwise isobaric heating in the heaters, it is a practical Rankine cycle, because it is a vapor power cycle. It is a vapor power cycle because it condenses its working fluid.
- Turbines 1-5 create the resisted expansion process of the system. Heaters HX-1, MIXER-2, HX-2, HX-3 and BOILER create the heating process. Pumps 1-3 create the compression process. CONDENSER creates the cooling process.
- MIXER-2 is an open heat-exchanger because the fluid from splitter SPL-3 has a higher temperature than the fluid from HX-1. This is done to achieve the design goal of MAINTAIN-SYSTEM-INTEGRITY, because an open heat-exchanger removes harmful impurities from the working fluid.
- Pumps PUMP-2 and PUMP-3 may act to prevent the working fluid from flashing. This would achieve the design goal of MAINTAIN-SYSTEM-INTEGRITY, because flashing would cause downstream pumps to cavitate, cavitation would cause the pump's fluid-flow-rate to decrease, and a decrease in fluid-flow to the boiler would cause the boiler to melt. [This inference is uncertain because it is

Rules for Instantiating Rankine Cycle Plan

```
(RULE ((:true (PLAN HEAT-ENGINE ?name ?cmp ?htg ?exp ?clg) :var ?p1
          :test (every #'liquid-pump? (cadr ?cmp))))
      (let ((pumps (make-pump-exprs ?cmp)))
      (assert! (:implies (:and ?p1 pumps)
                         (PLAN VAPOR-POWER ?name ?cmp ?htg ?exp ?clg))
            :vapor-power-cycle-inference)))

(RULE ((:true (PLAN HEAT-ENGINE ?name ?cmp ?htg ?exp ?clg) :var ?p1)
       (:true (PLAN VAPOR-POWER ?name ?cmp ?htg ?exp ?clg) :var ?p2)
       (:false (INTERLEAVED ?cmp ?htg) :var ?p3)
       (:false (INTERLEAVED ?htg ?exp) :var ?p4)
       (:false (INTERLEAVED ?exp ?clg) :var ?p5))
      (assert! (:implies (:and ?p1 ?p2 (:not ?p3) (:not ?p4) (:not ?p5))
                        (ISOBARIC ?htg) (ISOBARIC ?clg) (ISENTROPIC ?exp))
                        (PLAN RANKINE IDEAL ?name ?cmp ?htg ?exp ?clg))
            :ideal-Rankine-cycle-inference))
```

Figure 8

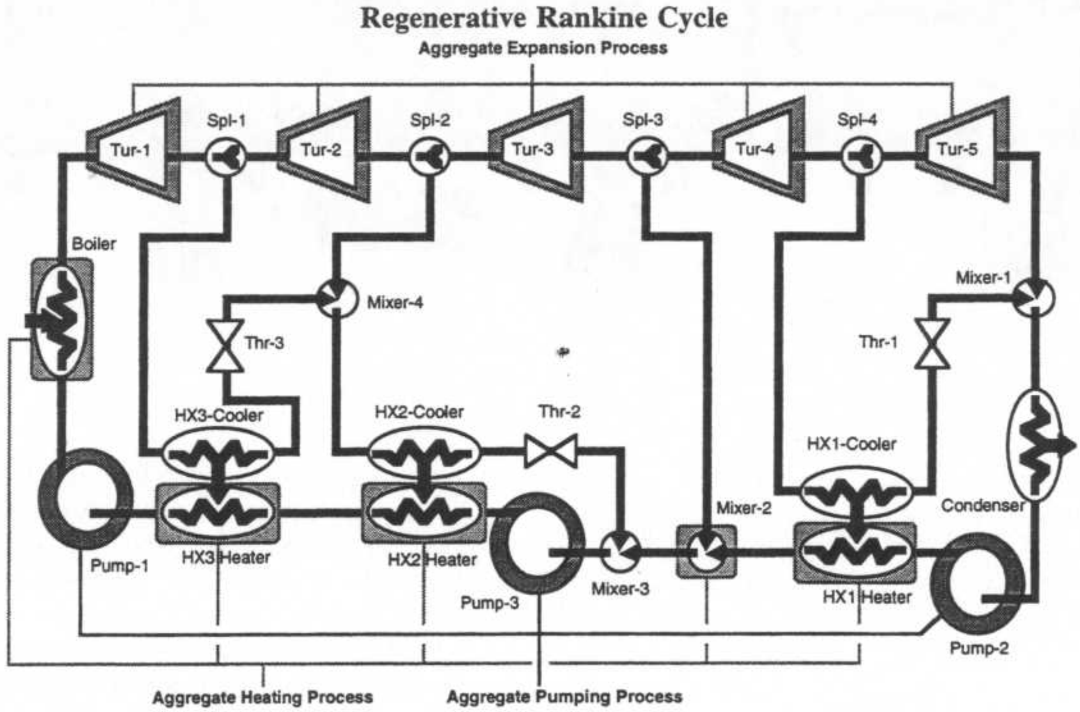

Figure 9

based solely on the cycle's topology; PUMP-2 and PUMP-3 have both heaters and pumps downstream of them, so it is possible that the removal of either pump would enable a downstream heater to cause the fluid to flash into vapor. Given numeric information, CARNOT can determine whether this would actually occur].

- Heaters HX-1, MIXER-2, HX-2 and HX-3 preheat the working fluid. This is done to achieve the design goal of MAXIMIZE-SYSTEM-EFFICIENCY, because a Rankine Cycle's efficiency is directly related to the average temperature of heat addition.

7. Related Work

Chandrasekaran has developed a theory of Functional Reasoning that is consistent with the work presented here. He has proposed that teleological knowledge be encoded in Causal Process Descriptions (CPDs) that are represented as directed graphs whose arcs are causal links (e.g., Chandrasekaran, 1994). CARNOT'S knowledge base is organized along similar lines, although we prefer not to encode the causal links explicitly, and instead allow the inference engine to instantiate them as they become relevant, via the view and role mechanisms.

Vescovi, Iwasaki, Fikes, and Chandrasekaran have proposed a modeling language, CFRL, for integrating qualitative and functional reasoning (Vescovi et al, 1994). CFRL composes a qualitative model from model fragments and then attempts to fit a causal story (encoded in CPDs) to a particular trajectory through the qualitative state space. Because thermodynamic cycle analysis is steady-state, we have been able to avoid the complexities arising from such explicit temporal reasoning.

Franke has proposed a rigorous language for teleological description (TeD) (Franke, 1993) that may in the future provide us with useful formalisms as we extend CARNOT. He approaches the issue of teleology from the designer's point of view, while CARNOT attempts to infer the intentions of the designer after the fact, given only the artifact.

Narayanan, Suwa and Motoda have described a system that predicts the operation of simple mechanical devices from labeled schematic diagrams (Narayanan et al, 1994). Their system produces explanations similar to CARNOT'S, via visual reasoning, whereas CARNOT'S input is a set of propositions describing devices in a particular structural configuration.

8. Discussion

We have described a set of teleological representations consisting of goals, plans, roles, and views that enable the production of functional explanations of complex thermodynamic cycles grounded in a qualitative domain theory. We have also shown that aggregating processes provides a powerful heuristic for recognizing cycles despite structural variations.

We believe the generativity of our approach will enable it to scale up to explain any thermodynamically valid system. CARNOT now explains all eight of the steady-flow cycles contained in an introductory text (Whalley, 1992), and twenty-four of the thirty-two cycles in a more comprehensive text (Van Wylen & Sonntag, 1985).

In the worst case, CARNOT could exhibit exponential behavior due to the dependency-directed search it conducts to resolve device roles. However, the structure of the domain is such that both the number and size of choice sets for this search remain small (typically five choice sets of size two or three each). We anticipate that CARNOT will continue to exhibit the polynomial performance we have seen to date.

CARNOT's current limitations stem from gaps in its knowledge base. For instance, the rules needed to infer the presence of an air-standard refrigeration cycle have yet to be implemented. Our next goal is the explanation of non-steady-flow systems, such as the Otto and Diesel cycles used in automotive engines. We believe that some improvements to the algorithm combined with roughly a one-third increase in CARNOT'S current rulebase (which now contains about 140 rules) will enable the explanation of all thirty-five cycles contained in *Analysis of Engineering Cycles* (Heywood, 1980), considered to be the definitive text on thermodynamic cycles.[2]

As an initial test of its capabilities, we intend to incorporate CARNOT into the coaching module of a thermodynamics tutoring system. We also intend to test the applicability of this theory to other domains, such as hydraulics or pneumatics.

Acknowledgments

This work has been supported by the Computer Science Division of the Office of Naval Research.

References

Chandrasekaran, B. 1994. "Functional Representation and Causal Processes." *Advances in Computers*, Vol. 38. ed. Marshall Yovits. Academic Press.

de Kleer, J. 1984. "How Circuits Work." *Artificial Intelligence* **24**:205-280.

Forbus, K. 1984. "Qualitative Process Theory." *Artificial Intelligence* **24**:85-168.

Forbus, K. and de Kleer, J. 1993. *Building Problem Solvers*. MIT Press, Cambridge, MA.

Forbus, K. and Whalley, P.B. 1994. "Using Qualitative Physics to Build Articulate Software for Thermodynamics Education." In *Proceedings of the Twelfth Annual AAAI Conference*, Seattle, WA.

Franke, D. 1993. *A Theory of Teleology*. Ph.D. dissertation, The University of Texas at Austin.

Heywood, R.W. 1980. *Analysis of Engineering Cycles*, 3rd edition. Oxford: Pergamon Press

Narayanan, N., Suwa, M., and Motoda, H. 1994. "How Things Appear to Work: Predicting Behaviors from Device Diagrams." In *Proceedings of the Twelfth Annual AAAI Conference*, Seattle, WA. AAAI Press.

Schank, R.C. & Abelson R. 1977. *Scripts Goals Plans and Understanding: An Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Lawrence Earlbaum Associates, Publishers.

Van Wylen, G. & Sonntag, R. 1985. *Fundamentals of Classical Thermodynamics*, 3rd edition. New York: John Wiley & Sons.

Vescovi, M., Iwasaki, Y., Fikes, R., and Chandrasekaran, B. 1993. "CFRL: A Language for Specifying the Causal Functionality of Engineered Devices." In *Proceedings of the Eleventh Annual AAAI Conference*, Washington, DC. AAAI Press.

Whalley, P.B. 1992. *Basic Engineering Thermodynamics*. Oxford: Oxford University Press.

[2] P.B. Whalley, personal communication.