N89 - 15553

Functional Reasoning in Diagnostic Problem Solving*

Jon Sticklen [1]
AI/KBS Group - CPS Dept
A714 Wells Hall - Michigan State University
East Lansing, MI 48824-1027

W. E. Bond and D. C. St. Clair [2]
McDonnell Douglas
Research Laboratories
St. Louis, MO 63166

1 Abstract

This work is one facet of an integrated approach to diagnostic problem solving for aircraft and space systems currently under development. The authors are applying a method of modeling and reasoning about deep knowledge based on a *functional* viewpoint. The approach recognizes a level of device understanding which is intermediate between a compiled level of typical Expert Systems, and a deep level at which large-scale device behavior is derived from known properties of device structure and component behavior. At this intermediate functional level, a device is modeled in three steps. First, a component decomposition of the device is defined. Second, the functionality of each device/subdevice is abstractly identified. Third, the state sequences which implement each function are specified. Given a functional representation and a set of initial conditions, the functional reasoner acts as a consequence finder. The output of the consequence finder can be utilized in diagnostic problem solving. The paper also discusses ways in which this functional approach may find application in the aerospace field.

2 Introduction

Over the last decade there has been a calling for deep-level reasoning capabilities [11,15] in knowledge-based systems. This calling has, in general, expressed two intuitions:

---

* Partial support for this research is from the McDonnell Douglas Independent Research and Development program. Dr. Sticklen is also currently supported as an Ameritech Fellow at MSU, and by equipment grants from Apple Computer and Texas Instruments.

1 Part of the research reported here was performed while Dr. Sticklen was associated with the Laboratory for Artificial Intelligence Research, the Ohio State University.

2 Dr. St. Clair is also Professor of Computer Science at the University of Missouri-Rolla Graduate Engineering Center in St. Louis, MO.

1. The first intuition is that domain experts have many layers of understanding about their areas of expertise, ranging from highly *compiled* knowledge to deep-level understanding which is not tuned specifically to a particular problem solving task. The ability to layer understanding has consequences both for problem solving and for the explanation of problem solving.
2. The second intuition is that approaches to knowledge-based systems which rely solely on patterns of data to establish working hypotheses will not, in the final analysis, be robust enough to support large systems. Although there have been convincing arguments that highly compiled systems can, in principle, be complete [7], there nonetheless is a practical difficulty in endowing a system with all of the patterns it is likely to need during problem solving. The central issue is that problem solvers should be able to deal with *novel* situations. If pre-stored patterns of data formed the only cornerstone to problem solving, it would be very difficult to deal with such novelty.

Although there is general agreement that deep reasoning capabilities are desirable, there is no apparent consensus on how to model such deep knowledge, nor on how to reason about it. In fact, there is no clear consensus on what the terms "deep-level understanding" or "deep-level reasoning" mean.

Recently, Sticklen [22,23] developed an integrated approach to diagnostic reasoning in the medical domain. One component of his diagnostic architecture is a deep-level reasoner which acts as a adjunct problem solver to a compiled level unit. The authors of this paper have recently begun to apply and extend Sticklen's approach in the domain of aerospace systems. In this paper, the rudiments of the methodology are described, its utility for aerospace problems is pointed out, and the points of contact between the research reported here and other lines of inquiry are described.

3 Functional Approach

Prior research in deep reasoning methods have typically fallen into two broad areas: the *causal net approach* [16,17,18,26] and the *naive physics approach* [1,9,10,12,13]. The causal net approach centers on representing deep knowledge via a causal net structure. Deep reasoning then means controlling the process of navigating the causal net. The naive physics approach centers on deriving the functionality of a device from the function and structure of its constituent subdevices.

On close examination, the causal net approach appears to be a compiled problem solving approach operating in the representation framework of a semantic net. The naive physics approach is aimed at *deriving* the behavior of a system from its components. However, there is a level of device understanding that is intermediate between compiled level understanding and the level at which composite behavior is derived. That middle ground of device understanding centers on the *known* purpose to which a device is put; i.e., on the function of the device and its subdevices.

Starting from the broad framework of the Generic Task (GT) theory of knowledge based systems [3,4,5], and building on the initial representational framework provided by Sembugamoorthy and Chandrasekaran [20], the present approach to representing and reasoning about devices *via* deep-level knowledge incorporates facets of the two other approaches.

3.1 Underlying Intuitions

Intuition 1: Limitation to Devices. First, concern is limited to the world of *devices*, where a device is defined as "a naturally occurring or artifactual assemblage whose purposes-goals-functions are known." Any physical object can, in principle, be described in terms of its physical properties, but once the *use* of a device is understood, comprehension has progressed to a higher level.

Device level understanding is distinguishable from the "attribute description" level by the indexing capability gained at the device level; e.g., if a car's purpose is understood to be centered around transportation, then its color attribute is irrelevant. The color attribute of a stealth aircraft, on the other hand, is highly relevant to its purpose, and hence would find a place in a functional representation.

Intuition 2: Device Decomposition. Device complexity is managed by *decomposing* the device into a number of components until a level is reached at which one can grasp how subdevices at that level operate. Understanding the operation of the overall device can then be stated in terms of the operation of the components.

Intuition 3: Indexing. As device decomposition proceeds, understanding of the device is organized in terms of the purposes of the device; i.e., in terms of its *functions*. It is important to note that in the functional approach, device functionality is given, **not** derived.

Intuition 4: Composability. Intuition #2 above dealt with decomposing a complex device into a number of subdevices. Intuition #3 is that we organize our understanding about each of those subdevices in terms of their individual functions/goals. Both #2 and #3 can be characterized as dealing with the *static* representation of device understanding. Note, that the complexity of device understanding is managed by a process of decomposition. Thus, the process of reasoning about a particular device in a particular situation must have the ability to dynamically select those parts of device functionality which are relevant to the current situation.

Intuition 5: Information Processing Task. The last intuition concerns the input-output relation of the task set for using the functional representation. In more formal terms, this is called the Information Processing Task (IPT) [14]. The IPT for functional level reasoning is a focused *consequence finding* in which the problem solver is given a set of initial conditions about either the state of the device or the nonavailability of some of the normal functions of the device. The output is the consequent changes that take place in state variables of the device.

3.2 Functional Representation and Reasoning

Building on earlier work by Sembugamoorthy and Chandrasekaran [20], Sticklen [22,23], developed methods of representing and reasoning about functional level phenomena, including a family of languages for representation and a full consequence finding algorithm. A description of this functional approach follows.

3.2.1 Representation

The functional level description of a device includes three core components: a description of device structure, a description of the functions of the device, and a description of the behaviors of the device which carry out given functions. To illustrate the various components of the functional representation, consider the simple device shown in Figure 1 and the partially complete functional description shown in Figures 2 and 3.

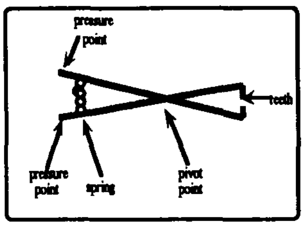

Figure 1: Simple Device - A Clothespin

Function: ***Open*** of ***CLOTHESPIN***
ToMake: teeth-more-open
Provided: (pressure-point-force-applied GreaterThan restoring-force)
By: ***OpenBehavior***

Figure 3: Open Function of Device Clothespin

Device: CLOTHESPIN
SubDevices: (***ARM, PIVOT, SPRING, ...***)
Functions: (***Open, Close, Hold,...***)

Figure 2: Part of Device Specification for Clothespin

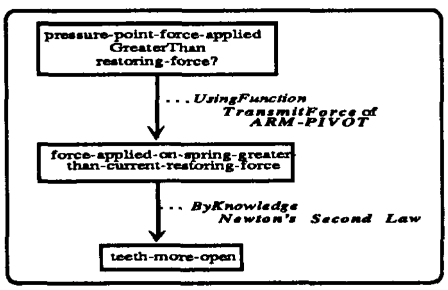

Figure 4: A Behavior of Device Clothspin

The device description shown in Figure 2 includes two items: a listing of the subdevices which compose the current device, and a listing of the functionalities of which the device is capable. It is possible to conceive of a clothespin as made of many different sets of sub-devices. A characteristic of the functional level description is that there are many possible ways of decomposing a device into its components. The decomposition selected depends on the reasoning purposes for which the representation is constructed.

The function description of Figure 3 contains three items:

1. a *Provided* clause which states the conditions under which the function is applicable. The Provided clause may be thought of as a precondition for the behaviors which carry out the function.
2. a *ToMake* clause which states the result(s) that will be obtained if the invoked function completes successfully. The ToMake clause may be thought of as a postcondition for the behaviors that carry out the function.
3. a *By* clause which states the behavior(s) carrying out the function.

The function declarations within the functional description provide a means of knowing what can be achieved (the ToMake clause), what must be true for a given function to be applicable (the Provided clause), and a pointer for where to look if a more detailed description is needed of how the function achieves its goal (the By clause).

The description of behavior at the functional level contains two ingredients, as illustrated in Figure 4.

1. Partial state descriptions of the device. For example, *teeth-more-open* is a *partial* state of the Clothespin device. It does not represent a total state description of the Clothespin.
2. Annotated links between states. These links represent both the temporal flow of one state leading to another, and the *reason* that one state follows another. Note that the annotation could be a pointer to another function or behavior, or to a non-decomposable fragment of world knowledge.

From the above discussion, the following facets of the functional description of a device can be seen.

1. The functional representation of a device is a conceptual abstraction of what a device is and how it works. The "what it is" part is represented as a collection of subdevices related by a ComponentOf relation. The "how it works" is represented as the functionalities of which it is capable and the behaviors that accomplish those functions.
2. The functional description allows a natural modularity in understanding devices. This modularity is important in three specific ways. First, a subdevice of the overall device may be replaced with another totally different subdevice which accomplishes the same functions. Second, in understanding from the top level how the device functions, one is normally led via a chain of

   device => function => behavior => sub-device ...

   to lower and lower levels of sub-devices. However, this path of understanding may be terminated before the lowest levels of the device are reached. Once a level is reached at which a particular functionality of some underlying sub-device may be "assumed true," further probing along the current path is unnecessary. This ability to probe only as far as needed follows directly from the modularity of representation adopted.

Third, and most importantly, since the functional representation is organized around *fragments* of device understanding, the ability to dynamically compose answers is pivotal. Without significant modularity, such composability would not be possible.

3.2.2 Functional Reasoning

As noted above, the Information Processing Task of the functional approach is a type of consequence finding. The consequence finding is undertaken in response to a *particular set of conditions*, and amounts to building up a full state change diagram from the relevant fragments that exist in the behaviors of the functional representation. Sticklen [22] describes the operation of the consequence finding portions of the functional reasoner in full detail. Informally, the consequence finding steps in the functional approach can be described as follows.

First, the current nondefault conditions of the device are input. The consequence finder uses these initial conditions to *index* the applicable functions of the device by examination of the preconditions of each function. Through a filtering operation, the highest level applicable functions are selected as starting points. As one of the selected functions is taken, each implementing behavior is retrieved. A behavior is represented as a directed graph structure having two types of nodes: one representing changes in device state, the other representing a "knowledge pointer" to the reason for the state change. The knowledge pointers are of two types: decomposable and nondecomposable.

Next, the consequence finder recursively expands each one of the decomposable knowledge pointers. Not all functions and behaviors are applicable in a particular situation since all preconditions must be met before a function or behavior can be used. With that selectivity taken into account, the process is not unlike the process of *macro expansion* in typical computer science terms.

Following the second step, a particularized causal net like structure for the current device and situation is produced. Each node in this structure represents a change in a state variable in the device. The overall consequences on the device are then determined by simply traversing the structure and "adding up the results" of all the state changes that have occurred. Moreover, because a record of the preconditions of applied functions/behaviors can be easily maintained, it is also possible to state the consequence finding results in terms of *assumptions* which must hold in order for the derived consequences to be valid.

4 Functional Reasoning in Diagnosis

For diagnostic systems based on compiled classificatory problem solvers [2,6,21,22,24], each diagnostic hypothesis is represented by a classification specialist.

Each of these specialists must be able to establish its own viability, given the particular device/situation under consideration. One way of establishing this hypothesis is to rely on compiled associations in one way or another. To depend solely on such compiled knowledge assumes that novel situations will not be encountered. In general, this assumption is not valid.

The Function Level Consequence Finding strategy, as outlined above, provides a way to *derive* associational links that can be used for hypothesis establishment, and the assumptions that must hold for the links to be valid. In order for a compiled level diagnostic hypothesis to make use of a functional level consequence finder, the diagnostic hypothesis needs one additional knowledge structure - a representation of what the diagnostic hypothesis means in functional terms. Given such initial conditions, the functional reasoning process derives the consequences for the device. At the diagnostic level, *results* of the functionally based consequence finding can be used as patterns which should be present *if* a particular diagnostic hypothesis is valid for the current case. Sticklen has described this interaction between a deep problem solver and a compiled level problem solver in detail [22].

5 Functional Reasoning in Aerospace Applications

To date, the functional approach described has been applied primarily in the medical domain. However, a preliminary examination of several typical aerospace systems (electronic, hydraulic, and a fuel system) indicates that this approach is also applicable to engineered systems, especially as a aid to the design/redesign process, as well as in trouble shooting situations.

During the design stage, an engineer will initially define the high-level functionality of a device. Then the device design is refined by specifying subdevice functionality. When sufficient detail has been developed, specific components and interconnections are chosen to implement the low-level functionality. This process is fully supported by the functional approach described. In addition, the functional approach appears to support specific implementation requirements:

1. The ability to handle systems that exhibit both steady-state and dynamic behaviors (example-both the operation of a fuel system under constant demand conditions as well as the conditions that occur when the engine being supplied goes from 50% to 100% power). Many diagnostic procedures assume a system that is operating under steady-state conditions. Although many failure conditions can be detected using that assumption, other failures occur only during device transients.

2. The ability to identify situations that produce both "hard" failures and also situations where a device fails to perform "to specifications." Many diagnostic systems assume that a failure will occur in a distinct manner. This assumption is incorrect in some instances. Instead, the system may

malfunction by failing to perform to desired specifications or by producing symptoms which are not distinctive.

3. The ability to explicitly address device connectivity. Aerospace systems are often constructed by assembling a large number of interconnected components. The representation of connectivity should be explicit. This representation allows the designer to quickly wire together functional units. It also provides the ability to perform quick reconfigurations for design modifications.

To be successful, the approach must be able to interface with design and test environments. It is anticipated that the functional approach will allow considerations of testability to enter in the design process much earlier.

6 Discussion

In this report, a functional approach to representing and reasoning with deep knowledge of devices has been described, and a method to use the results of functional reasoning in diagnostic problem solving has been sketched. Recently, several other lines of research have been reported which attempt diagnostic problem solving from first principles, for example, see Reiter [19]. This recent body of work is largely an extension of the diagnostic outlook taken by Davis [8] and others who set out to perform diagnostic problem solving directly from a deep level. On the other hand, the functional approach described here integrates both a compiled level and a deep-level problem solver into a single composite unit in which the compiled level problem solver can play the important role of focusing the deep-level problem solver.

Beyond its uses in diagnostic problem solving for aerospace systems, a functional approach can also be applied to some of the problems associated with the issue of design knowledge capture [25]. Design knowledge capture is the process of organizing design knowledge in a machine-interpretable form. Design knowledge is composed of the specific design solution description together with the underlying rationale for the solution. The design description defines what the solution does and why it does so. The rationale includes the information used by the designers, the analyses that were performed, and the decisions that were made to produce the solution. Because the central aspect of the functional approach is the organization of knowledge about how a device works, it directly provides a description of what the device does and how it performs its function. It also provides an active representation which can be used in the future to simulate the operation of the device or to perform diagnostics.

7 References

1. Bylander, T. 1986. "Consolidation: A Method for Reasoning about the Behavior of Devices." Ph.D. The Ohio State University, Columbus, Ohio.

2. Chandrasekaran, B. 1982. "Decomposition of Domain Knowledge into Knowledge Sources: The MDX Approach." *Proc. 4th Nat. Conf. Canadian Society for Computational Studies of Intelligence.*

3. Chandrasekaran, B. 1983. "Towards a Taxonomy of Problem-Solving Types." *AI Magazine*, 9-17.

4. Chandrasekaran, B. 1985. "Generic Tasks in Knowledge-Based Reasoning: Characterizing and Designing Expert Systems at the "Right" Level of Abstraction." *Proceedings of The IEEE Second Annual Conference on Artificial Intelligence Applications*, IEEE.

5. Chandrasekaran, B. 1986. "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design." *IEEE Expert*, 23-30.

6. Chandrasekaran, B., F. Gomez, S. Mittal and J. W. Smith. 1979. "An Approach to Medical Diagnosis Based on Conceptual Structures." *Proceedings of IJCAI 6.*

7. Chandrasekaran, B. and S. Mittal. 1983. "On Deep versus Compiled Knowledge Approaches to Medical Diagnosis." *International Journal of Man-Machine Studies*, 425-436.

8. Davis, R. 1984. "Diagnostic Reasoning Based on Structure and Behavior." *Artificial Intelligence*, 347-410.

9. de Kleer, J. and J. S. Brown. 1984. "A Qualitative Physics Based on Confluences." *Artificial Intelligence*, 7-83.

10. Forbus, K. D. 1984. "Qualitative Process Theory." *Artificial Intelligence*, 85-168.

11. Hart, P. 1982. "Directions for AI in the 80s." *SIGART News Letter.*

12. Kuipers, B. 1984. "Commonsense Reasoning about Causality: Deriving Behavior from Structure." *Artificial Intelligence*, 169-203.

13. Kuipers, B. and J. P. Kassirer. 1984. "Causal Reasoning in Medicine: Analysis of a Protocol." *Cognitive Science*, 363-385.

14. Marr, D. 1982. "Vision." W.H. Freeman.

15. Michie, D. 1982. "High-road and Low-road Programs." *AI Magazine*, 21-22.

16. Patil, R. S. 1981. "Causal Representation of Patient Illness For Electrolyte And Acid-Base Diagnosis." Ph.D. M.I.T., Cambridge, MA.

17. Pople, H. 1977. "The Formation of Composite Hypotheses in Diagnostic Problem Solving: An Exercise in Synthetic Reasoning." *Proceedings of IJCAI 5*, 1030-1037.

18. Pople, H. 1979. "Heuristic Methods for Imposing Structure On Ill Structured Problems: The Structuring of Medical Diagnosis." In *Artificial Intelligence In Medicine*, P. Szolovits. 119-189. Westview Press.

19. Reiter, R. 1987. "A Theory of Diagnosis from First Principles." *Artificial Intelligence*, 57-95.

20. Sembugamoorthy, V. and B. Chandrasekaran. "Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems." In *Experience, Memory, and Learning*, J. Kolodner and C. Reisbeck. Lawrence Erlbaum Associates.

21. Smith, J. W. 1986. "RED: A Classificatory and Abductive Expert System." Ph.D. The Ohio State University, Columbus, Ohio.

22. Sticklen, J. 1987. "MDX2: An Integrated Medical Diagnostic System." Ph.D. The Ohio State University, Columbus, Ohio.

23. Sticklen, J. and B. Chandrasekaran. 1985. "Use Of Deep Level Reasoning in Medical Diagnosis." *Proceedings of The Expert Systems in Government Symposium*.

24. Sticklen, J., B. Chandrasekaran and J. R. Josephson. 1985. "Control Issues in Classificatory Diagnosis." *Proceedings of IJCAI 9*.

25. Sticklen, J. and B. Chandrasekaran. 1988. "Using a Functional Approach for Knowledge Acquisition." *Proceeding of 4th Banff Conference on Knowledge Acquisition*, in press.

26. Weiss, S., C. Kulikowski and A. Safir. 1977. "A Model-Based Consultation System for the Long-Term Management of Glaucoma." *Proceedings of IJCAI 5*, 826-832.