AAAI 1992 Spring Symposium Series
Computational Considerations in Supporting
Incremental Modification and Reuse
March 25-27, 1992

Classifying and Indexing Design Modifications via Descriptions of Purpose*

David W. Franke

Department of Computer Sciences
University of Texas at Austin Austin, Texas 78712
franke@cs.utexas.edu

Abstract

Information regarding the structure and behavior of a mechanism (design) is readily captured in current design support systems and methodologies. While structure and behavior descriptions can be used to index design modifications, we claim that a more productive classification of design modifications for explanation and reuse is achieved via descriptions of purpose of these design modifications. This research includes a language for descriptions of purpose (teleological descriptions), definitions of behavior abstraction, and a classification technique based on the teleology language and behavior abstraction. The teleology language and behavior abstraction permit generalization of teleological descriptions that subsequently provide for the identification of potential design modifications beyond the original intent of the modification. This generalization can span mechanism domains such as electrical, thermal, and hydraulic, thereby providing a technique for selecting potential analogies to be applied to the current design problem.

1 Introduction

Teleological descriptions capture the *purpose* of an entity, mechanism, or activity with which they are associated. When one examines human-generated descriptions of systems or mechanisms, one finds that they are rich with descriptions of purpose, as well as descriptions of structure, behavior and causality. Descriptions of purpose are very valuable in communicating and understanding designs, since they convey an important aspect of the design process, namely the designers' intent. We propose that it is important to represent and reason with teleological descriptions in the problem solving tasks of explanation, design, and diagnosis.

*This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-8602665, IRI-8905494, and IRI-8904454, by NASA grants NAG 2-507 and NAG 9-200, and by the Texas Advanced Research Program under grant no. 003658175.

The approach described herein is motivated by the observation that design is a sequence of modifications [2] (to an initial design) by which a designer attempts to meet a set of design goals and requirements. This design process model includes the redesign of mechanisms whose specifications have changed. In particular, we have focused on goals and requirements that can be expressed in terms of behaviors. These behaviors can be attributed to the designed artifact itself, or to processes that create or utilize the artifact, such as its manufacture. The language for teleological descriptions given here (TeD) expresses the purpose a design modification in terms of behaviors prevented or guaranteed by the modification.

As an example of our approach, consider an electrical engineer designing a circuit. The engineer begins with specifications describing the desired static (e.g. size) and dynamic (behavior) characteristics of the resulting circuit design. For example, one behavior specification for the input selection circuit is "invert the data signal when the control signal is high (logic true) and leave the output unchanged when the control signal is low (logic false)". The engineer also begins with specifications from the domain of CMOS circuit design, such as "the input value to a logic gate should not maintain a steady, intermediate value (voltage) between low and high, causing the gate to consume power by allowing current to flow".

Interacting with a design system, the engineer proposes a design, evaluates the design with respect to the specifications, and makes modifications to the design in an attempt to make the design meet the specifications. In this interaction, teleological descriptions can be acquired and subsequently used to assist the engineer in completing the design, as demonstrated in the following example.

Engineer: Generate the possible behaviors of design 1 (*see schematic in Figure 1a.*) for signal `ctl` high and signal `in` transitioning from low to high.
Computer: The possible behaviors are: (*displays the behaviors*).
Engineer: Evaluate the behaviors with respect to the design specifications. **Computer:** For signal *in*, the design does not meet specification 5: "the input value to a logic gate should not maintain a steady, intermediate

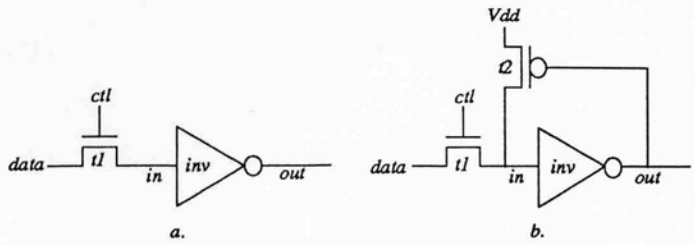


Figure 1: CMOS Input Selection Circuit - Schematics

value (voltage) between low and high, causing the gate to consume power by allowing current to flow". (*If the design system has a database of teleological descriptions and a teleological description referencing this specification exists in the database, a recommendation for modifying the design can be made.*)
Engineer: (*Modifies the design by adding feedback transistor t2, as shown in the schematic in Figure 1b.*) Generate the possible behaviors of design 2 for signal `ctl` high and signal `in` transitioning from low to high.
Computer: The possible behaviors are: (*displays the behaviors*).
Engineer: Evaluate the behaviors with respect to the design specifications.
Computer: The design meets all specifications. The purpose of the design modification transforming design 1 into design 2 is to guarantee specification 5.

2 Teleology Language

Structure and Behavior. Structure languages used here describe models (designs) hierarchically composed from simpler models called components. Hierarchy is achieved by applying this decomposition recursively to components. Each component has an interface, expressed in terms of terminals, that can be connected to terminals of other components. At some point in the hierarchy, components are described in terms of the primitives of the associated behavior language. We will call these primitives *variables* and *behavior constraints*. Behavior is represented as a sequence of states, where a state assigns values to variables defined by the structure.

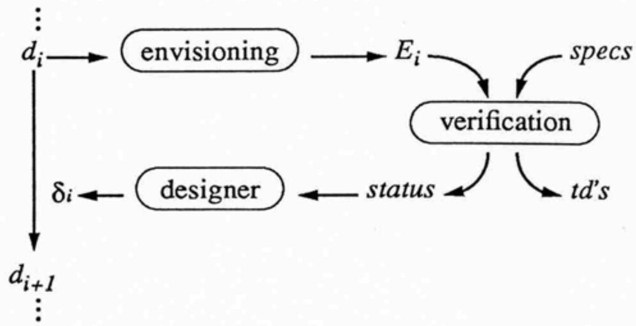
The structure language provides a means for describing a single point in the history or evolution of a design. A description of *design history* also requires a means for describing the transitions from one state of the design (distinct from a behavior state of an artifact or instance of the design) to another. The term *design modification* denotes such a transition. A design history is a pair comprised of an initial design and a sequence of design modifications, denoted

$$(d_0, \langle \delta_1, \delta_2, \ldots, \delta_n \rangle)$$

where d_0 is the initial design and δ_i are design modifications. This design history defines a sequence of designs

$$d_0, d_1, \ldots, d_{n-1}, d_n$$

where d_i is the result of applying design modification δ_i to design d_{i-1}. The design history is captured during initial design, evaluation, and design modification. Figure 2 gives a detailed process description for the evaluation (envisioning[1] and verification) and design modification (designer) steps.



d_i - i^{th} version of the design
E_i - envisionment for design d_i
specs - design specifications
td's - teleological descriptions captured in verification
status - results of verification
δ_i - modifications generated by the designer

Figure 2: Design Process Flow (Single Step)

Teleology. The teleology language relates design modifications (changes in structure) to design specifications (desired static and dynamic characteristics). With the teleology language, we can formally express the designer's intent in modifying a design, namely to guarantee some design specification. Teleological operators are the language primitives for teleological descriptions. In the context of a design modification, a single teleological operator relates the unmodified design to the modified design in terms of the *specification predicates.*[2] Let ϕ be a specification predicate, d and d' be designs, δ a design modification such that d' is the design obtained by applying δ to d, and E and E$'$ be the envisionments of d and d', respectively. We define the teleological operator **Guarantees** as:

$$\delta \textbf{ Guarantees } \phi \;\Leftrightarrow\; \begin{cases} \exists\, b \in \mathrm{E}, \;\; \neg\phi, \\ \forall\, b' \in \mathrm{E}', \;\; \phi. \end{cases}$$

[1]Envisioning is the procedure for generating an envisionment, the set of behaviors exhibited by a mechanism (or its design).

[2]Specification predicates express properties of a design which are desired, and verification of a design determines whether these properties hold for the design.

3 Behavior Abstraction

Design specifications most often address only a single aspect or small number of aspects of the artifact to be designed, such as the physical dimensions (length, width, height) or the behavior of some portion of the artifact. Consequently, one needs to represent and reason about behavior descriptions that reference only part of the artifact. Further, the design specification may be given in terms more general than the details of the designed artifact, such as stating that a particular variable value should always be positive, although the specific positive value is unspecified.

A relation between behaviors, $\sqsubseteq_\sigma$ (read "is less general than or equal to"), is defined [6, 5] and shown to partially order the space of behavior abstractions. The basic idea of behavior abstraction is to map partial states in the abstraction to states in the behavior such that state ordering is preserved and a partial state abstracts the state to which it is mapped. This relation is defined in terms of subordinate relations (also shown to partially order their respective spaces) on variable names and types, variable values, and states. The relation $\sqsubseteq_\sigma$ is the basis for classification of teleological descriptions.

4 Classification

Classification of teleological descriptions and their associated design modifications is based on the $\sqsubseteq_\sigma$ relation defined for behavior. An index structure (a partial lattice based on $\sqsubseteq_\sigma$) is constructed to classify and access behavior abstractions and associated teleological descriptions. An initial index structure is built from behaviors involving a single state and variable. Possible types for this variable are taken from the generic variable types of bond graph modeling of dynamic physical systems [10]. Variable types are further specialized to domain specific types such as voltage and current in the electrical domain. Abstract (qualitative) variable values are also considered to generate the initial index structure. In particular, the qualitative values `(minf 0)`, `0`, and `(0 inf)` from the quantity space `(minf 0 inf)` (sometimes called `+ 0 -`) and the possible qualitative values from the quantity space `(minf - 0 + inf)` are considered.

Abstract behaviors referenced by teleological descriptions are added to the lattice in the natural way, namely starting with the most general behavior description, determining whether the behavior is less general than the lattice node, and moving down (to more specialized behaviors) recursively until the most specific generalizations of the behavior have been found. For a given behavior, the current implementation classifies the full behavior, a generalization of the behavior in which all values are abstracted to values in the `(minf - 0 + inf)` quantity space, and a further generalization in which all variable types are abstracted to their generic type (effort, flow, etc.). We select these generalizations out of the many that can be generated for two reasons. First, we abstract scenarios away from specific quantity spaces since quantity spaces are likely to differ in landmark names and landmark order. This abstraction provides a connection or grouping in the lattice for scenarios containing variables of the same domain specific types with similar time varying behavior. Second, we abstract away from domain specific variable types to provide a connection or grouping among scenarios containing variables of the same generic types with similar time varying behavior. This second abstraction provides the basis for design reuse across domains, namely using a teleological description originally captured in one domain (e.g. hydraulic) when designing in another domain (e.g. thermal).

5 Queries

The full index structure in this research provides three perspectives on the database of teleological descriptions,

- specification predicates and their abstractions,
- design history (modification sequence for a design), and
- teleological operators.

Based on this index structure, we describe query or use scenarios of the teleological description database for explanation and design reuse.[3] These queries answer to following questions:

- Explanation: "What's the purpose of component X?"
- Design Reuse: "How have previous designs addressed specification ϕ?"

For presentation purposes, we express these queries via the following Prolog predicates:

- `td(`*design-history*`,` *modification*`,` *operator*`,` *spec*`)`
- `references(`*modification*`,` *component*`)`
  Succeeds when *modification* references structural entity *component*.
- `msg(`*spec*`,` *generalization*`)`
  Succeeds when *generalization* is a most specific generalization of specification *spec*.
- `mgs(`*spec*`,` *specialization*`)`
  Succeeds when *specializations* is a most general specialization of specification *spec*.

Explanation Queries. For explanation, queries of the database of teleological descriptions are restricted to those descriptions involving modifications from the design

[3] Queries supporting diagnosis are supported also.

history of the mechanism being examined. In this case, queries are of the form

$$\texttt{td}(d, \texttt{Mod}, \texttt{Op}, \texttt{Spec}), \texttt{references}(\texttt{Mod}, c) \quad (1)$$

where d is the design of the mechanism being examined, and c is the component or parameter for which an explanation is desired. The predicate `references` selects those modifications that involve c in some way.

Reuse Queries. In a design reuse context, the designer is faced with the problem of modifying a design so that it will meet the specification predicates. Hence, the initial query made by the designer will be one based on a specification predicate of the design. While one might also specify the appropriate teleological operator (**Guarantees**, if the specification predicate is to hold everywhere, or **Prevents** if the specification predicate is to be prohibited), retrieving modifications based solely on the specification predicate can be of interest to the designer, whether the teleological operator is **Guarantees**, **Prevents**, or **Introduces** [5, 6]. For example, a design modification that has introduced the specification predicate in a previous design may be of interest when the reusing designer is attempting to prevent the specification predicate from holding because the reusing designer may be able to reverse the design modification that introduced the predicate and hence prevent it. This base query would be

$$\texttt{td}(\texttt{Dh}, \texttt{Mod}, \texttt{Op}, \phi) \quad (2)$$

where ϕ is the specification predicate of interest.

It is likely, however, that the exact specification predicate does not appear in the database, in which case the designer would like to retrieve design modifications (i.e. teleological descriptions) concerned with specification predicates "close to" the one of interest. This "closeness" property is realized by generalization and specialization links among specification predicates in the database of teleological descriptions. Consequently, the initial set of teleological descriptions retrieved for potential reuse should involve the most specific generalizations (`msg`) and the most general specializations (`mgs`) of the specification predicate of interest, ϕ. Note that if ϕ appears exactly in the database, this set will be $\{\phi\}$, the specification predicate itself. This modified query becomes the union of

$$\texttt{msg}(\phi, \texttt{Spec}), \texttt{td}(\texttt{Dh}, \texttt{Mod}, \texttt{Op}, \texttt{Spec}) \quad (3)$$

and

$$\texttt{mgs}(\phi, \texttt{Spec}), \texttt{td}(\texttt{Dh}, \texttt{Mod}, \texttt{Op}, \texttt{Spec}). \quad (4)$$

To constrain the query with respect to a specific teleological operator (e.g. **Guarantees**), one can use the queries

$$\texttt{td}(\texttt{Dh}, \texttt{Mod}, \texttt{guarantees}, \phi) \quad (5)$$

or

$$\texttt{td}(\texttt{Dh}, \texttt{Mod}, [\texttt{conditionally}, \phi_1, \texttt{guarantees}], \phi). \quad (6)$$

6 Related Work

Representation and acquisition of descriptions of purpose has been addressed by de Kleer (EQUAL [3]), Sembugamoorthy, Chandrasekaran, Goel, Sticklen, and Bond (Functional Representation [8, 11, 12]), NASA Ames Research Center (CDK Project [1]), Downing (BIOTIC [4]), and Gruber (ASK [9]). While each associates a purpose with structural entities, only the CDK and ASK systems explicitly reference and examine design modifications and design alternatives. The teleology language (TeD) developed in this work provides a formal representation language that can express descriptions of purpose captured in these other approaches. Additionally, TeD provides a means for effectively classifying and indexing these descriptions for use in explanation and design reuse, and provides a basis for acquiring teleological descriptions interactively during design [7].

7 Summary

The capabilities described here have been implemented and demonstrated on examples from the electrical, hydraulic, thermal, and mechanical domains. In addition, a simple acquisition capability has been implemented.

We believe that this work represents an advance in the representation, acquisition, and application of teleological knowledge, and will play a role in future explanation, diagnosis, design reuse, and design by analogy systems.

References

[1] Catherine Baudin, Cecilia Sivard, Monte Zweben, "Model-Based Approach to Design Rationale Conservation", in *Proc. of 1989 Workshop on Model-Based Reasoning*, Detroit, August 1989, pp. 88-90.

[2] B. Chandrasekaran, "Design Problem Solving: A Task Analysis", in *AI Magazine*, Vol. 11 No. 4 (Winter 1990), pp. 59-71.

[3] Johan de Kleer, "How Circuits Work", in *Qualitative Reasoning About Physical Systems*, Daniel G. Bobrow, ed., The MIT Press, Cambridge, MA 1985, pp. 205-280.

[4] Keith Downing "The Qualitative Criticism of Circulatory Models via Bipartite Teleological Analysis", in *Proc. of 1990 Workshop on Qualitative Reasoning.*

[5] David W. Franke, "Deriving and Using Descriptions of Purpose" in *IEEE Expert*, Vol. 6, No. 2 (April 1991), pp. 41-47.

[6] David W. Franke, "A Theory of Teleology", Ph.D. Dissertation, University of Texas at Austin, Feb. 1992.

[7] David W. Franke, "Acquisition of Teleological Descriptions", in *Applications of Artificial Intelligence X: Knowledge-Based Systems*, April 1992, Orlando.

[8] Ashok Goel, B. Chandrasekaran, "Functional Representation of Designs and Redesign Problem Solving", in *Proc. of the* 11^{th} *IJCAI*, August 1989, Detroit, pp. 1388-1394.

[9] Thomas Gruber, "Learning Why by Being Told What", in *IEEE Expert* Vol. 6, No. 4 (August 1991), pp. 65-75.

[10] Ronald C. Rosenberg, Dean C. Karnopp, *Introduction to Physical System Dynamics*, McGraw-Hill, New York, 1983.

[11] V. Sembugamoorthy, B. Chandrasekaran, "Functional Representation of Devices as Deep Models", in *SIGART Newsletter*, No. 93 (July 1985), pp. 21-22.

[12] Jon Sticklen, B. Chandrasekaran, W. E. Bond, "Applying a Functional Approach for Model-Based Reasoning", in *Proc. of 1989 Workshop on Model-Based Reasoning*, Detroit, August 20, 1989, pp. 165-176.

A Characterization of the Work

1. Characteristics of the Particular Modification Task
   (a) *Domain*: Explanation, design reuse, redesign of physical devices (electric, hydraulic, etc.). We conjecture this approach will work for software design.
   (b) *Task*: Inputs – design specifications, proposed designs and design modifications, verification procedures for the design domain, database of previously acquired teleological descriptions; Output – status of the design with respect to specifications, explanations of modifications and recommendations for modifications, teleological descriptions relating design modifications to specifications classified in a database.
   (c) *Environment*: No particular requirements.
   (d) *Incentives for Modification*: Incremental modification for new and reused designs is the common practice in these domains (physical mechanisms).
   (e) *Tradeoffs*: This approach requires precise specifications and in return provides effective techniques for capture and use of modifications in the form of teleological descriptions.

2. Characteristics of the Approach
   (a) *Autonomy*: We attempt to automate the acquisition and subsequent use of descriptions of purpose, including explanation and modification recommendation.
   (b) *Assumptions*: Precise specifications describing the desired static (e.g. physical dimensions) and dynamic (behavior) characteristics of the mechanism to be designed. For automated acquisition, verification techniques are required.
   (c) *Generic Information*: Design history, including teleological descriptions.
   (d) *Knowledge Use*: Teleological descriptions are retrieved with respect to the specifications and modifications they reference, and are presented as explanations or recommendations.
   (e) *How is Modification Conducted*: By human designer or automated design agent.
   (f) *Domain Dependence*: No domain dependence, where domain ranges over electrical, hydraulic, thermal, mechanical, acoustic.
   (g) *Validation*: By verification techniques of the particular domain.

3. Overall Model
   (a) *Fit*: The elements required to capture and use teleological descriptions, namely modifications, specifications, and verification procedures are the elements of the teleology language and the design process model.
   (b) *Generality*: The representation for teleology can be applied in explanation, design reuse, redesign, design by analogy, case-based reasoning, and diagnosis.
   (c) *Rationality*: Correctness and optimality of modifications and modified artifacts is determined by verification procedures and the designer.
   (d) *Efficiency*: While complete specifications for a design may be very large, they are hierarchically structured, and detailed specifications though numerous are concise.
   (e) *Evaluation*: We prove properties of teleological descriptions, such as "A modification *guarantees* generalizations of specification predicates".
   (f) *Comparison*: The teleology language provides a formal representation language that can express descriptions of purpose captured in these other approaches, a means for effectively classifying and indexing these descriptions for use in explanation and design reuse, and a basis for acquiring teleological descriptions interactively during design.