Proceedings of DETC2000:
ASME International Design Engineering Technical Conferences
September 10-13, Baltimore, Maryland

DETC2000/DTM-14550

A MODEL FOR INFORMATION FLOW IN DESIGN

Steven B. Shooter*
Department of Mechanical Engineering
Bucknell University
Lewisburg, PA 17837
shooter@bucknell.edu

Walid T. Keirouz
Department of Computer Science
Lebanese American University, Byblos Campus
475 Riverside Dr., #1845
New York, NY 10115

Simon Szykman
National Institute of Standards and Technology
100 Bureau Dr., Stop 8262
Gaithersburg, MD 20899-8262

Steven Fenves+
National Institute of Standards and Technology
100 Bureau Dr., Stop 8262
Gaithersburg, MD 20899-8262

ABSTRACT

The complexity of modern products and design tools has complicated the exchange of design information. It is widely recognized that the capture, storage, and retrieval of design information is one of the major challenges for the next generation of computer aided design tools. This paper presents a model for the flow of design information that supports a semantics-based approach for developing information exchange standards. The model classifies design information into various types, organizes these types into information states and levels of abstraction, and identifies the various transformations that operate between the information states. The model is then applied to an example of a transmission for a cordless drill.

1 INTRODUCTION

Traditionally, design was undertaken by a small team of designers operating out of a single location. The team captured design information as notes and sketches in logbooks and as design drawings. As a result, team members could easily exchange the relevant design information.

The exchange of design information is now much more difficult given the complexity of modern products and design processes. At present, product realization may be a collaborative effort among teams operating at different geographical locations. Design information now comes in many forms and is generated or transformed by a wide variety of computer-based tools. However, these tools are typically used during the latter stages of design and are mostly geometry-oriented. They store information that is the outcome of design activities with little regard to capturing the information produced during the development of the design. Furthermore, these tools essentially limit exchange to geometry-related information. The shortcomings of these tools provide fertile ground for misunderstandings between participants in a product realization effort.

The inadequacy of the current generation of computer-based design tools is apparent in the popular drafting packages. These packages implicitly assume a bottom-up approach to design and require detailed geometric information about components before these components can be combined into complete systems. As a result, these tools provide little support for top-down concept ideation.

It is anticipated that the next generation of design tools will address these shortcomings and will operate throughout the entire design life cycle of an artifact including its maintenance, upgrades, etc. The OpenADE project at the National Institute of Standards and Technology (NIST) is addressing design information interchange and agent interoperability issues within the context of a "Collaborative Design Framework" (Lyons et al., 1999, Angster et. al, 1998). In such a framework, shown in Figure 1, distributed teams of designers, production engineers, etc., develop products. These teams use heterogeneous

* This work was performed while Professors Shooter and Keirouz were guest researchers at the National Institute of Standards and Technology.

+ Senior Research Associate, University Professor Emeritus of Civil and Environmental Engineering, Carnegie Mellon University systems, both software and hardware, to generate or transform design information. Furthermore, these teams use a global network to exchange design information and to collaborate on the product development effort. Figure 1 shows the OpenADE Interface that facilitates the communication of design information among design agents. Central to this effort is the development of communication and storage protocols for design information. The Integrated Design Resource Database is intended to store information on design case studies, component data and other resource information about the design. The Design Evolution Product Database is intended to capture information about the design process.

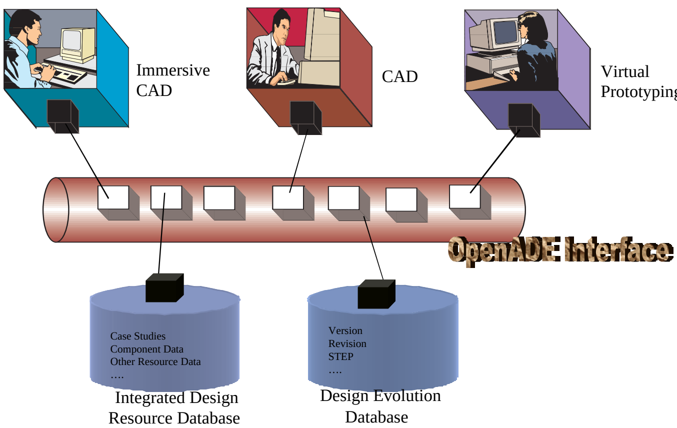

Figure 1: The OpenADE Framework

Future design tools must be able to share design information at a much higher level of abstraction than the current generation of tools does. The next generation of standards must address the issue of sharing design information at these higher levels of abstraction. An exchange of design information between tools is possible only if these tools share semantics. The information exchange will consist of the exchange of data along with an indication of the semantics of the data being exchanged.

To enable this information exchange, there is a need to: (1) formalize the semantics of design information, and (2) standardize the exchange of this semantic information. This formalization is already under way. The NIST Design Repository Project (Szykman, 2000a) is formalizing the semantics of product development information (e.g. product structure, product function, etc.). There is a need to extend this semantic model to cover all design information within the life cycle of an artifact.

The effort described in this paper is intended to characterize the flow of design information. The flow of design information is distinguished from a design process model because information flows regardless of the sequence of individual design activities. Process models are intended to prescribe a suggested approach to design. However, individual design teams can follow a multitude or a mixture of design process models. For example, designers may develop a design alternative without first establishing formal specifications or customer needs. However, specifications and customer needs are the type of information that may be formulated in the process. Through recognition of the states and formats of design information, it will lead to techniques and data structures for adequately capturing, storing, and retrieving it. A description of that effort is described in Szykman et. al. (2000b).

The paper begins with a description of the design information flow model. After introducing the model, the paper characterizes information transformations *within* a level of abstraction. It also describes how information is transformed *between* levels of abstraction. The paper then describes the use of this model in an example of the design of a gear transmission for the Black & Decker VP840 cordless drill.

2 DESIGN INFORMATION FLOW MODEL

Design is a complex activity and design processes vary widely from one organization to another reflecting the cultures of design teams. Furthermore, researchers in design theory disagree as to the nature of design methodologies (Kalay, 1999). As a result, the modeling of design information needs to support a wide range of approaches to design without imposing undue burden on any such approaches.

The model for the flow of design information presented here identifies abstractions and requirements that can be used to develop object models that support a wide variety of design methodologies. The model classifies design information into various types, organizes these types into information states and levels of abstraction, and identifies the various transformations that operate between these information states. Design information is simply defined as the data generated or transformed during a product development effort.

The information flow model assumes that design activities operate in two modes, *iterative* and *layered*, that are deeply intertwined. The iterative mode of the design process accounts for the various feedback loops that occur as designers seek to satisfy design goals. Furthermore, designers develop solutions to a design problem by reasoning about it at various levels of abstraction. The layers in the design process correspond to these levels of abstraction. Abstraction simply means the absence of detail. A level of abstraction is a view of a design problem that includes only the issues designers are considering at a given time in the design process.

Designers continuously shift between these two modes with minimal effort and accumulate information generated at various levels of abstraction. Design information can be viewed as following a spiral that is laid out along the surface of a cone as shown in Figure 2. This spiral is wide at the bottom reflecting the size of the search space; it also starts out to be thin reflecting the available information at the start of the design process. The spiral gets narrower as the design process converges towards a design; the line gets fatter as designers accumulate design information. The path jumps from one layer in the spiral to another when designers achieve an insight into the design. The path also falls down from one layer to another below it when designers reach a dead end and must re-examine past decisions.

Design as Information Transformation and Generation

Fundamentally, design activities operate on information, namely the description of the product being designed. The outcome of a design effort is information on what a product is, what it looks like, what it is made of, how it functions, how it should be manufactured, etc. The *artifact* representing the physical entity being designed is at the center of this information. It is described in terms of *form*, *function* and *behavior*. These terms have been used widely with slight variations of definition. It is, therefore, useful to explain their use in this paper.

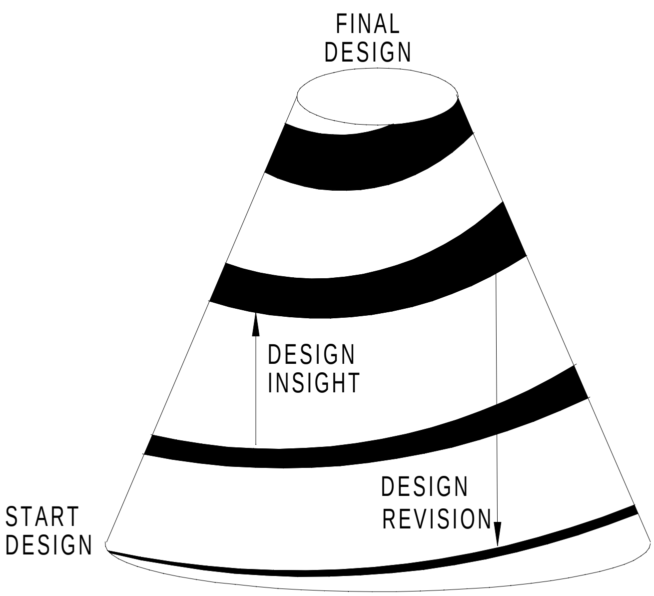

Figure 2: Design Information Development

- The artifact's form represents its physical characteristics and includes, among others, its geometry and material properties.
- The artifact's function represents what the artifact is supposed to do. An artifact satisfies engineering requirements through its function. Function is often used synonymously with *intended behavior*.
- The artifact's behavior is how the artifact implements its function and is governed by engineering principles. These engineering principles may be incorporated into a behavior or causal model that can be used to describe or simulate the artifact's observed or actual behavior based on its form. The behavior model allows designers to explain how function is achieved.

In the context of this work, the artifact is considered to be a collection of symbols which stand for the artifact's characteristic properties, be they form, function, or behavior-related. Relationships between these symbols ultimately encode the artifact's behavior model. Design activities involve actions upon the symbols that describe the artifact primarily in two steps: (1) identify the symbols and their inter-relationships, (2) bind values to the symbols to incrementally add to the artifact's description until the design is complete. Iteration is accomplished by unbinding values that were bound at some previous point or reconsidering the original symbols established.

The complex and indirect relationships among an artifact's form, function, and behavior make design difficult. While designers design an artifact with function in mind, they do so indirectly. Designers cannot specify function directly and have no control over the laws of physics. As a result, they also do not specify an artifact's behavior directly. Instead, they try to achieve a desired function by specifying the artifact's form, which, in turn, drives the artifact's behavior.

The complex array of symbols surrounding the description of an artifact is often unwieldy for all but the simplest design problems. Designers must therefore use levels of abstraction to control the complexity of the design problem and limit the symbols under consideration. The design process involves transformations of symbols within a level of abstraction and transformations of symbols between levels of abstraction.

3 TRANSFORMATION WITHIN A LEVEL OF ABSTRACTION

While design theorists differ on the details of the design process, design models generally agree on the general flow of information from the recognition of customer needs through design generation at various levels of detail, with ongoing evaluation and culminating with a final evaluation. It is clear that these design stages are not distinct, but involve iteration. In fact, the information at each of the stages evolves throughout the design development. This evolution relates directly to the consideration of levels of abstraction that are a part of the dynamic design enterprise. Therefore, one can consider each of the stages within a given level of abstraction. The symbols associated with the information at a given level of abstraction represent the design under consideration at that point of time. The symbols may be a subset of the final design or a different set entirely.

Figure 3 identifies the *states* of information within a single abstraction level. These states are differentiated based on whether certain types of information have been created. The branches in the state diagram denote the flow of information from one state to another. Design activities transform design information and move this information from one state to another.

Note that the arrows in the state diagram only indicate the flow of information from one state to another. The text labels attached to the arrows indicate design activities that may create such design information. The design activities that achieve the state-to-state transitions and the order in which these transitions occur are determined by the product development process. The design information flow model presented in this paper is purely descriptive of how design information is transformed as design progresses. This model can be used to support various activity and process models, but does not seek to prescribe any specific process model.

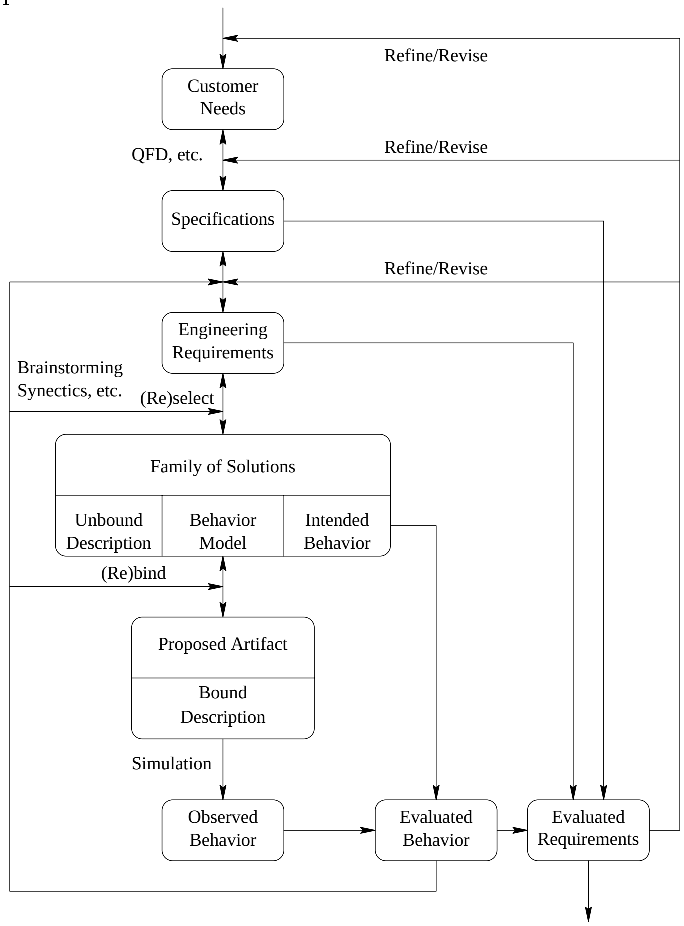

Figure 3: Design Information Flow Model within a Level of Abstraction

A forward walk-through of Figure 3 yields the following. Design information comes into being in the *Customer Needs* state when customers describe their need for a product. The information reaches the next state, *Specifications*, when designers and customers formalize the customers' needs into evaluation criteria. Information in the *Engineering Requirements* state formalizes and details the requirements that the artifact must satisfy from the design team's point of view. To reach the *Family of Solutions* state, information must include one or more partial descriptions of a proposed design. A description is complete at a given level of abstraction when information reaches the *Proposed Artifact* state. In the *Observed Behavior* state, information includes the artifact's behavior as derived from its description. It is likely that designers will generate multiple proposed artifacts. The diagram does not include all of these in the interest of clarity. Answering the question "does the proposed artifact's behavior match its intended behavior?" transitions information into the *Behavior Evaluation* state. Designers use this answer to decide, among other things, whether they should further evaluate the proposed artifact, refine the artifact description, or develop an alternate conceptual solution. Design information in the *Requirements Evaluation* state includes an answer to the question "does the proposed artifact satisfy the engineering requirements?". Note that back-arrows in the state diagram indicate that designers may decide that a given transition has produced an unacceptable result. For example, design information shifts back to the Customer Needs state if designers decide the current specifications cannot be met. The various states are described in more detail further below.

3.1 Customer Needs

Design information reaches the Customer Needs state when customers describe their need(s) for a product. Customers describe their needs in their own languages using both formal and informal terms.

In the context of design information, a customer is a stakeholder in the outcome of the product design process. These stakeholders include, among others, end-users, suppliers, buyers, manufacturing engineers, maintenance departments, and marketing departments.

3.2 Specifications

Design information moves into the Specifications state when the designers translate the customer needs into specifications expressed in formal technical terms. The specifications relate measurable properties of the artifact to allowable value ranges or limits, and are characterized by metric-and-value pairs. The metric describes the desired property and the value quantifies its acceptable levels.

Designers use several techniques, including Quality Function Deployment (Hauser and Clausing, 1988), to develop the specifications. The intent of the specifications is to capture the voice of the customer into measurable characteristics that are useful for selecting among solution alternatives.

Ideally, designers and customers will communicate throughout this transformation. The designers may also negotiate with the customers to revise the customers' needs should they believe that these needs are not realizable.

3.3 Engineering Requirements

Designers transform the specifications into Engineering Requirements. This transformation introduces the notion of an *artifact* as the solution to the design problem—meeting the specifications—and reduces this problem to finding an artifact that satisfies the requirements. The requirements state what the artifact should do and look like, etc., to meet the specifications from the designers' perspective. They express the artifact's needed functionality as relationships between the artifact's properties and as conditions that must be satisfied by these properties. The requirements will refer to a subset of the artifact's properties that will typically include the artifact's key characteristics.

The transformation of specifications into engineering requirements formalizes the specifications into a structure that supports ideation. This target structure is chosen to support the design methodology used by the designers. For example, when Systematic Design (Pahl and Beitz, 1996) is used, designers represent requirements as function blocks characterized by flows of energy, material, or information. These function blocks are later associated with engineering principles to satisfy function. For an alternative methodology, in Axiomatic Design (Suh, 1990), designers create a list of Functional Requirements and use this list to construct a matrix that maps the Functional Requirements to Design Parameters.

The information flow model presented here suggests an organization for the engineering requirements, but does not mandate it. The engineering requirements can be categorized into two sets.

1. Function requirements related to the artifact's performance. They are characterized by verbs operating and transforming the artifact's performance characteristics. As in the NIST Design Repository Project (2000a), they can be modeled as input and output flows mediated by functions.
2. Form requirements directly related to the artifact's physical aspects. This set includes requirements on size, shape, and material. It also includes assembly requirements needed when the artifact is a component in an assembly or will be attached to fixtures in production processes.

The transformation from specifications to engineering requirements may require several iterations. When the mapping is difficult to achieve, the designers may have to revise the specifications and may even have to renegotiate the customer needs with the customer. Furthermore, the requirements will evolve as design progresses. The resulting updates and revisions are documented when deemed significant by designers.

3.4 Family of Solutions

Design information reaches the Family of Solutions state when designers identify a *general* solution of the design problem at hand. This general solution is an abstraction of a family of artifacts that may meet the engineering requirements and is similar to Gero's "Design Prototypes" (Gero, 1991). It describes members of this family of solutions as a collection of symbols that represents an artifact's characteristic properties. However, the solution family leaves this description incomplete by binding only a subset of the symbols to values; the remaining symbols are left unbound.

The family of solutions differs from the traditional view of conceptual design (which does not appear in the model). A conceptual design implies a particular level of abstraction. The family of solutions represents a design alternative at any level of abstraction. At a more abstract level it can be viewed as a design concept which evolves to a stage of more detailed design. The family of solutions contains symbols that are not yet bound to a particular instantiation of an artifact. Yet, the family of solutions allows the designer to formulate a behavior model.

The relationship between a solution family and its member artifacts is similar to the relation between a class and its instances in object-oriented programming. The class is a template that defines the properties common to its instances in terms of properties or instance variables. The instances customize their behavior by binding the instance variables to values. However, the difference between the formalism and the traditional class-instance view is that, during sequential refinement steps, what is an artifact (i.e., an instance) at the *i*th level of abstraction may become a family of solutions (i.e., a class) at the (*i*+1) th level of abstraction. For example, at one level of abstraction, designers may be considering the family of motors and may bind variables to select a brushless DC motor as an artifact that will satisfy requirements. Having made that choice, at the next level of abstraction, designers may be considering now the family of brushless DC motors and may bind additional variables as part of subsequent refinement steps.

In the case of design information, the family of solutions defines the properties of its member artifacts and describes the general characteristics of the form, function, and behavior of these members. The solution family has an *intended behavior* associated with it. As a matter of fact, the solution family is chosen with this intended behavior in mind. This intended behavior is expected to implement the artifact's needed function or a close approximation to it. The solution family also has a *behavior model* associated with it. This model is used to derive a member artifact's *observed behavior* when the member's description has been completed. Note that not all members of the solution family are guaranteed to satisfy the engineering requirements. Designers must complete a family member's description to verify that the member meets all the requirements.

As an example of a family of solutions, consider the following design problem.

1. A customer needs to transmit a certain motion between two points.
2. The specifications formalize the problem description and specify success metrics.
3. The engineering requirements describe in detail the motion that must be transmitted (e.g., shapes and durations of motion segments).
4. The designers identify a subset of four-bar linkages as a solution family by looking up similar trajectories in a catalog of four-bar linkages.
5. The intended behavior describes the desired motion in a manner that can be evaluated against a behavior model.
6. The behavior model consists of the laws of kinematics that characterize the motion of mechanisms.

The description of engineering requirements, identification of a family of solutions, and instantiation of this family of solutions into an artifact occur at a specific level of abstraction. In this case, the designers are trying to solve the delivery of motion problem and are not concerned with second-order effects such as force transmission and flexibility of the mechanism members. The designers have decided to use a four-bar linkage as a family of solutions. At the current level of abstraction, the mechanism remains under-specified as the designers have not specified the lengths of the mechanism members.

Designers use techniques such as brainstorming, intuition, catalog searches, or other structured and unstructured techniques to arrive at a solution family. It is also possible that designers explore multiple solution families simultaneously with each containing the types of information described above. The designers may need to revise or refine the needed engineering requirements when they can not identify an appropriate family of solutions. This revision may in its turn necessitate a revision of the engineering requirements.

3.5 Proposed Artifact

Design information reaches the Proposed Artifact state when the designers complete the description of the artifact at the current level of abstraction. The designers do so by binding values to the unbound symbols in the description of the solution family, thereby selecting a specific member from the family of solutions.

The designers may be able to use the completed description for a quick evaluation of the proposed solution or for a more detailed analysis using the behavior model specified by the family of solutions. For example, in the case of the delivery of motion problem, the designers specify the lengths of the links in the four-bar linkage. This allows them to simulate the motion delivered by the selected mechanism.

As in previous cases, the designers may have to select a new family of solutions if they are unable to instantiate the current solution family into a proposed artifact that meets the engineering requirements. This backtracking step may in its turn trigger additional backtracking steps such as reconsidering the choice of engineering requirements.

3.6 Observed Behavior

The design information reaches this state when designers derive the artifact's behavior from its description and the behavior model specified by the solution family. Designers may have several options to derive this observed behavior. They can use a mental simulation, build a physical prototype, or use a general purpose or domain-specific simulation engine, among others.

For the delivery of motion example, designers use a mechanism simulation tool to determine the path traced by specific points on the mechanism. At this level of abstraction, a simulation may only take into account the kinematic aspects of the mechanism's behavior and may not account for other aspects such as force transmission.

3.7 Evaluated Behavior

Design information moves into the Evaluated Behavior state after the designers evaluate the artifact's behavior. They do so by comparing the artifact's intended and observed behaviors and classifying any discrepancies between the two as a variation in *intended* behavior or an *unintended* behavior. This classification is based on two criteria: (1) how closely the artifact's observed behavior matches its intended one, and (2) whether the discrepancy reflects a phenomenon of a different nature than the one anticipated by the designers. The evaluation effectively answers the question, "does the artifact do what it is supposed to do?" and can lead to one of the following three courses of action:

1. The behavior discrepancy is within acceptable bounds and is classified as a variation in intended behavior. Designers decide that the discrepancy does not warrant a further revision of the proposed artifact at the current level of abstraction. The discrepancy itself is noted and may lead to the development of tolerances at the appropriate level of abstraction.
2. The behavior discrepancy is outside acceptable bounds. However, designers assess that they are still dealing with the same phenomenon and classify the discrepancy as a variation. In this case, designers have a choice. They may decide that the current solution family remains promising and that the proposed artifact can be improved. The designers then modify the proposed artifact by binding the solution family's unbound properties to a new set of values. Designers may use an optimization approach to determine the changes needed. Alternatively, designers may decide that a further investigation of the solution family is unwarranted and select a new solution family. This backtracking may itself result in revising the engineering requirements before the artifact's design is refined further.
3. The discrepancy is substantial enough that designers assess that they are dealing with a phenomenon whose nature they did not anticipate; they need to use different terms when describing the intended and observed behaviors. As such, the behavior discrepancy is not simply out of bounds, but rather out of set, and is classified as an unintended behavior. It may be desirable or undesirable. This unintended behavior may lead designers to revise the description of the intended behavior in the current family of solutions, select another family of solutions, or backtrack further to revise the engineering requirements. Alternatively, the unintended behavior may be carried forward to the evaluated requirements for analysis.

For the delivery of motion example, the designers can choose to optimize the lengths of the links to match more closely the desired motion. Alternatively, they may decide that the current four-bar linkage is not promising and start with a new set of link lengths.

As an example of how an unintended behavior can appear in a design, consider the design of a circuit that is part of an electro-mechanical device. Designers decide to ignore heat dissipation in the early phases of the design. As a result, their intended behavior does not include a heat dissipation aspect. However, heat turns out to be a factor when they use a diode in their circuit. The behavior model for electric circuits has a heat aspect irrespective of whether a diode is used. Designers typically ignore this aspect when circuit components generate little heat, but can no longer do so because diodes generate considerable heat. When analyzing the behavior of their circuit, they must use heat-related terms that were not initially used to describe the intended behavior. In this case, designers can develop a new circuit without a diode, or alternatively, they can retain the diode and include heat-dissipating elements in the circuit thereby transforming the unintended dissipation into an undesirable, but anticipated behavior.

3.8 Evaluated Requirements

The design information reaches the Evaluated Requirements state after the designers evaluate whether the artifact satisfies the engineering requirements and meets the specifications at the current level of abstraction. This evaluation takes into account any of the unintended behaviors of the proposed artifact and can lead to one of several courses of action.

1. The proposed artifact meets all requirements and specifications as derived from customer needs. As such, the designers decide that the design is complete at the current level of abstraction.
2. The proposed artifact satisfies the requirements at the current level of abstraction, but not all the specifications. The designers refine the description of requirements to a more detailed level of abstraction and will then refine the design to meet the new set of requirements.
3. The proposed artifact does not satisfy the requirements at the current level of abstraction. The designers must then iterate to consider alternate artifacts, alternate families of solutions, or reconsider the customer needs, specifications or engineering requirements.

It is also likely that multiple alternatives would be compared at this point. There would be an evaluation of the degree of satisfaction of the requirements among the alternatives. In performing this evaluation, it is important that each of the alternatives be compared at the same level of abstraction. Otherwise, the determination can be invalid.

Revisiting the transmission of motion problem, designers have determined the link lengths in a mechanism so it traces the proper trajectory. At the next level of abstraction, a more detailed behavior model that considers the transmission of forces in addition to the transmission of motion may be used.

4 Transformations Between Levels of Abstraction

Designers transform design information as they navigate between levels of abstraction (Hoover and Rinderle, 1994). Design alternatives are explored through a series of interactions through the various levels of abstraction. The alternatives are refined through an interplay of solution families and instantiated artifacts in an effort to satisfy the engineering requirements. As described in section 3, a proposed design alternative represented as a family of solutions is associated with the following information: an intended behavior, a behavior model, and a description containing bound and unbound symbols. The exploration of the design alternative involves the binding of the symbols in an effort to evaluate the intended behavior with the observed behavior. Three types of interactions between levels of abstraction can be identified as illustrated in Figure 4.

1. *Refinement*—the designers solve a design problem by identifying an artifact that meets the requirements at a given level of abstraction. They then *refine* the problem's description by incorporating additional detail, thereby leading to a new abstraction level, before returning to the solution of this problem again. This is the interaction between abstraction levels $Level_i$ and $Level_{i+1}$. The description at $Level_i$ uses the symbols $S_1 \ldots S_l$. By contrast, the description at $Level_{i+1}$ includes additional detail represented by the symbols $S_l \ldots S_{l+m}$.
2. *Reformulation*—the designers decide that a problem is too difficult to solve at the given level of abstraction. As such, they *reformulate* the problem description, making it more tractable, and attempt to solve the reformulated alternate problem before returning to the original one. In the example of Figure 4, the designers reformulate the problem in $Level_i$ described by symbols $S_l \ldots S_l$ to use symbols $S'_1 \ldots S'_p$ in $Level_{i,1}$.
3. *Reconciliation*—having previously reformulated a design problem and solved the new problem including any refinements applied, the designers *reconcile* the solution of the alternate problem to the requirements of the original problem. This is the interaction between levels of abstraction $Level_{i,N}$ and $Level_i$ in Figure 3.

The levels of abstraction paradigm provides a *formalism* in the model that is needed to develop computational tools for manipulating design information. This formalism is achieved without placing undue burden on designers as the designers always control the *granularity* of the model. They choose a level of abstraction by selecting the symbols used to describe an artifact. They can apply fine-grained transformations, such as changing the value of one symbol, or coarse-grained transformations such as reformulating the design problem. Because levels of abstraction are not pre-defined, there is no constraint governing how many unbound variables must be bound before refining to the next level of abstraction. The intent is not to constrain which design activities take place, nor their sequence. The intent is to describe the types of information that are formulated and lay a formalism for collecting that information.

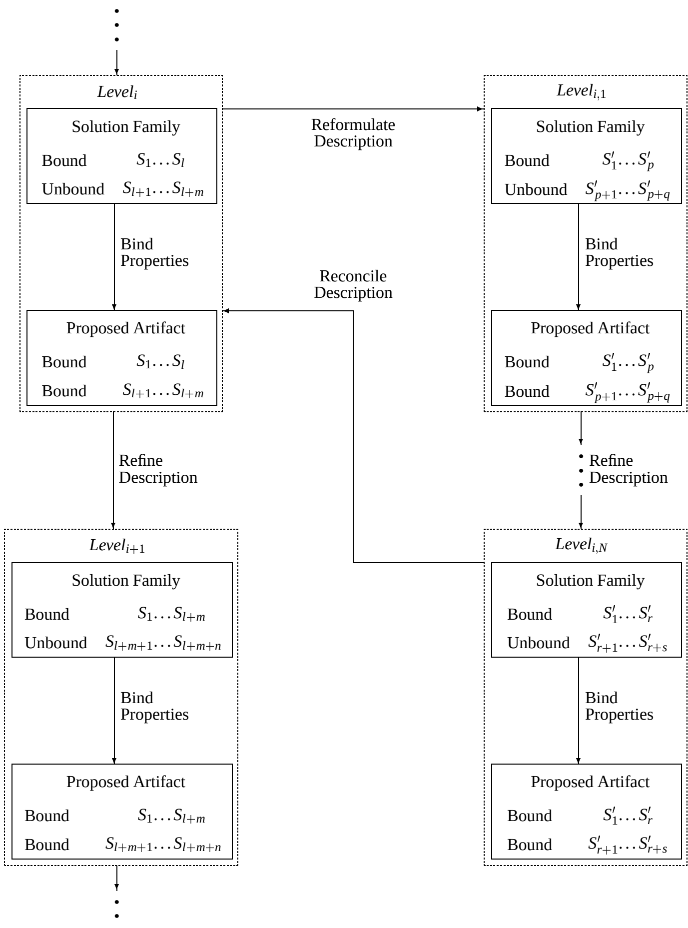

Figure 4: Interaction Between Levels of Abstraction

5 EXAMPLE – Design of a Planetary Gear Transmission

This example uses the model to map the flow of design information for the transmission in the Black & Decker[1] VP480 cordless drill. The transmission is one subassembly of several that make up the drill and is located between the motor and the clutch head. A schematic of the drill is shown in Figure 5. The example begins with the recognition of the need for some transmission to provide an angular velocity reduction/torque increase between the motor shaft and the chuck that holds the drill bit. For this example, each of the information states are acted upon in the order described in Figure 3 for two complete levels of refinement to the point where the transmission structure is formulated. This is not meant to indicate a prescriptive process model, but rather to illustrate the types of information formulated at each state for different levels of abstraction. The design information is presented in the form of an outline in the interest of clarity.

Regarding units, the Black and Decker Cordless Drill is marketed with ratings in English units. In the interest of consistency, the example here is presented with English units. Metric equivalents are included for reference only. However, these are conversions performed after completion of the design and would not likely have been part of the original information flow. The use of English units in this example is significant because gears specified in an English standard are different than those in metric standard. They are not interchangeable.

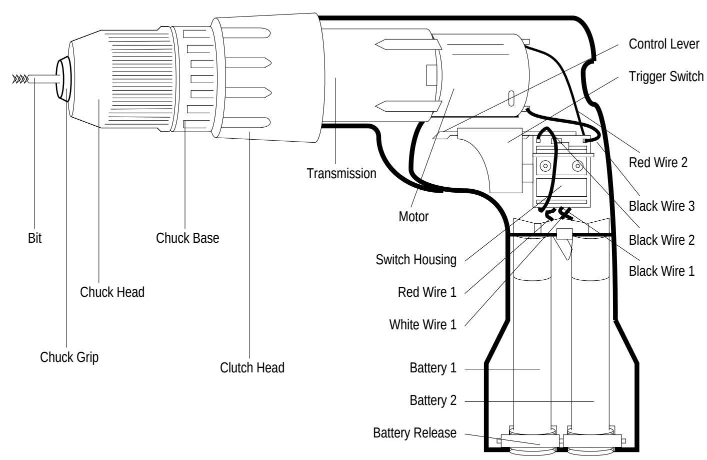

Figure 5: Power Drill Schematic

5.1 First Pass Through the Information States

The design begins with the recognition of the need for speed reduction and torque increase from the motor to the chuck that holds the drill bit. The designers must establish the specifications and perform a preliminary feasibility search on possible solutions.

5.1.1 Customer Needs

The customer needs for the transmission are a subset of the overall customer needs for the drill. At the initial level of abstraction the designer may begin by listing those needs that appear to have relevance for the transmission design.

1. Sufficient torque to (a) drill a hole, (b) drive a screw (forward and reverse)
2. Adequate speed for all operations
3. Variable speed for different operations.
4. Manageable weight.
5. Ergonomics: easy to grip, activate, etc.
6. Balanced handling.
7. Portable: cordless.
8. Limit torque/don't break the bit.

[1] Use of any commercial product or company names in this paper are intended to provide readers with information regarding the implementation of the research described and does not imply recommendation or endorsement by the National Institute of Standards and Technology.

9. Manageable size.
10. Quiet.
11. Cost competitive for home use market.

5.1.2 Specifications

The specifications are formulated by considering the customer needs. The specifications are characterized by measurable parameters with target values and constraints. Depending on the level of abstraction, the target values may not be known and may require further investigation to establish meaningful values. It is often useful to describe the rationale for the requirement and its target value. The assumption is made that a standard motor has been pre-selected and that its use is strongly preferred.

In formulating the specifications, an effort was made to maintain a one-to-one correlation with customer needs. However, this correlation is often not that simple and requires more sophisticated techniques. In any case, the flow of design information model will contain the information relating to the technique used by the designers.

1. Supply torque of "target value" for each desired operation.
   (a) Supporting factors:
      i. Material to drill i.e., wood (hard to soft), metals, etc.
      ii. Bit size range
         A. Wood up to ½ inch (12.7 mm).
         B. Steel up to 3/8 inch (10 mm).
   (b) Torque and speed values for drilling a hole.
   (c) Torque and speed values for driving a screw.
2. Cordless: battery power.
   (a) Influences motor size that has been previously standardized.
   (b) Influences input angular velocity: motor produces 9600 rpm in low and 19200 rpm in high.
   (c) Influences input torque: motor can produce 1.9 in-lb. (0.21 N-m).
   (d) Two VersaPak batteries supply 7.2 volts.
3. Output requires lower rpm, higher torque.
   (a) Values of rpm range for drilling, screwing, self-tapping screwing.
   (b) Two speeds 300/600 rpm set by low/high switch on motor.
   (c) Need reverse.
   (d) Torque requirement of 60 in-lb. (6.78N-m) established by competitive benchmarking.
4. Weight limit of 6 ounces (140 g)—minimize. Transmission is a subsystem of drill which governs the weight.
5 Balanced System.
   (a) Concentric center of gravity—longitudinal.
   (b) Concentric center of gravity—radical.
6. Stress limits on gear box components.
   (a) Torque values influence components.
   (b) Forces influence component selection.
7. Size Restriction: minimize values.
   (a) Volume limits.
   (b) Length.
   (c) Width.
   (d) Shape.
8. Cost: total drill sales price of $45; transmission must be a small fraction of that.

5.1.3 Engineering Requirements

The engineering requirements formalize the specifications to a structure that facilitates ideation. The technique used here is to formulate function requirements and form requirements. Function requirements are characterized by input/output flows operated on by a transformation as described by Szykman et al. (2000a). Form requirements dictate the form of the artifact.

Many different process models suggest their own formalism to support ideation. The information flow model proposed here does not seek to dictate the ideation formalism. However, it is recognized that most process models do establish a formalism that should be captured in the design information flow.

For this example at this instance, it is not necessary to formalize all of the specifications to engineering requirements. The primary function to be explored is the need for angular velocity reduction. The form requirements are also limited to the minimum set needed for exploration at this pass.

Function Requirements:
1. Flow—*Convert*.
2. Input—*Rotation.*
   (a) speed =9600 rpm (low) and 19200 rpm (high).
   (b) torque =1.9 in-lb (0.21 N-m) (max value).
3. Output—*Rotation*.
   (a) speed =300 rpm (low) and 600 rpm (high).
   (b) torque = 60 in-lb. (6.78 N-m) (max value).

Form Requirements:
1. Concentric shafts, mate with output shaft of motor, mate with input shaft of chuck, concentric center of gravity (symmetry).
2. Size restriction:
   (a) Length < 2.0 inches (50.8 mm).
   (b) Width < 1.5 inches (38.1 mm).
   (c) Height < 1.5 inches (38.1 mm).

5.1.4 Family of Solutions

The designers must explore possible solutions. In this case, the four alternatives shown below were generated. These alternatives represent broad concepts that suggest a form at a coarse level of abstraction.

1. Gear box.
2. Belt/pulley.
3. Direct Couplings.
4. Variable Speed Motor.

This example illustrates the exploration of the gearbox family of solutions. The other families of solutions would be explored in a similar manner. The refinement through the levels of abstraction is labeled with subheading numbers to organize the levels of abstraction. These levels correspond to the binding described in Figure 3.

Level 1—Gearbox:

In exploring the gearbox solution family, the designer(s) decide to first determine the feasibility through a catalog search of available gearboxes. The intended behavior is refined to indicate the desired gear reduction of 32:1, which is in the syntax of gearbox specifications. The desire for colinear shafts and the space restrictions are included. At this exploratory stage of conceptual design, the behavior model may be replaced by – or approximated by – a catalog exploration. The intent is to employ a technique to bind symbols in the exploration of the design alternatives. The description of the behavior model includes bound and unbound symbols. The bound symbols are those that have been instantiated at this level of abstraction.

1. Description:
   (a) Bound
      i. Input speed = 9600/1920 rpm.
      ii. Input torque = 1.9 in-lb. (0.21 N-m) (max).
   (b) Unbound
      i. Output speed.
      ii. Output torque.
      iii. Gear ratio.
      iv. Form.
2. Intended Behavior:
   (a) Speed reduction/torque increase from gear ratio of nearly 32:1.
   (b) Colinear shafts.
   (c) Size restriction.
      i. Length < 2.0 inches (50.8 mm).
      ii. Width < 1.5 inches (38.1 mm).
      iii. Height < 1.5 inches (38.1 mm).
3. Behavior Model:
   At this exploratory level of conceptual design, the behavior model may be replaced by – or approximated by – a catalog exploration.

5.1.5 Proposed Artifact— (Level 1,1):

The catalog search results in a large number of possible artifacts to satisfy the engineering requirements. It is learned that gear reductions are available from 3:1 to 3000:1. Gearboxes with Colinear shafts tend to be cylindrical with size ranges of .96 to 1.75 inches (25 to 43 mm) in diameter and 1.4 to 2.2 inches (35.5 to 56 mm) in length. Other information is garnered from the catalog search that can be categorized as unintended behavior because they were not listed as part of the intended behavior set. These characteristics will need to be explored for significance to this design problem. The catalog also listed a cost of $350 for a precision gearbox. The cost will need to be considered with respect to the design specifications.

Notice that the bound and unbound symbols do not change for this level of refinement. The catalog search involved an exploration, but the design artifact was not bound.

1. Observed Behavior:
   (a) Gear reductions available 3:1 to 3000:1.
   (b) Colinear shafts.
   (c) Sizes available: 0.96 – 1.7 in. diameter (25 – 43 mm), 1.4 – 2.2 in. length (33.5 – 56 mm). Note that the form is in the shape of a cylinder.
2. Unintended Behavior: List of specification parameters from the catalog.
   (a) Gear ratio, number of gear clusters—affects ratio not cost.
   (b) Diameter.
   (c) Colinear shafts.
   (d) Shaft size (input and output).
   (e) Direction of rotation.
   (f) Max rated output torque (starting and operating).
   (g) Backlash.
   (h) Weight.
   (i) Shaft end play (radial and longitudinal).
   (j) Moment of inertia of the input shaft.
   (k) Lubrication.
   (l) Gear tolerances.
3. Description—Bound and Unbound symbols do not change.

5.1.6 Evaluated Behavior

The evaluated behavior involves a comparison of the observed behavior with the intended behavior. The resolution is that it is acceptable at this level of abstraction. The catalog search showed that a gearbox can meet the intended behavior.

5.1.7 Evaluated Requirements

The evaluated requirements involves comparing the results from the behavior model with the engineering requirements. At this level of refinement, the gearbox alternative shows promise but there is a need to consider the cost of $350.

5.2 Second Pass Through the Information States

The decision has been made to explore the gear box family of solutions in greater detail. New information has been garnered from the first level of abstraction. The second pass through the design information flow will require refinement at each of the stages with consideration given to what was just learned.

5.2.1 Customer Needs

The customer needs do not change. The cost of $350 discovered from the catalog exploration forces the designer to recognize a total retail sales cost for the drill of $45. The transmission must be a small fraction of this.

5.2.2 Specifications

The engineering requirements are revised with consideration of the characteristics included in the catalog. These new characteristics are rated on a level of importance to the design, as high, medium, or low.

1. Production cost for transmission with (100,000 lot size limited to $5)—high.
2. Gear ratio, number of gear clusters—affects ratio not cost—high.
3. Diameter: Meet size restrictions on the housing—high.
4. Colinear shaft—high.
5. Shaft size (input and output)—low.
   (a) Input must mate with motor output shaft.
   (b) Output must mate with chuck.
6. Direction of rotation—low.
   (a) Must coordinate with motor polarity.
7. Max rated output torque (starting and operating)—high.
   (a) Established previously.
8. Backlash—low.
   (a) Not important for consumer market.
9. Weight—high.
   (a) Component weight is part of overall weight. Limit weight to 8 oz.
10. Shaft end play (radial and longitudinal)—low.
11. Moment of inertia of the input shaft—low.
    (a) Motor must be able to overcome moment of inertia for starting.
12. Lubrication—low.
    (a) Do not expect the user to maintain. Self lubricating or tightly sealed.
13. Gear tolerances—low.
    (a) Play not likely as important for this application.

5.2.3 Engineering Requirements

The engineering requirements remain mostly unchanged. In the first pass the size restriction for the form requirement was stated for a volume in the form of a box. The catalog exploration indicated that gearboxes with colinear shafts are in the shape of a cylinder. This change seems appropriate and the form requirements are updated.

1. Function Requirements—unchanged.
2. Form Requirements.
   (a) Alter size restriction to consider cylindrical shape.
      i. Length < 2.0 inches (50.8 mm).
      ii. Diameter < 1.5 inches (38.1 mm)
   (b) Other Form Requirements stay the same.

5.2.4 Family of Solutions—Gearbox Family

After the customer needs, specifications and engineering requirements are refined with insight from the first pass, the designer(s) continue the exploration of the gearbox family of solutions. The designers decide to explore their own design of a planetary gearbox.

As described in section 4, design alternatives are explored through a series of interactions through various levels of abstraction. There is an iterative interplay between the family of solutions and the proposed artifacts to satisfy the intended behavior.

The planetary gearbox family of solutions establishes the abstract form shown in Figure 6. The abstract form includes the various components such as the sun, planets and ring gears that are connected by an arm. The sizes of the gears will establish the gear reduction behavior of the gearbox.

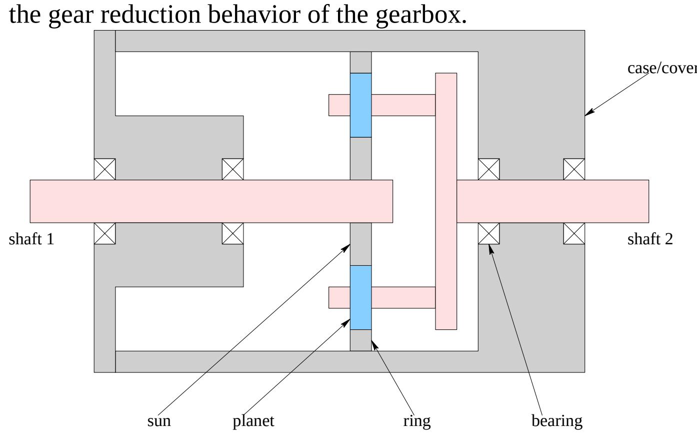

Figure 6: Basic Planetary Gear System

The formulation of the behavior model requires the designer to incorporate information on gear design. The same general behavior model from the solution family can be called upon during the instantiation of artifacts for exploring the design space. The information about the planetary gearbox is listed as level 2 as an indication of a refinement in the description from the first pass where the catalog exploration was performed. The bound symbols include the input speed and torque from the previous level of abstraction. The list of unbound symbols increases to include unknown information about planetary gear system elements.

5.2.4.1 Level 2—Planetary Gearbox

1. Description
   (a) Bound – same as level 1.
   (b) Unbound.
      i. Sun Gear Size.
      ii. Planetary Gear(s) Size.
      iii. Ring Gear Size.
2. Intended Behavior
   (a) Gear ratio of 32:1.
   (b) Output torque of 60 in-lb. (6.78 N-m) from a motor torque of 1.9 in-lb. (0.21 N-m).
   (c) Diameter less than 1.5 inches (38.1 mm).
3. Behavior Model

   The behavior model then contains equations that relate the components and allow the exploration of the intended behavior. The derivation for the angular velocity reduction of planetary gear trains can be found in a kinematics text such as Mabie and Reinholtz (1987). The information model would contain the derivation, but it is not reproduced here in the interest of space. The general equation for a planetary gear train of this configuration is then expressed as:

$$\frac{\omega_{in}}{\omega_{out}} = 1 + \frac{N_{ring}}{N_{sun}} = 1 + \frac{D_{sun}}{D_{ring}} \qquad (1)$$

Equation (1) is the essential equation for exploring the intended behavior. It relates the input and output gear ratio. With this information the designers can explore individual possible solutions by binding symbols with actual values.

5.2.5 Proposed Artifact

The family of solutions model now contains bound and unbound symbols and a behavior model. With this information the designers can explore alternative proposed artifacts by binding symbols.

5.2.5.1 Level 2,1—Single Planetary Gear Train

The first alternative involves the selection of a single planetary gear train. The parameters describing the gear sizes are obtained from a design repository. Counting the number of teeth on each of the gears results in the following:

1. Description
   The Bounded Description is summarized in the table below:

Sun	N_1	54
Planets	N_3	27
Ring	N_4	108

2. Behavior Model
   Equation 1 from the family of solutions' behavior model

$$\frac{\omega_{in}}{\omega_{out}} = 1 + \frac{N_{ring}}{N_{sun}} \qquad (2)$$

$$\frac{\omega_1}{\omega_2} = 1 + \frac{N_4}{N_1} = 1 + \frac{108}{54} = 3 \qquad (3)$$

This means that the output shaft will turn once for every three revolutions of the input shaft reducing the angular velocity by a factor of three and conversely increasing the torque by a factor of three.

5.2.5.1.1 Observed Behavior

The observed behavior for the single planetary gear train resulted in a gear reduction of 3:1.

5.2.5.1.2 Evaluated Behavior

The observed behavior is then compared to the intended behavior inherited from the family of solutions. The comparison results in the designers formulating some sort of resolution. The evaluated behavior is summarized in the table below:

Behavior	**Reduction**
Intended	32:1
Observed	3:1

This instantiation of a single gear planetary gear train is unacceptable. Insight garnered from the behavior model indicates that the ring gear must be significantly larger than the sun gear to produce the desired reduction. This insight will be used for the next instantiation.

5.2.5.2 Level 2,2—Single Planetary Gear Train

The second instantiation for a proposed artifact involves a different approach. In this case the designers establish the size of the sun gear and then determine the size of the ring gear that will provide a gear ratio of 32:1. The designer(s) are able to call upon the behavior model from the planetary gear box family of solutions. This reformulation of the description assigns a new sub-level for the proposed artifact.

1. Description
   To obtain a bound description, establish a desired gear ratio of 32:1 and set the number of teeth on the sun to 9.

Planets	N_3	*unbound*
Sun	N_1	*9 bound*
Ring	N_4	*unbound*

2. Behavior Model
   For a desired gear ratio of 32:1, Equation 1 from the family of solutions' behavior model indicates that:

$$\frac{\omega_{in}}{\omega_{out}} = 1 + \frac{N_{ring}}{N_{sun}} = 1 + \frac{D_{sun}}{D_{ring}} \qquad (4)$$

$$\frac{\omega_1}{\omega_2} = 32 = 1 + \frac{N_4}{9} \qquad (5)$$

Solving for N_4 results in:

$$N_4 = 279 \qquad (6)$$

Also,

$$\frac{D_4}{D_1} = 31 \qquad (7)$$

$$D_1 = \frac{1.5in}{31} = 0.048in = 1.3mm \qquad (8)$$

5.2.5.2.1 Observed Behavior

The behavior model was used to force the desired gear reduction of 32:1. The governing equations then provided information on the gear sizes that satisfy that aim. The ring gear would need to have 279 teeth and be 31 times larger in diameter than the sun gear. For a ring diameter of 1.5 inches

(38 mm), the sun gear diameter would be only 0.05 inches (1.3 mm)!

5.2.5.2.2 Evaluated Behavior

The evaluated behavior is summarized in the table below:

Behavior		
	Intended	32:1
	Observed	32:1
	Unexpected	Sun gear diameter of only 0.05 in. (1.3mm)
Resolution		Acceptable gear ratio

The behavior evaluation indicates that this planetary gearbox will provide the desired gear ratio which is the intended behavior under investigation. However, there is an unexpected behavior in that the sun gear diameter must be only 0.05 inches (1.3 mm) in order to satisfy the behavior requirement of a diameter less than 1.5 inches (38.1 mm).

5.2.5.2.3 Evaluated Requirements

The proposed artifact satisfies the intended behavior established for the family of solutions. It is then necessary to consider the engineering requirements. In addition to the function requirement of the desired gear reduction, there is a form requirement that considers the size of the system. In order to meet the form requirement of a gear box diameter less than 1.5 inches (38 mm), the sun gear would need to be only 0.05 inches (1.3 mm) in diameter. This is too small and unacceptable. The resolution from the requirements evaluation is then to reconcile the description and explore another proposed artifact.

5.2.5.3 Level 2,3—Two Linked Planetary Clusters

For the next level of refinement of the proposed artifacts, the designers decide to explore the possibility of nesting two planetary gear trains. Each gear train cluster will provide part of the gear reduction. Because the gear reduction for each cluster is less, the relative size of the sun gear to the ring gear will not have to be as great. The intent is that this will allow a sun gear of acceptable size.

Each gear cluster is of the same form described by the planetary gear box family of solutions. The behavior model for this alternative can draw from the general planetary gearbox behavior model. However, the new description includes a formulation that connects the two planetary gear trains.

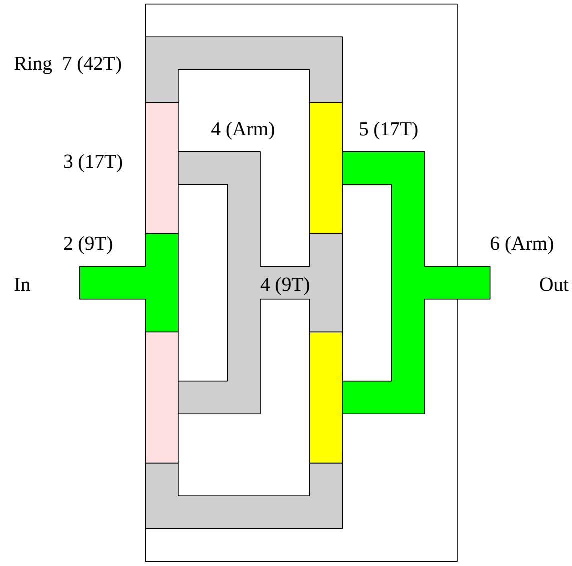

Figure 7: Two Linked Planetary Clusters

1. Description
   The new alternative consists of two planetary gear trains and is illustrated in Figure 7. The input shaft is attached to the first planetary gear train cluster at gear 2. The output shaft is attached to the second planetary gear train cluster at arm 6. The internal ring gear 7 is common to both gear trains and the arm 4 of the first cluster is attached to the sun 4 of the second cluster. The bound description includes the numbers of teeth for each gear indicated in the figure.
2. Behavior Model
   Use the following numbering system for the two gear train cluster components:

	Cluster 1	No. of Teeth	Cluster 2	No. of Teeth
Sun	2	9	4	9
Arm	4	-	6	-
Planet(s)	3	17	5	17
Ring	7	42	7	42

Equation 1 from the family of solutions' behavior model for the two planetary gear train clusters results in:

$$\frac{\omega_{in}}{\omega_{out}} = 1 + \frac{N_{ring}}{N_{sun}} = 1 + \frac{D_{ring}}{D_{sun}} \qquad (9)$$

Cluster 1:

$$\frac{\omega_2}{\omega_4} = +\frac{N_7}{N_2} = 1 + \frac{42}{9} = 5.67 \qquad (10)$$

Cluster 2:

$$\frac{\omega_4}{\omega_6} = +\frac{N_7}{N_4} = 1 + \frac{42}{9} = 5.67 \qquad (11)$$

Linking equation 10 with equation 11 through the common w_4 results in:

$$\frac{\omega_{in}}{\omega_{pit}} = \frac{\omega_2}{\omega_6} = 32.1 \qquad (12)$$

5.2.5.3.1 Observed Behavior

The proposed artifact demonstrates a gear reduction of 32.1:1.

5.2.5.3.2 Evaluated Behavior:

The behavior comparison is summarized in the table below:

Behavior	**Gear Reduction**
Intended	32:1
Observed	32.1:1
Variation	0.1
Resolution	Acceptable

The observed gear reduction of 32.1:1 indicates a variation of 0.1 from the intended gear reduction of 32:1. This variation will likely not have a significant effect on the performance of the gear train. The designer(s) decide that this variation is acceptable for this level of abstraction. The proposed planetary gearbox with two planetary clusters meets the intended behavior.

5.2.5.3.3 Evaluated Requirements:

The intended behavior was described as providing a gear reduction of 32:1. This value was derived from the requirement of a desired output speed from a known input speed. It is actually the output speed and the accompanying output torque that is of most interest for the design. Therefore, it is necessary to explore how the proposed gearbox meets those requirements.

1. Motor speed of 9600 rpm (low) and 19200 rpm (high) results in output speed of 299 rpm (low) and 598 rpm (high).
2. Input torque from motor of 1.9 in-lb. (0.21 N-m) results in output torque of 61 in-lb. (6.89 N-m).
3. Ring diameter of 1.5 inches (38.1 mm) will result in a sun gear diameter of 1.5/4.67 = 0.32 inches (8.1 mm).

The designers decide that each of these results is acceptable at this level of abstraction. The double-cluster planetary gearbox will satisfy the engineering requirements within an acceptable variation.

5.3 Continued Design of the Gear Box

The design at this level of abstraction now contains an adequate description to satisfy the function requirement *convert input rotation to output rotation.* Resolution of other specifications will dictate the formulation of more refined engineering requirements. For example, the issue of input and output torques will guide the determination of the number of planets as well as the selection of diametral pitches and materials for the gears.

Refinement of the artifact will continue through the iterative process to new levels of abstraction. The design context that captures the design information will be updated. The final artifact will contain a complete description of bound symbols that characterize the form, function and behavior.

6 CONCLUSIONS

This paper described a design information flow model. The initial motivation for this work was the development of knowledge representations to support an agent-based architecture for the OpenADE project at the National Institute of Standards and Technology. An assessment of the content-related needs for such a representation identified the requirements for the support of the architecture. An analysis of the transformations that the product representation undergoes during product development resulted in the specification of a generic model of information flow that is not tied to any single process. It therefore lends itself to an implementation that can support a variety of product development processes and activities. The model supports semantics-based translation and exchange of data to support the flow of information from one product development activity to another.

It is important to recognize that design information evolves through transformations in levels of abstraction. Process models prescribe a suggested approach for developing designs through these levels of abstraction. The flow of information model described here does not prescribe the design process. Because levels of abstraction are not pre-defined, there is no constraint governing how many unbound variables must be bound before refining to the next level of abstraction. For example, it is acceptable for a designer to immediately select a planetary gearbox as a solution family. The formalism does not require a designer to first pick geared transmissions as alternative and then select a planetary gearbox rather than a standard gearbox. The design activities or phases are not constrained, nor their sequence.

Different representations of design information in various product development processes share semantics because these development processes are based on the same information flow and activity classes. In other words, the form of the representation may vary but there will be much in common at the level of abstraction of what is represented. The design information flow model establishes a foundation for developing representations to capture, store and retrieve design information. An extension of this work resulted in a detailed description of an object-level representation for the capture, storage, and retrieval of design information presented in Szykman et al. (2000b). That work describes the concept of an "engineering context", a corpus of information that evolves with time throughout the product development process. The engineering context may exist in a unified database, or in a "virtual" data base consisting of various bodies of engineering information that reside with particular applications. The intent of these efforts is to develop a foundation for interoperability in the next generation of product development systems.

REFERENCES

S. Angster, K. Lyons, P. Hart, and S. Jayaram (1998). "Interoperability of Assembly Analysis Applications Through the Use of the Open Assembly Design Environment," *Proceedings of DETC98, 1998 ASME Design Engineering Technical Conference*, Atlanta, GA, September 13-16.

Gero, J.S. (1991). "Design Prototypes: A Knowledge Representation Schema for Design," *AI Magazine,* 11(4):26-36.

Hauser, J.R. and D. Clausing (1988). "The House of Quality," *Harvard Business Review*, May-June, pp. 63-73.

Hoover S. and J.R. Rinderle (1994). "Abstractions, Design Views and Focusing," *Proceedings of the 6th International Conference on Design, Theory, and Methodology,* volume DE-68, pages 115-130, Minneapolis, MN, September.

Kalay, Y.E. (1999). "Performance-based Design," *Automation in Construction,* 8(4):395-409.

Lyons, K., S. Shooter, W. Keirouz, P. Hart (1999). "The Open Assembly Design Environment: An Architecture for Design Agent Interoperability," *Proceedings of the 1999 ASME Design Technical Conferences, number DETC99/DFM-8945*, Las Vegas, Nevada, September 12-15.

Mabie, H.H. and C.F. Reinholtz (1987). *Mechanisms and Dynamics of Machinery, 4th Edition*. John Wiley and Sons, New York, NY.

Pahl, G. W. Beitz (1996). *Engineering Design: a Systematic Approach.* Spring-Verlag, 2nd edition, London.

Srinivasan, V. (1999). "On Interpreting Key Characteristics," *Proceedings of the 1999 Design Engineering Technical Conferences, number DETC99/DAC-8701*, Las Vegas, Nevada, September 12-15.

Suh, N.P. (1990). *The Principles of Design.* Oxford series on advanced manufacturing. Oxford University Press, New York.

Szykman, S., J.W. Racz, C. Bochenek, and R.D. Sriram. (2000a). "A Web-based System for Design Artifact Modeling," *Design Studies*, Vol. 21, No. 2 pp. 145-165.

Szykman, S., S. Fenves, S. Shooter and W. Keirouz (2000b). "A Foundation for Interoperability in Next-Generation Product Development Systems," *Proceedings of the 2000 ASME Design Technical Conferences*, number DETC2000/CIE-14622, Baltimore, Maryland, September 10-13..

Whitney, D.E. (1996). "The Potential for Assembly Modeling In Product Development and Manufacturing," WWW http://imvp.mit.edu/imvpfree/Whitney/assembly.pdf.