1 Introduction

Electromechanical design often requires designers to create an engineering solution that spans multiple engineering domains. Often, this forces engineers to synergistically consider design elements from numerous engineering fields such as mechanical, electrical, and computer. For successful designs to emerge, the integration of engineering domains must be considered through all phases of product design, both conceptual and physical. Design elements must work together flawlessly, requiring adequate communication, monitoring, processing, and decision-making abilities. The modeling of information, i.e., signal flow, is vital for accurate and complete design models that support the design process.

One approach to developing system models to support multidisciplinary engineering projects and products is functional modeling. A functional model is a description of a product or process in terms of the elementary functions that are required to achieve its overall purpose [1]. A function is an operation by a device or artifact on a flow of material, energy, or signal passing through the device or artifact. While the concept of defining product function as a step in the early phases of product design has existed in the literature since at least the mid-19th century [2–4], it still lacks the formalization of other accepted engineering design analysis techniques. At the same time, however, functional modeling offers significant capabilities for product design in several areas [1].

- Systematic methods to model product functionality. Functionality is arguably one of the fundamental constructs of product design. The ability to model a product's functionality independent of and prior to form is critical for products that meet customer demands.
- Creativity in concept generation. The ability to decompose a complicated task into simpler pieces is a critical step in the synthesis process [5]. Identifying the key functionality of a product to be designed significantly increases the ability of designers to break problems down and find innovative solutions.
- Archival and reuse of design knowledge. Products are transient, and the design process and decisions that led to the product are even more fleeting. A functional model provides a record of the abstract requirements of the embodied product. Recording the relationship between function and product components allows future designs to leverage the past knowledge of previous designs and designers.
- Product architecture definition. The functionality of a product, arranged graphically as a functional model, can guide the architecture of a product. Groups of functions operating on common flows (the connections between the functions) lead to modules or platforms on which a family of products may be designed.

Accepting the premise that functional modeling is an important step in a product design process, the general approach to functional modeling provides the graphical tools required to develop a complete model of a product during conceptual design. When the Functional Basis is used for terminology, there is consistency of nomenclature that allows for a universal understanding of function and flow [6]. The Functional Basis, however, does not explicitly address consistency of model structure. Structure must be provided through a well-defined grammar consisting of morphology and syntax. Without a properly defined grammar, difficulty arises in the development of uniform and synergistic models that completely and accurately represent all engineering domains. Material and energy flows have an established grammar built from trial and error conventions [7,8]. Signal flows, however, tend to be much more ambiguous. An anecdotal study by the authors of 20 electromechanical products that were dissected and modeled functionally with the Functional Basis revealed vastly different techniques on how signal flows were modeled. Thus, there still exists the possibility for flows and functions to be incorrectly used, leading to modeling errors.

To overcome this deficiency, a grammar consisting of morphology and syntax is needed to provide structure to functional modeling with the Functional Basis. In its current form, the Functional Basis is best thought of as a lexicon, requiring the structure of grammar for consistent usage. From linguistics, we can find the following definitions for lexicon, morphology, syntax, and grammar [9–11].

- Lexicon: the vocabulary of a language; the complete set of meaningful units in a language.
2 Prior Work

and can be found in many texts on engineering design methodology for function and flow terms where functions take the form of signal flows into an overall functional model is presented. A series of building block examples to show how they might be formulated in Sec. 4. In Sec. 5, the signal grammar is then applied to the Functional Basis in its current form is a refinement of past attempts to fill the void in typical control system modeling techniques and to bridge the gap between functional modeling of control systems and other noncontrol based product interactions [42,43]. These methods, however, do not present a methodology to integrate system interactions not requiring direct control into their model. Rajan et al. [44] and Rajan [45] applied functional modeling to control systems by developing a four-step methodology for modeling control systems based on the reconciled Functional Basis. Their work looks at the development of the model for the controller and its associated input transducers. Through modeling a wide array of input transducers, Rajan et al. established a signal modeling methodology for mechatronic systems that tries to fill the void in typical control system modeling techniques and to bridge the gap between functional modeling of control systems and other noncontrol based product interactions [42,43]. These methods, however, do not present a methodology to integrate system interactions not requiring direct control into their model.

In an effort to model control systems at a design level, Chen and Törngren researched the functional requirements of control systems, giving primary focus to informational flow (either data, control, or mixed) and its characteristics as applicable in a control system [41]. The research of Chen and co-workers established a functional modeling methodology for mechatronic systems that tries to fill the void in typical control system modeling techniques and to bridge the gap between functional modeling of control systems and other noncontrol based product interactions [42,43]. These methods, however, do not present a methodology to integrate system interactions not requiring direct control into their model.

In an effort to standardize the assembly of functional models and to shorten the learning curve, Sridharan and Campbell put forth grammar rules in the form of function structures [7,8]. The grammar rules are based on products dissected and modeled with the reconciled Functional Basis and are archived in the University of Missouri-Rolla (UMR) Product Design Repository. The grammar rules are meant to capture all possible flows between function structures. Some problems with their grammar rules are that their rules were only established for energy and material flows due to an inherent ambiguity of signal flows that Sridharan and Campbell attributed to environmental unknowns. Because signal flows were omitted from their grammar, it becomes difficult to fully utilize the function grammar set to generate an accurate functional model integrating multiple engineering domains.

In an electromechanical design, signals are vital to establish the synergy among engineering domains. Increased recognition of the need for an integrated design among engineering disciplines through the acceptance of fields such as mechatronics has increased the need for a clarified methodology to systematically represent system level signal flows in functional models.

At the heart of many mechatronic systems lies a control scheme based on system sensory data, requiring comprehensive integration of sensors and actuators. Traditionally, control systems are modeled via a block diagram, where individual blocks represent an element of either the controller or the system and are connected with lines representing the flow of information [40]. Blocks are typically labeled to denote what dynamics they represent and contain a transfer function modeling the dynamic response of the system element. Block diagrams fall short in aiding the overall system design due to a failure to functionally represent all component interactions, whether through signal, energy, or material flows.

In an effort to model control systems at a design level, Chen and Törngren researched the functional requirements of control systems, giving primary focus to informational flow (either data, control, or mixed) and its characteristics as applicable in a control system [41]. The research of Chen and co-workers established a functional modeling methodology for mechatronic systems that tries to fill the void in typical control system modeling techniques and to bridge the gap between functional modeling of control systems and other noncontrol based product interactions [42,43]. These methods, however, do not present a methodology to integrate system interactions not requiring direct control into their model.

Rajan et al. [44] and Rajan [45] applied functional modeling to control systems by developing a four-step methodology for modeling control systems based on the reconciled Functional Basis. Their work looks at the development of the model for the controller and its associated input transducers. Through modeling a wide array of input transducers, Rajan et al. established a signal modeling methodology to model the transmitted signal and energy flows within each transducer. Their research, however, considers the signal and energy flows separately, not making the connection that energy is the carrier of the signal flow. The models contain knowledge that is useful for product dissection and failure analysis but are far more detailed than can be achieved during the functional stages of conceptual design.

3 Signal Lexicon

The reconciled Functional Basis by Hirzt et al. [6] establishes a set of definitions, which hold in them the guidelines governing how functions and flows should interact. This function and flow terminology for signals constitutes a lexicon for signal representation. In the following subsections, the signal as a functional modeling flow is defined, and a signal lexicon, as defined by the Functional Basis, is provided.

3.1 Signal Defined. What is a signal flow? A signal is typically defined as information about a system or its surroundings. In
Ref. [17], signal flows are defined as information for the “internal decision-making capability of a device or sensory data provided to or by a device or process.” Stone and Wood further defined a signal by clarifying that a signal is either a material or an energy flow with the specialized purpose of carrying information [1]. A signal, while being its own class, is actually a subset of either energy or material. All signals are some form of energy or material, but not all energies and materials are signals. This is due to the very nature of a signal. Since signal flows are information that will enter into, out of, or travel through a system, they require a carrier of sorts. A mailbox flag is an example of a material energy carrier where the flag carries the discrete control signal to pick up mail.

Often, it is important during the design phase to represent the signal flow carrier relationship. This relationship is represented using primary/carrier signal flow properties. Primary/carrier signal flow properties, shown in Fig. 1, allow the designer to represent a primary flow required to meet the input/output requirements of the block box model and its supporting carrier flow that aids the primary flow in the completion of its block box requirements. A primary flow, being of principle influence to the system, is described by each function block that it passes through. Primary/carrier flow properties are typically consistent across function chains; however, there are times when it may be necessary to represent changed functionality of a flow that in a prior block was represented by a carrier. Typically, this occurs when the carrier flow changes its form. Due to this ability for a primary/carrier property to change, it can be said that primary flow properties are independent of the function chain. From the mailbox example, the material flow (mailbox flag) is the carrier flow and the signal is again the primary flow.

3.2 Flow Lexicon. A signal, at the primary level of the Functional Basis, is a flow with the purpose of providing vital system data. The Functional Basis clearly defines signal flows at the secondary and tertiary levels [6]. Signal flows are defined at the secondary level as being one of two separate flow types, status or control, which are then further defined at the tertiary level. At this tertiary level, status can be auditory, olfactory, tactile, taste, or visual, while a control can either be analog or discrete. These flows are defined below [6].

Signal

• Status: a condition of some system, as in information about the state of the system
  ◦ Auditory: a condition of some system as displayed by a sound
  ◦ Olfactory: a condition of some system as related by the sense of smell or particulate count
  ◦ Tactile: a condition of some system as perceived by touch or direct contact
  ◦ Taste: a condition of some dissolved substance as perceived by the sense of taste
  ◦ Visual: a condition of some system as displayed by some image
• Control: a command sent to an instrument or apparatus to regulate a mechanism

○ Analog: a control signal sent by direct, continuous, measurable, variable physical quantities
○ Discrete: a control signal sent by separate, distinct, unrelated, or discontinuous quantities

3.3 Function Lexicon. The reconciled Functional Basis provides a set of functions that act on, with, or in response to signal flows. From the Functional Basis, there are two functions, actuate and regulate, defined below, requiring the input of a signal to control the magnitude of another flow in a separate flow chain [6].

• Actuate: to commence the flow of energy, signal, or material in response to an imported control signal
• Regulate: to adjust the flow of energy, signal, or material in response to a control signal such as a characteristic of flow

The Functional Basis also provides a set of functions defined to output signal flows representing system information in response to another flow. These flows, under the primary function class of signal, are defined to ascertain and provide vital system information back into the system or to the user. System information is provided in the form of a signal flow. The functions and their definitions are provided below [6].

Signal: to provide information on a material, energy, or signal flow as an output signal flow

• Sense: to perceive or become aware of a flow
  ◦ Detect: to discover information about a flow
  ◦ Measure: to determine the magnitude of a flow

• Indicate: to make something known to the user about a flow
  ◦ Track: to observe and record data from a flow
  ◦ Display: to reveal something about a flow to the mind or eye

• Process: to submit information to a particular treatment or method having a set number of operations or steps.

There is also a set of function terms in the Functional Basis that are defined to take in or put out control signals (as well as other flows) from a system. These terms, under the primary function class channel, are import and export and are commonly used when the system emits a control signal outside of its boundary to be received either by another system or by itself. The import and export function terms are defined below [6].

• Import: to bring in a flow (material, energy, signal) from outside the system boundary
• Export: to send a flow (material, energy, signal) outside the system boundary

Finally, the function convert also operates on control signals and can transform their state. The control function term is defined below [6].

• Convert: to change from one form of a flow (material, energy, signal) to another. For completeness, any type of flow conversion is valid.

4 Signal Grammar

A grammar has been enumerated from functional models modeled with the Functional Basis for both energy and material flows [7,8]; however, signals, while also having inherent Functional Basis guidelines, have no previously established usage grammar. In the following subsections, a signal grammar based on the Functional Basis for function and flow terminology is presented. The signal grammar consists of morphology guiding the interconnectivity of function and flow terms and syntax providing functional modeling templates. The developed grammar provides a structure.
4.1 Signal Morphology. The following 11 rules have been derived from the definitions provided by the Functional Basis and constitute the morphology for signal flows. Signal morphology guides the arrangement of signal related function and flow terms through a functional model and is meant to clarify how and when signal flows are to be applied in electromechanical products. To aid in the visualization of the morphology, Rules 3–12 are illustrated in Fig. 2.

- **Rule 1:** Use status signals to provide information on auditory, olfactory, tactile, taste, or visual states of the system.
- **Rule 2:** Use control signals to send an analog or discrete operational command to an instrument or apparatus.
- **Rule 3:** Use a primary flow to denote a signal and a carrier flow to denote its energy or material carrier.
- **Rule 4:** Use the actuate function to discretely toggle a flow (material, signal, energy).
  - Actuate functions require a discrete control signal to toggle state.
- **Rule 5:** Use the regulate function to adjust a flow quantity (material, signal, energy) in an analog manner.
  - Regulate functions require an analog control signal to adjust flow quantity.
- **Rule 6:** Use the sense function to detect or measure a flow (material, signal, energy).
  - Sense functions require an input of the flow of interest to output a status signal representing data collected.
- **Rule 7:** Use the indicate function to provide system status to the user.
  - Indicate functions end flow paths; thus, status flows exiting an indicate function block leave the system and do not connect to other function blocks.
  - Indicate functions do not receive a carrier flow.
- **Rule 8:** Use the process function to execute a series of operations to extract conditional information on a signal flow.
  - Either control or status flows can enter process functions;

however, respective entering flows must also exit.

- **Rule 9:** Use the convert function to perform the conscience act of changing a signal flow’s type.
  - A status flow input to the convert function should output as a system-usable control signal.
  - A control flow input to the convert function should output as a status flow.
- **Rule 10:** Use the import function to bring a control signal flow from the outside of the system boundary to the inside of the system boundary.
  - Flow arrows should be drawn into an input function block to represent flow into the system.
- **Rule 11:** Use the export function to send a signal flow outside of the system boundary.
  - Flow arrows should be drawn leaving the export function to represent control flowing from the system.
- **Rule 12:** Use the transfer function to move a signal flow through a system.
  - Either control or status flows can enter transfer functions; however, respective entering flows must also exit.

4.2 Signal Syntax. The previously presented signal morphology rules are used to build syntax for signals. The syntax manifests itself as functional modeling templates that can be modified and inserted into a functional model, aiding in the manual or automatic assembly of functional models, thus increasing the accuracy of product and design representation. Each syntax rule is explained and visually represented in the following subsections.

4.2.1 Actuator. An actuator, shown in Fig. 3, is a discrete control device used to turn another flow on or off. In a conceptual design, if it is known that a flow will be toggled, an actuator should be implemented. To functionally build an actuator, the actuate function must be used in conjunction with Rules 3 and 4.

A control signal, its carrier, and the flow to be toggled should be imported with import function-flow blocks, as shown to the left of the functional requirement arrow in Fig. 3. Then, following Rule 12, a transfer control function-flow block represents the routing of the control signal to an actuate flow function-flow block. Option ally, the actuator can indicate its status. To indicate the status, Rule 9 is followed to convert the control signal to a status signal and Rule 7 is followed to indicate the status through an indicate status function-flow block.

4.2.2 Regulator. A regulator, shown in Fig. 4, is an analog control device to adjust a flow in a variable manner. When it is known in a conceptual design that a flow is to be adjusted in a variable manner, a regulator should be implemented. To functionally build a regulator, the regulate term must be used in conjunction with Rules 3 and 5. To implement a regulator, a control signal, its carrier, and the flow to be adjusted should be imported with import function-flow blocks, as shown to the left of the functional requirement arrow in Fig. 4. The control signal and its
carrier are routed to the regulate flow function-flow block via a transfer control function-flow block following Rule 12. If the regulator indicates its status, Rule 9 is followed to convert the control signal to a status with a convert control to status function-flow block, and Rule 7 is followed to indicate status.

4.2.3 Sensor. A sensor, shown in Fig. 5, is a device used to detect or measure a flow and then output a signal representing collected information. Sensors would be used during conceptual design if a designer realizes that a design must ascertain information about itself or its surroundings. To functionally build a sensor, the sense term must be used in conjunction with Rules 3 and 6. To implement a sensor, transfer a flow (material, energy, or signal) to a sense flow function-flow block. The sense flow function-flow block outputs primary status flow and its respective carrier flow, which can then be transferred with a transfer status function-flow block by Rule 12. The final destination of the status flow and carrier can be the system, the user, or both.

The sense function can be further detailed at the tertiary level by the reconciled Functional Basis with its tertiary functions detect and measure. The general sensor functional model, shown in Fig. 5, can modified to reflect the increased detail of tertiary terms by using one of the two tertiary terms in place of the sense function.

4.2.4 Indicator. An indicator, shown in Fig. 6, is any device with the goal of providing vital system information to the user. A designer might use an indicator during a conceptual design when it is known that some form of feedback is required from the system. To functionally build an indicator, the indicate term must be used in conjunction with Rule 7. To implement an indicator, run a status flow from the function-flow block from which system information should be obtained. An indicator is the exception to the primary/carrier rule (Rule 3) since it can be as simple as the operation of the system or as complex as a series of components providing full diagnostics on system behaviors; in either case, however, an indicator is not required to send the signal outside of the system boundary. An indicator status function-flow block ends a flow path (Rule 7), thus the status flow that exits an indicate status should not enter another function block.

As with the sense function, the indicate function can be further detailed at the tertiary level by the reconciled Functional Basis with its tertiary functions track and display. The general indicator functional model, shown in Fig. 6, can be modified to reflect the increased detail of tertiary terms by using one of the two tertiary terms in place of the indicate function.

4.2.5 Processor. A processor, shown in Fig. 7, is any device that analyzes a status signal obtained from a sensor that has ascertained information, either internal or external to the system. Following signal analysis, the processor sends control information to system elements. A processor might be used during a conceptual design if a designer knows that the product will need to analyze the state on a series of conditions and make decisions based on the analysis. To functionally build a processor, the process term must be used in conjunction with Rules 3 and 8. To use a processor, run a primary status flow and a carrier flow into a process status function-flow block. Following Rule 9, to get a system usable control signal, connect the process status function-flow block to a convert status to control function-flow block with another primary status flow and carrier flow pair. Then connect the convert status to control to a transfer control function-flow block with a primary control signal and carrier flow. By the application of Rule 12, the control and its carrier flow is routed via a transfer control function-flow block to the controlled system elements.

4.2.6 Receiver. A receiver, shown in Fig. 8, is used to bring a control signal into the system. To functionally build a receiver, the import term must be used in conjunction with Rules 3, 10, and 12. To implement a receiver, a new control signal flow and its respective carrier must be imported into the system and thus cross the system boundary, as shown to the left of the functional requirement arrow in Fig. 8. The primary and carrier flows are then routed into the overall system by tying the import control function-flow block to a transfer control block.

4.2.7 Emitter. An emitter, shown in Fig. 9, is used to send a control signal from the system. To functionally build an emitter, the export term must be used in conjunction with Rules 3, 11, and 12. To implement an emitter, run a control signal flow and its carrier flow through a transfer control function-flow block. Then, tie the control signal and its carrier to an export control function-
flow block. From the export control function-flow block, draw an exiting control signal flow and its carrier to represent them leaving the system boundary.

5 Application

As an application of the signal usage grammar, consider the following building block examples representing five basic types of control systems. Following that, a complete example of a more complicated electromechanical product is presented.

5.1 Building Block Examples. The following building block examples investigate the aforementioned signal flow syntax: actuator, regulator, sensor, indicator, processor, receiver, and emitter. The syntax is matched to individual components to show how they might be applied to represent an actual engineered system.

5.1.1 Switch. A switch is a mechanical device based on the actuator syntax. It is designed to either initiate or terminate a flow based on one of its two discrete states. Provided in Fig. 10 is a sample functional model of a switch. The operation of the switch requires imported mechanical energy to change the switch state. By Rule 3 for primary and carrier flows, mechanical energy is represented as the carrier flow used to set the switch state. By Rule 4 for actuate functions, a discrete control signal, carried by the mechanical energy, is used to actuate the mechanical energy flow. It is important to note that the mechanical energy used to set the switch state is separate from the electrical energy that will pass through the switch. For user feedback, Rule 9 for convert functions is applied to convert the control signal to a status, and the indicator syntax is applied via morphology Rule 7 for indicate functions.

5.1.2 Valve. A valve is a mechanical device based on the regulator syntax. It is designed to adjust a flow in a variable manner. Provided in Fig. 11 is a sample functional model of a valve. The operation of the valve requires mechanical energy to be imparted on the valve to regulate flow. By Rule 3 for primary and carrier flows, mechanical energy is functionally represented as the signal/energy flow and the mechanical energy carrier is represented functionally following Rule 8. The signal processor analyzes the status signal obtained from the thermostat, converts the status to a control signal, and transfers the control signal to the regulator. The control signal transferred into the regulator system is carried by an electrical energy flow that is used to regulate the flow of electrical energy in the example. The regulator system is

5.1.3 Transistor. A transistor is an electrical device based on the regulator syntax and Rule 5. It is used in electronic devices to adjust the flow of electrical energy. Provided in Fig. 12 is a sample functional model of a transistor. Transistors do not inherently provide user feedback; thus, the sample functional model in Fig. 12 does not include the optional indicator syntax. The operation of a transistor is controlled through an applied voltage signal (electrical energy) to the transistor’s base, which allows a current (electrical energy) to pass. The voltage signal is a signal/energy executor whose applied value changes the current flow by altering the transistor’s configuration. The relationship between the signal flow and the electrical energy carrier is represented functionally by following Rule 3 for primary and carrier flows. In a perfect transistor, there would be no interaction between the signal/energy executor (voltage signal) and the electrical energy (current) whose flow is being controlled.

5.1.4 Thermostat/Processor. A thermostat is based on the sensor syntax and follows Rule 6. It is designed to detect or measure thermal energy and provide a primary status signal carried by an electrical energy flow. Provided in Fig. 13 is a sample functional model of a thermostat transferring system information into a functional model of a signal processor. The signal processor is based on the processor syntax following Rule 8. The signal processor analyzes the status signal obtained from the thermostat, converts the status to a control signal, and transfers the control signal to the system. The control signal transferred into the regulator system (the four functions operating on the flow of electrical energy) is carried by an electrical energy flow that is used to regulate the flow of electrical energy in the example. The regulator system is
5.1.5 IR Emitter/Detector. An IR emitter is based on the emitter syntax which follows Rule 11 for export functions. Its purpose is to export a control signal carried by an electromagnetic energy flow outside of the system. An IR detector is based on the receiver syntax, which follows Rule 10 for import functions. An IR detector’s purpose is to import a control signal carried by electromagnetic energy into the system. Provided in Fig. 14 are sample functional models of both an IR emitter and an IR detector. The IR emitter sends an electromagnetic energy signal, which, by Rule 3, carries a discrete control signal. The control signal is detected by the IR detector, which acts as a switch where the signal/energy executor is an electromagnetic control signal mix that actuates an electrical energy flow. The electromagnetic energy is converted to electrical energy and applied to the transistor, thus activating the current flow through the transistor. In the example, the emitter and receiver syntaxes are used in conjunction with an actuator syntax to actuate an electrical energy flow in a system.

5.2 Electromechanical Product Example: Digger Dog. The electromechanical dog simulator, Digger Dog, has been dissected and modeled functionally following the three-step functional modeling procedure outlined by Stone and Wood [1]. Digger Dog, shown in Fig. 15, uses one of three input options to start a series of events (kicking, barking, and tail wagging) to simulate a dog digging a hole in the yard [46]. As such, the Digger Dog product is a signal-rich environment in which to test the signal grammars. Audio and motion sensors allow for a level of automation where a Digger Dog senses ambient sounds and motion to automatically start its simulation events. A third input option is a manual “try-me” button that, when pressed, starts the simulation events. When not in operation, a Digger Dog sits idle waiting for a control signal.

Fig. 16 Digger Dog black box model

5.2.1 Step 1: Generate Black Box Model. The first developmental step to the generation of a functional model is to create a black box model representing the design or product’s overall functionality in verb-object form. The black box is drawn as a single function block with a series of input/output flows, which are required to achieve a high-level functionality.

The generation of a black box model for a Digger Dog involves determining its core functionality and then determining all input/output flows that are required to complete the determined core functionality. Since the focus of this paper is signal flows, signals will be the primary input/output flow focus.

As previously mentioned, Digger Dog is a toy designed to simulate a dog, which makes its core functionality in verb-object form to simulate dog, shown in the black box model provided in Fig. 16. A Digger Dog has four signals, on/off, sound, motion, and manual operation. The on/off control signal is the master control having to be set in the “on” position to allow simulation. Manual operation is performed by a push button that starts the simulation, while the sound/motion feature allows for a level of automation that automatically turns on the simulation. The two output signals are the position of the on/off master switch and actual visible reaction from the simulation.

5.2.2 Step 2: Create Function Chains for Each Flow. The second step to generating a subfunctional model is to generate function chains for each input flow. Each subfunction chain should consider all changes and operations that occur in each flow.

Again, considering only the signal input flows, function chains are generated for the sound, motion, on/off, and manual operation signal inputs. From Rule 6 of the signal flow morphology, detect-
tion of sound or motion requires the use of a sensor. Sensor functional models for detecting sound and motion are generated by applying the sensor syntax and are provided in Figs. 17 and 18, respectively. To sense sound, acoustic energy must first be imported into the system so that it can be sensed with a sense acoustic energy function block. Motion is detected by first importing electromagnetic energy into the system to be sensed with a sense electromagnetic energy function block. To turn the status signals generated by the sense function blocks into control signals, the processor syntax, consisting of a process status function block and a convert status to control function block, is implemented. Control signals that can be used to activate the simulation of a Digger Dog are now generated.

The on/off and manual operation signal inputs are switches and

![Fig. 21 Aggregated functional model of a Digger Dog](image-url)
are modeled following the actuator syntax. The on/off switch, like the switch building block example in Fig. 10, contains the optional user status indicator via position indicator markings. The manual operation switch, however, does not have a user indicator, relying instead on the simulation to prompt the user of a proper operation. Figure 19 provides the subfunction chain for the on/off switch, and Fig. 20 provides the subfunction chain for the manual operation switch. Both subfunction chains import human energy as a means to actuate the switch. The human energy is then converted to mechanical energy, which is converted to a control signal. The control signal is carried by mechanical energy (1) to actuate the master on/off or (2) to actuate the dog simulation. The manual operation subfunction chain ends with transfer control since there is no user status indication. However, the control signal in the on/off subfunction chain is converted to a status that is then indicated to the user.

5.2.3 Step 3: Aggregate Function Chains Into a Functional Model. The final step is to aggregate the functional model. During this phase, subfunction chains are connected and new subfunctions may be added. The new functional model will be a comprehensive model representing the whole, unified functionality of the product.

For the Digger Dog example, each of the signal subfunction chains have been aggregated into the energy and material subfunction chains to create the unified functional model provided in Fig. 21. The motion and sound detection have been aggregated together to use the same process status, convert status to control, and transfer control function blocks. The motion/sound control signal is then aggregated with the electrical energy function chain to control the flow of electrical energy via an actuate electrical energy function block in the simulation part of the functional model. The manual operation is also used to control the flow of electrical energy and is transferred to the same actuate energy function block in the simulation portion as the motion/sound detection signals. The on/off control signal has been transferred into the master actuate electrical energy function block, and upon leaving the master actuate electrical energy function block, the control signal is converted to a status signal that is indicated to the user. The final signal functions to be added to the aggregated functional model are the indicate status function blocks. An indicate visual function block has been added to both of the export mechanical energy function blocks where one represents a Digger Dog’s leg kicking and another represents the tail wagging. A final indicate visual function block has been added, leaving the export acoustic energy function block representing the output of the acoustic energy for a Digger Dog barking.

To illustrate how each function-flow pair works in the Digger Dog system, the function-flow pairs within the aggregated functional model have been mapped to a component within the Digger Dog system. A complete component breakdown including all primary and secondary functionalities can be found on the UMR Design Repository.

6 Conclusions

The signal grammars presented address semantic deficiencies of a model structure when generating functional models. The signal grammar, comprised of morphology and syntax, and the Functional Basis lexicon provide the nomenclature and structure, allowing for a more uniform functional modeling. This uniformity helps to ensure the understanding and consistent archival of design information. The addition of carrier flow information aids in the capturing and archiving of vital system information, allowing future designers to better understand the original intent and functionality of archived product information. Signal usage syntax provides templates that can be modified for design intent to create more complete and correct functional models of electromechanical products. In particular, the grammars, when combined with the previous work of Sridharan and Campbell [8], support manual creation of functional models through rote application of the rules, given a black box as a starting point. Also, as Sridharan and Campbell have previously shown, the grammars can be integrated into an automated computer application to generate functional models. Building block examples demonstrate a syntax template application with various common electromechanical components.

Limitations of this work derive mainly from its inductive nature. Since all instances of signal functionality are not observed, there is no way to know if the grammar set is complete. However, a review of control systems literature has identified the basic control elements that must be represented in order to model electromechanical and mechatronic products. Additionally, grammars are notoriously difficult to implement. This fact can be mitigated through the use of the active center concept from the polymerization phenomenon [47,48], as utilized by Sridharan and Campbell [7,8].

Future work will aim at the integration of grammar into a functional modeling CAD package so as to identify discrepancies in a functional modeling structure. The software package will provide the designer with templates based on the syntax to aid in the functional model generation. Also, to validate the uniformity of functional models when a grammar is employed, case study functional models will be generated for electromechanical products with elements of automation that rely heavily on signal flows for overall functionality.

References


4http://function.basiceng.umm.edu/repository


