INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

INVESTIGATION OF SYSTEM SENSITIVITY TO PROPAGATED CONFIGURATION FAULTS

Robert L. Nagel[1], Robert B. Stone[1], James A. Greer[2] and Daniel A. McAdams[3]
(1) Oregon State University (2) US Air Force Research Lab (3) Texas A&M University

ABSTRACT

Research on the propagation of system faults occurring due to the loss of expected configurations and operations is reported in this paper. A technique known as process analysis is applied to model the activities surrounding a product's usage to investigate the propagation of faults along flow paths critical to the successful operation of the product. To quantify the sensitivity of the product to the propagated faults, two sensitivity measures are provided: (1) a qualitative rating detailing the model level to which a fault propagates and (2) a quantitative sensitivity metric indicating a percentage of changes that can no longer be performed due to a given fault. Sensitivity metrics provide a gauge to indicate the amount a process is affected by the fault and as a tool to focus redesign efforts. When applied with the propagated failure analysis, the methods allow a designer to evaluate multiple design configurations, determine where unwanted redundancy might exist or where redundancy might be required, and make judgments as to where safety planning might be required to increase system robustness against undesirable faults.

Keywords: Fault propagation, configuration failure, sensitivity, process, function

1 INTRODUCTION

During product design, the design scope is often limited to a single, primary operation of the product and the design activities focus on the functionality of that product in a single operational environment. Customer needs are identified, requirements set, functionalities determined, components selected, failure analysis performed, and, finally, the product is fabricated, marketed and sold. This process works well for simple products; however, as products increase in complexity so, too, do the models representing them. Products contain functionality to cope with the entire system, which includes the various processes where the product will be used as well as the operating conditions and environments where the product will be used. Products also exhibit failures not only because of faults with internal components but also because of faults related to the product's configurations, operating conditions and environmental influences, and while traditional failure analysis techniques are designed to identify component faults, they often fail to identify faults propagating from the system beyond the physical product.

This paper presents the following three techniques for the identification, classification and analysis of system failures occurring from process faults related to product configurations: (1) Propagated Fault Analysis (PFA), (2) Process Fault Levels (PFL), and (3) System to Process Sensitivity (SPS). These techniques are based on process models which detail the outcomes expected of the product, the actions that the product is expected to perform and the configurations required for operation. Process models are based on traditional functional models, and, like functional models, represent flow transformations rather than specific components allowing their application during conceptual design when components and solution principles are rarely known. Flows of materials, energies and signals connect transformations in both events and configurations, and product operations are described functionally. Within process analysis there are two major types of models: (1) event models detailing the environments or situations where a product will be used and (2) configuration models which functionally model changes occurring to the entire product [1, 2].

PFA guides a designer through the investigation of a fault occurring at critical points in a process model to reveal how the loss of a flow vital to the system adversely affects the operation of the product. Potential faults are identified and their propagation is traced along faulted flows. Two sensitivity metrics, PFL and SPS, when taken together, provide sensitivity measures to gauge the scope and criticality of projected faults. The qualitative sensitivity rating, PFL, provides an initial assessment for the impact a fault on the overall process based on projected scenarios for the continued operation of the process. Also, PFL provides a way for the designer to focus on those faults, which have the potential to be more devastating to the system as a whole. The second sensitivity metric, SPS, is quantitative, and is used as an indicator to the loss of configurability of a product. This paper covers the generation of process models and the methodology for the application of PFA, PFL and SPS through the example of an automatic grind and brew coffee maker.

2 RELATED WORK & BACKGROUND

A number of tools currently exist to investigate failures and failure propagation; these methods, however, tend to focus on faults due to failures of internal components and/or functionality of the product being designed. Many of these methods are component specific making them ill suited for application during early phases of design before the components are completely known, or fail to provide a technique to propagate failures occurring beyond the product's boundaries within the system at large. The following sections enumerate on related failure analysis techniques as well as on the process modeling technique employed in this research.

2.1 Related Work

Failure Modes and Effects Analysis (FMEA) is considered the industry standard methodology for failure analysis. FMEA was originally developed from the failure modes and effects criticality analysis (FMECA) defined in MIL-P-1629A [3, 4]. An effort was later made by Ford, Chrysler, General Motors and the Automotive Industry Action Group to standardize FMEA, and a reference manual was published [5]. FMEA provides a technique where potential failure modes, consequences and severity are identified for each component in a system. Unfortunately, both component and expert knowledge is required from the team performing the analysis if FMEA is going to effectively identify possible failures and consequences for each of a product's component, thus making FMEA better suited for product redesign.

Efforts have been made to overcome the shortfalls of standard FMEA and to move failure analysis into the conceptual design phases with alternative FMEA-based approaches utilizing intended system functionality instead of component specific knowledge. Russomanno has proposed an Expert System for FMEA (XFMEA) [6]. XFMEA automates FMEA activities and utilizes behavioral, functional and structural representations that allow failure analysis activities to be performed before system components are known. A knowledge base is aimed at assisting the expert-based failure analysis team. Advanced FMEA also tries to bring the failure analysis activities into the conceptual design phase by applying behavior models, which map control-based functionality to system components [7, 8]. Where there are deviations from the intended functionality, a failure has occurred. Function Hazard Analysis (FHA) is performed based on the system's and then the subsystem's functional decompositions allowing it to be performed early in the design stages [9]. Experts determine potential failures based on the behaviors of the system's functions to determine function-failure combinations. FHA, however, relies on experts to determine potential failure modes for a system. The Function Failure Design Method (FFDM) utilizes functional modeling with the Functional Basis to apply failure analysis during the conceptual design phase and overcome the shortfalls of expert-based systems through the application of a knowledge-based repository of failure data [10, 11].

These aforementioned FMEA-based approaches, however, fail to identify the cascading or propagation of failures through a system. Fault tree analysis and event tree analysis are specifically designed to investigate failure propagation. Fault Tree Analysis (FTA) applies backward logic to develop a top-down chain of events which have the potential to lead back to a single negative event [12-14]. Events propagating to the negative event are modeled using Boolean logic (AND logic and OR logic gates) to create chains of potential propagated failures. Event Tree Analysis (ETA), conversely, uses forward logic to investigate a single initiating event. From a single initiating event, probable failures of events, which can occur in sequence, are analyzed to determine the likelihood of success or failure [3, 15]. Both ETA and FTA can be performed on a system during conceptual design to identify failures both internal to the system and external to the system. Probabilistic Risk Analysis (PRA) combines the failure propagation techniques of FTA and ETA with failure effects identification techniques such as FMEA to answer three questions: what can go wrong, what is the severity, and what is the liklihood [3, 15, 16]. PRA can be performed on initiating events that are either internal or external to the system through all phases of a product's life cycle. PRA, like FMEA, FTA and ETA, tends to rely on expert knowledge and does not typically employ structured modeling as an abstraction of the system.

To analyze fault propagation within a product during conceptual design, function-based failure propagation and Functional Failure Identification and Propagation (FFIP) have both been proposed. Both methods apply functional modeling with the Functional Basis to failure identification. Function-based failure propagation identifies two failure modes, "No Flow" and "No Failure," which occur due to the propagation of failures along flows [17]. Failures and their likelihoods are identified and calculated based on information stored within a failure knowledge base. FFIP identifies failures in a system occurring from the loss of functionality [18]. Functional failures occur from negative events in a behavior model and are propagated through the functional model to determine the impact to the system.

Each of the previously mentioned techniques for identifying failures and their resultant faults deal primarily with the workings of the system. Fault propagation graphs, however, are based on structured hierarchical process and sub-process models and apply causal relationships to a set of possible failure modes [19]. Fault propagation graphs applied to process models, can be used to analyze failure propagation between the configurations where configurations are defined as the different phases of an entire process. AND and OR causal relationships are used to model the propagation of failures from a process to its sub-processes. In this same vein, graph theory-based approaches may be utilized to analyze the time and resource requirements of processes via Program Evaluation and Review Technique (PERT) charts whereby processes are modeled as network diagrams with the nodes representing events and the arcs representing time and monetary constraints [20]. Unexpected events or developments are modeled through random graph structure. To measure the effect of these unexpected events, Bowman presents a sensitivity analysis, to estimate resultant probability distributions for program performance measures based on changes to the constraints [21].

2.2 Process Modeling Background

Process modeling provides a technique to model the events where a customer will use a product and configuration changes occurring to a product during normal or expected operations [1, 2]. Process models are based on functional modeling techniques [22] to allow for integration with functional modeling during conceptual design activities [23]. When applied during conceptual design, the combination of functional and process models allows for integrated modeling of the entire product's system where functional models focus *inside* of the product on the specifics of how the product will operate and process models focus *outside* of the product on how the customer will interact with the product. To clarify the nomenclature associated with process models the following terms are provided:

- **Process Modeling** – The overall approach to modeling a series of customer-driven, product-based actions related through input and output flows, the product being designed, and time.
- **System** – The combination of artifacts and actions, which together form a complete, operational product. In the realm of product design, components are combined to produce a complete functioning product to meet identified customer needs. The system combines the product with its use and its usage environment.
- **Configuration** – A specific discrete instance of the overall functionality of the product. A configuration is modeled as a functional change to the product as a whole.
- **Event** – A set of configurations of a product, which may relate to the environments where the product is used, changes to the operability of a product, specific applications of a product, or sequencing of operations during the usage of a product.
- **Process** – The sum of defined events that occur with respect to the product as a whole and aim to meet a particular goal. Processes are tied together via the product being designed, material, energy and signal flows.

Process modeling considers two levels of models: (1) an event level model and (2) a configuration level model. Event level models, like functional models are often modeled at two levels of fidelity: (1) a black box model for the process and (2) its decomposition. Flows at the highest-level model (black box) are the sum of the flows required by lower-level event and configuration models. Event models provide a hierarchical decomposition of the process and are comprised of multiple sub-events.

A configuration model is a decomposition of a single event representing discrete functional changes to the system as a whole. An illustration of this model hierarchy is provided as Figure 1.

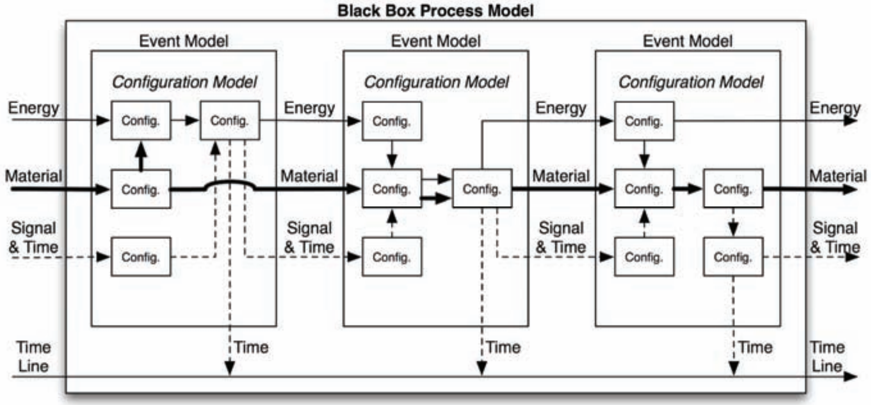

Figure 1: Example of the process modeling hierarchy containing the black box, event and configuration models.

To generate a process model, the following six steps are followed: (1) Identify the overall process to be completed and the requirements necessary to complete the process; (2) Generate a black box model for the process being modeled defining the overall process, the product being designed and all required material, energy and signal flows; (3) Identify and formulate events necessary to complete the process as well as their required input/output flows; (4) Formulate the event model consisting of chains of event; (5) Decompose each individual event into a more detailed configuration model detailing the discrete changes to the product; (6) Verify that each process requirement is addressed within the process models [1, 2].

As an example of the generation of a process model, consider an automatic grind and brew coffee maker. For this paper, an existing design is modeled to simplify the demonstrations; however, the models and fault analysis procedures would be applied identically for a conceptual design problem. For a design problem, the first step to generating a black box model for the process of brewing coffee with the coffee maker is to understand the customer's expected outcomes and objectives. Of course, the customer's expected outcome is to brew a cup of coffee; however, to focus the modeling, the overall objective is narrowed to the brewing of 12 cups of coffee from whole beans. To accommodate the whole beans, the coffee maker will need to combine a coffee bean grinder with a coffee maker; through this paper, this combined product will be called a grind and brew coffee maker. For the actual product, these customer needs are met as follows: Operation of the grind and brew coffee maker beings with the operating setting up the machine by placing the machine firmly on a solid surface and supplying power. Fresh beans are loaded into the grinder at the top of the coffee maker by first lifting the reservoir cover and then removing the grinder cover. Beans are loaded directly into the grinder, and then the grinder cover is replaced. Water is loaded into the water reservoir adjacent the bean grinder. Once beans and water are loaded, the reservoir cover is closed, and the filter basket, filter basket cover and filter basket holder are swung into placed. Only once the reservoir cover and the filter basket compartment are properly secured can the coffee maker brew coffee. During this brewing process, coffee beans are ground, and then transferred to the filter basket. Water is heated, distributed over the ground coffee beans, and passed through the grounds before being directed into the carafe.

The event model, shown in Figure 2, provides an abstract model of the operation of the grind and brew coffee maker (which is itself only a part of the larger lifecycle model of the product). The steps to brewing coffee with the grind and brew coffee maker include: general setup, loading of coffee beans, loading of water and operation. Each of these steps are modeled as an individual event, which is represented as a single box in the event model (Figure 2). Each event is named with two verb-noun pairs, one with free language and a second with the Functional Basis. The setup coffee maker event requires the material flows of the *coffee maker*, *a power receptacle*, *human* and *surface*. These material flows, and all other material flows, are modeled with a bold arrow following functional modeling conventions. The energy flows, modeled with a thin arrow, are *human energy* and *electrical energy*, and the signal, *visual,* modeled with a dashed arrow. Following the convention established in [1], flows required for multiple events traverse the event model from left to right, while flows unique to a specific event traverse an event from the top to the bottom. Thus, all of the flows that enter the first event also enter the second event, *load beans*, except for the visual signal. The second event also receives the additional material flow of beans. All flows continue into the third event, *load water*, where the material flow, *water*, is added to the system. A signal flow also leaves the third event indicating the water level to the operator. In the final event, *operate coffee maker*, the signal flow, *on*, is required to activate the coffee maker. Leaving the operate coffee maker event are, in some form, all of the flows which have entered the system including the material flows: *coffee maker*, *coffee*, *grounds*, the *power receptacle*, *human*, *surface* and *steam*; the energy flows: *thermal* and *acoustic*; and the signal flows: *olfactory*, *auditory* and *visual*.

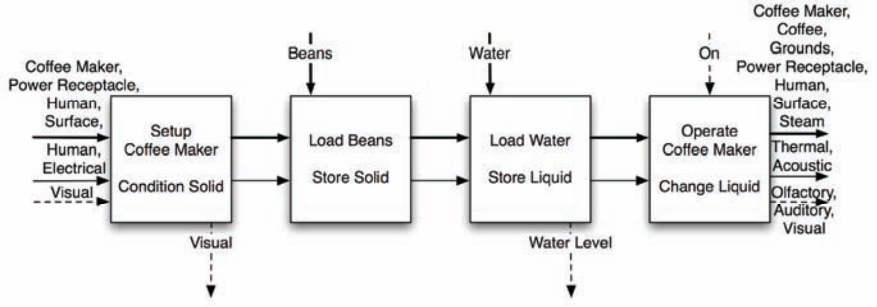

Figure 2: Event model for the automatic grind and brew coffee maker

Each of the events in the event model for the coffee maker (Figure 2) can be decomposed into a configuration model detailing the specific changes required to complete each event in the coffee brewing process. A configuration model is modeled similarly to a functional model using function flow pairs; it is important to note, however, that unlike functional models, configuration models include the product as a unique flow to capture interactions with the product as a whole. The configuration model for loading beans, provided in Figure 3, imports four flows: the *coffee maker*, *human material* and *energy*, and *coffee beans* (modeled as a *solid material*). The human material and energy flows, which work together, are coupled together in the same flow chain in the diagram, and are modeled as being guided to twice change the coffee maker. The first *change coffee maker* block represents the opening of the reservoir lid, and the second represents the removal of the grinder lid. The loading of coffee beans is modeled first as a *transfer solid* to move them into the system and then as a mixing of the coffee maker and the beans. Once the coffee beans are loaded, the grinder lid is replaced (modeled as *secure coffee maker*) before the *coffee maker*, *human material* and *energy* are exported from the system.

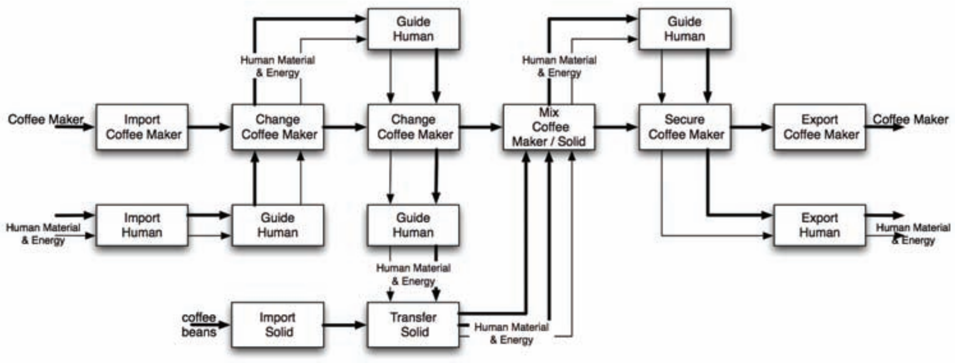

Figure 3: Configuration model for the loading of coffee beans

Configuration models for each of the other events – *setup coffee maker*, *load water*, and *operate coffee maker* – in the coffee maker process are generated similarly to the configuration model for loading of the coffee beans by considering the operations occurring on each flow. This method is considered "being the flow" [24] and is used to imagine how the flow is routed, operated upon and worked with by the operations within the system.

3 APPROACH & METHODOLOGY

Propagated Fault Analysis (PFA), Process Fault Levels (PFL) and the associated sensitivity metric, System to Process Sensitivity (SPS), grew out of a joint project between the U.S. Air Force Academy (USAFA) and Missouri University of Science and Technology (Missouri S&T) to investigate terrorist activities involving Improvised Explosive Devices (IEDs). The researchers applied process and functional analysis to IEDs to study their development and deployment from a design engineering perspective. The IED process models generated by the researchers were analyzed for possible fault points to help find potential weaknesses that might be exploited in efforts to prevent future IED events. PFL and SPS were developed as sensitivity metrics to assess various potential faults and to indicate the impact of faults on the IED process such that potential faults could be assessed to determine the value of affecting the identified fault points. Thus, PFA, PFL and SPS were developed for a purpose opposite to how they would be helpful for a designer. The methods were developed to find and assess faults in the actions surrounding a system that might be exploited to terminate a system prematurely. However, these methods applied to a system during preliminary design would allow the mapping of fault propagations in the process models of a product's operation to shed insight into vulnerable product configurations and would allow a designer to plan against potentially devastating failures. The general method taken to investigate process-level faults is summarized in Figure 4.

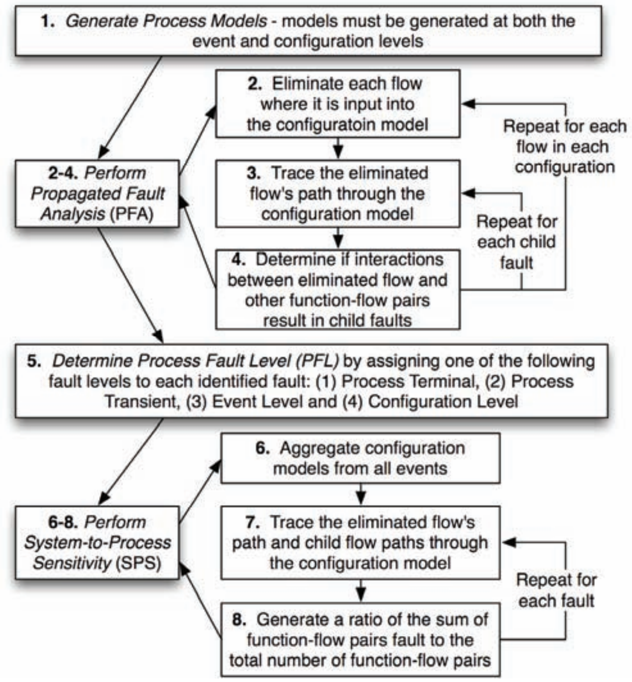
1. Generate Process Models - models must be generated at both the event and configuration levels
2-4. Perform Propagated Fault Analysis (PFA)
2. Eliminate each flow where it is input into the configuratoin model
3. Trace the eliminated flow's path through the configuration model
4. Determine if interactions between eliminated flow and other function-flow pairs result in child faults
Repeat for each flow in each configuration
Repeat for each child fault
5. Determine Process Fault Level (PFL) by assigning one of the following fault levels to each identified fault: (1) Process Terminal, (2) Process Transient, (3) Event Level and (4) Configuration Level
6-8. Perform System-to-Process Sensitivity (SPS)
6. Aggregate configuration models from all events
7. Trace the eliminated flow's path and child flow paths through the configuration model
8. Generate a ratio of the sum of function-flow pairs fault to the total number of function-flow pairs
Repeat for each fault

Figure 4: Methodology for analyzing propagated system configuration faults

3.1 Steps 2 - 4, Propagated Fault Analysis (PFA)

First, process models of the system are generated following either the approach reviewed in Section 2.2 or a similar technique. Once process models are generated, propagated fault analysis is applied to the models by considering the elimination of flows at three locations: (1) flow importation into a model, (2) flow crossing in a model, and (3) flow branching in a model. PFA investigates how changes to these flows, vital to an overall process, affect the desired final outcome. Thus, PFA is based upon each flow in a process model having a key role in the successful completion of the overall functionality. If a function or flow is disrupted, the final desired outcome is also disrupted. The elimination of flows is propagated through the configuration models, where the eliminated flows are denoted with an X in the model. As faulted flows are propagated, potential interactions, which result in other flows being eliminated, should be considered. Flows eliminated by other faulted flows are termed child faults, and they too are traced through the system recursively considering new interactions.

PFA utilizes two failure methods to show failure propagation through a process model. First is a "No Flow" failure, which has been taken from function-based failure propagation [17] and occurs when a function (or configuration change) fails. This failure results in the termination of the flow on which the function acts allowing the failure to propagate along the flow path. The second failure method is a "No Function" failure where a specific function (or configuration change) fails, but flow through the configuration is not affected and the following function-flow chain continues to be operational, albeit potentially degraded.

Consider again, as an example, the loading coffee beans event from the automated grind and brew coffee maker modeled in Figure 3. The configuration model for this event includes the configuration changes for removing and replacing two lids and coupling beans with the system. PFA performed on the configuration model describing the bean loading event would be used to find potential faults that could occur during interactions with the product while removing or replacing the lids or coupling the beans with the product. If a failure occurs with the operator trying to remove the outermost lid, then neither coffee beans nor water can be added to the coffee maker resulting in "No Function" faults for the coffee maker product flow and "No Flow" failures for the operator flow. The initial failure of not being able to open the outermost lid is represented with shaded X's in Figure 5, and the resultant propagation is represented with unshaded X's. The coffee maker product flow can no longer perform the required configurations following lid removal, and the operator no longer has a product to operate upon.

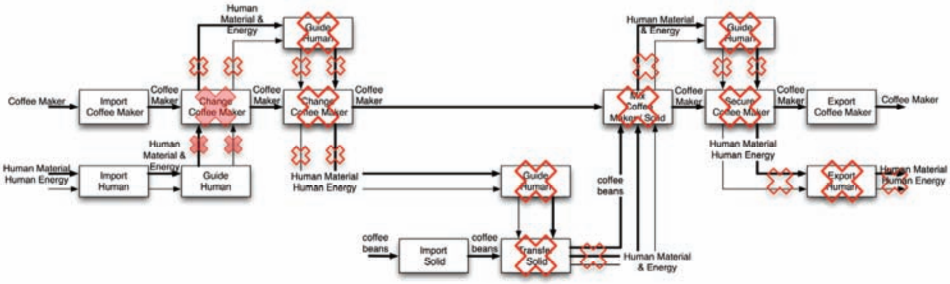

Figure 5: Faulted operator flow when opening the reservoir lid

If a fault, however, occurs with the coupling of the coffee beans with the product, all other configuration changes around the coffee maker product can still occur, and the coffee maker will still try to brew coffee. Hot water, of course, will collect in the carafe. Figure 6 shows the configuration model with the flow of beans faulted. The initially faulted flow and function block are again represented with shaded X's, and the propagation is represented with unshaded X's. This fault is a "No Function" failure since the failure of the configuration mix coffee maker / solid does not result in the termination of the coffee maker flow.

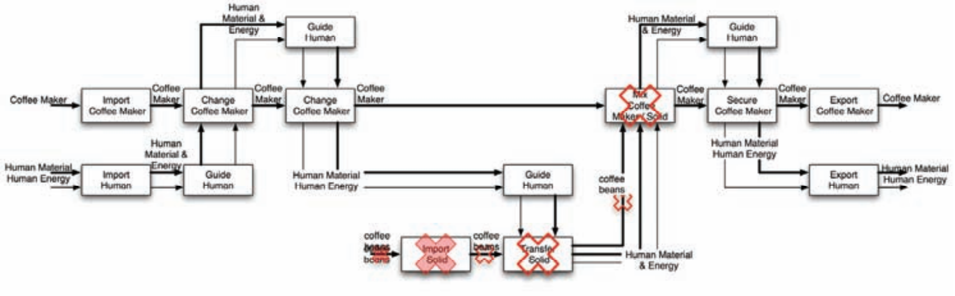

Figure 6: Faulted coffee bean flow during the loading of coffee beans event

3.2 Step 5, Propagated Fault Level (PFL)

At the completion of PFA, configuration models should be marked with a number of X's denoting identified fault paths. Each fault is rated with a PFL considering whether the fault affects a single configuration, multiple configurations and consequently an entire event, or the entire process. PFL is a qualitative rating of the impact a fault has on the overall process based on projected scenarios for the continued operation of the process, and PFL provides a way for the systems analyst to focus on those faults, which have the potential to be more devastating to the system as a whole.

From the PFA performed on the modeled IED processes, four distinct types of faults were identified which has lead to four PFL ratings: (1) process terminal, (2) process transient, (3) event level or (4) configuration level. The fault level is assigned based on how far a fault propagates through a process; for instance, a configuration level fault affects only a change in a system's configuration. An event level fault, however, affects the functionality of an entire event. The final two types of faults are process terminal and process transient. With transient process faults, the process may be restarted, and with the terminal process faults, the process is irreparably faulted and cannot be restarted. The fault levels are:

- **Process Terminal Fault** - ends a process completely. The process cannot be restarted. Process Terminal is the most severe fault.
- **Process Transient Fault** - ends a current instantiation of a process. The process can be restarted at a future time.
- **Event Level Fault** - occurs when the process is not stopped. Instead, an event within a process does not function properly.
- **Configuration Level Fault** - occurs when a single configuration change within an event can no longer occur. Configuration faults do not cause the entire event to fault, yet may degrade the performance of the event.

Since failures propagate through an entire process, a minor failure during one event might result in a more significant failure at a later event. Thus, a single failure might result in more than one PFL. Also, the fidelity to which a process is modeled could result in a shift (either more severe or less severe) in the assigned PFL. Thus, the assigned PFL values must always be considered in the context of the process model from which they were derived.

For the coffee maker example, two failures were considered. First was the loss of the ability to open the outer reservoir lid, which results in an inability to open the grinder and place beans into the product. Thus, the event based on the loading of coffee beans would fail. Additionally, the failure would propagate into the next two events, load water and operate since both depend on the opening of the reservoir lid. This failure, therefore, would be rated as a **process terminal fault** if the coffee maker could no longer be used or **process transient fault** if the coffee maker could be repaired. The second failure considered was to the flow of coffee beans into the product. If coffee beans could not be added to the product, then coffee cannot be brewed. As previously discussed, the coffee maker will still try to brew coffee without coffee beans following the remaining two events as intended. Since the absence of coffee beans only results in the failure of the load beans event, the failure is rated as an **event level fault**.

3.3 Steps 6 - 8, System to Process Sensitivity (SPS)

Last, the SPS is calculated for each fault. SPS provides a percentage of the flow paths faulted in the configuration model due to single initiating failure propagated over the sum of all configurations within a desired range of an aggregated configuration model. In this way, SPS provides a gauge to the configurations that will be faulted due to a single initiating failure. For instance, if the sensitivity of the entire system to a single fault is desired, the each configuration model from all events must be compiled into a single aggregated configuration flow model. Then each of the faults identified during PFA are traced through the aggregated model. The summation of faulted configurations is used to generate a ratio of the faulted configurations to the sum of all of the configurations over the desired range. This is shown as Equation 1.

$$Sensitivity = \frac{\sum ConfigurationsFaulted}{\sum Configurations} \cdot 100\% \tag{1}$$

Consider, once more, the coffee maker example. For the faulted coffee bean flow (Figure 6), only the three configurations can fault in the combined configuration model since there are no flows, which propagate the fault into the remaining events. As previously discussed, the remaining configuration changes can still occur. If the desired range is the entire coffee maker system, then the total number of configuration changes, 31, is divided into the number of faulted configurations to determine a sensitivity of 9.7%. For the second failure concerning the opening of the reservoir lid, the fault propagates to all three of the final events in the process. This totals to 23 faulted configurations and results in a sensitivity of 74.2%. These fault calculations are shown as Equations 2 and 3, respectively.

$$CoffeeBeanFlaultSensitivity = 9.7\% = \frac{3}{31} \cdot 100\% \tag{2}$$

$$\mathrm{Res}ervoirLidFaultSensitivity = 74.2\% = \frac{23}{31} \cdot 100\% \tag{3}$$

These SPS sensitivities provide a numerical way to quantify the affect of a fault over a specific range in a configuration model, and are in no way meant to represent the severity of a failure. They are instead meant as a numerical guide to compare multiple failure modes of a specific range of configurations in a process model. The ranges are meant to be set by the designer based on the models being generated such that specific design questions may be answered. For instance, with the coffee maker example above, the range was set as the entire process of setting up the coffee maker and brewing coffee, thus a sensitivity of 9.7% implies that about ten percent of the configurations will fault with a lack of coffee beans while a sensitivity of 74.2% implies that about seventy-five percent of the configurations will fault with a lack of water stemming from a reservoir lid fault.

The use of configuration models for each event ensures that the percent sensitivities can be compared between models with different numbers of events and configurations, and because SPS provides a percentage of flows faulted to those not faulted at the configuration level, it can be used by designers to compare multiple potential configuration mappings for a product to identify alternative product interactions that are more or less sensitive to faults. Sensitivities, however, are only as good as the models from which they are derived, thus it is important for models to be developed at consistent levels of fidelity when comparisons are to be made. This is less important when a single process model is being developed, and SPS is being applied to investigate a single product design.

3.4 Discussion

Once potential faults are identified in the configuration models, they may be propagated to a functional representation of the product. Propagating the failures to a functional representation provides insight into the transformations within the product that are at risk to fail from a specific configuration fault. Having an understanding of the failed transformations internal to the product allows designers to increase robustness. For instance, in the likely event of an operator of the grind and brew coffee maker deciding to operate the coffee maker without coffee beans for sole the purpose of heating water, it would not be desirable to have internal components that fail in the absence of coffee beans. A propagation of this fault into the functional model of the grind and brew coffee maker is provided as Figure 7, and follows the same format used for PFA. The initial fault of no coffee beans is represented with a bold X, while the resultant X's are shown unshaded.

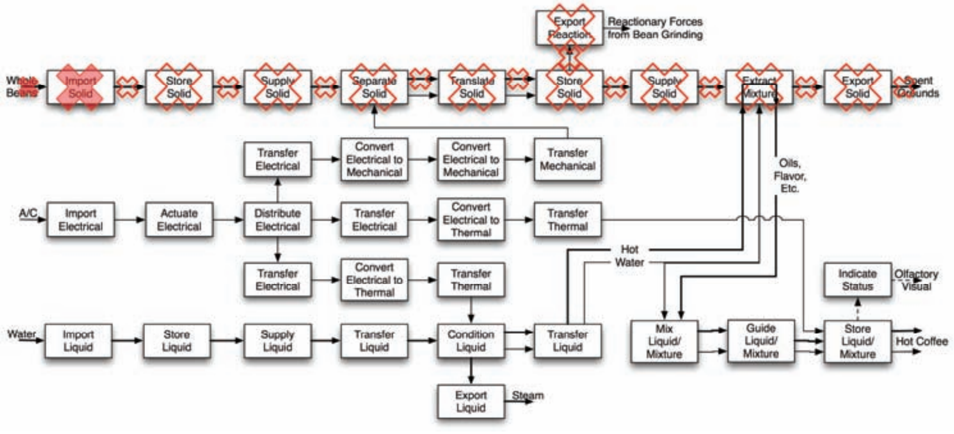

Figure 7: Faulted coffee beans flow in the grind and brew coffee maker functional model

The results of the PFL and the SPS indicate that a failure with the reservoir lid would, however, be more devastating for the product than coffee beans being excluded. Since the reservoir lid covers not only the water reservoir but also the hopper for the coffee beans, a failure with the reservoir lid would prevent the operator from importing both coffee beans (solid material) and water (liquid material) into the product. As these failures are also propagated through the functional model (Figure 8), it becomes clear that the product should be designed to protect against damage. Without fault checking built into the design, the electrical energy would still be converted into thermal energy, and without water there would be no medium to absorb the thermal energy.

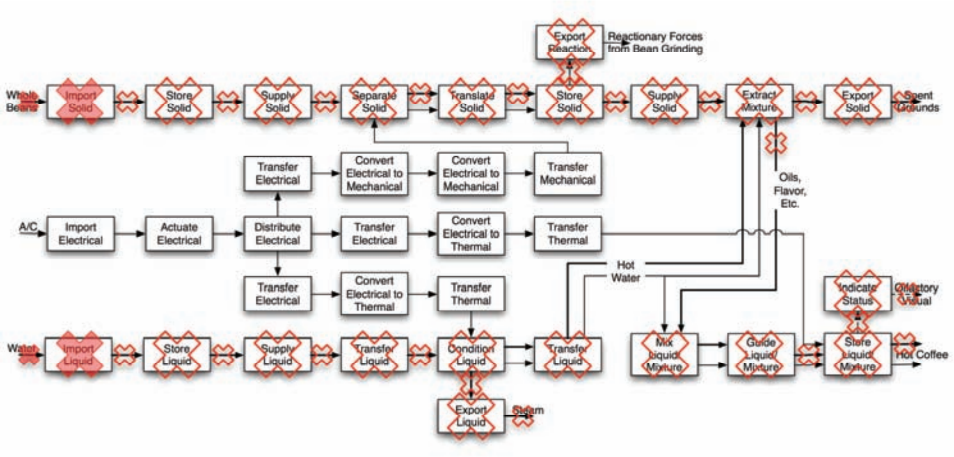

Figure 8: Propagation of faulted reservoir lid into the functional model of the grind and brew coffee maker

4. CONCLUSIONS & FUTURE WORK

Propagated fault analysis, process fault levels and system to process sensitivity provide a suite of analysis techniques to the designer that allow the investigation and analysis of faults occurring with product configurations. Faulted flows are propagated across the configurations of a product to investigate their effect. Faults can be investigated early in the design process—during conceptual design—since the analysis has no reliance on components due to the application of integrated functional and process modeling. The integration of functional modeling with process analysis compliments existing functional modeling based failure and risk tools such as function-based failure propagation [17], RED [25], FFIP [18], and FFDM [10]. For complex systems, qualitative (PFL) and quantitative (SPS) sensitivity metrics provide designers with the ability to focus design efforts on those configurations whose failures are most catastrophic to the product. Alternative product design configurations may be investigated through multiple applications of PFA, PFL and SPS to various configuration models of the same product allowing designers to weigh alternative approaches to determine which models are the least sensitive to configuration changes and faults.

Further research will investigate tighter integration with existing conceptual design methodologies. PFL will be investigated as a technique to identify event and configuration redundancy, where if a failure occurs at only the configuration or event level and the output seems to remain unchanged then redundancy may exist in the product. This analysis could be useful for verifying redundancy in critical systems or for removing redundancy in disposable systems. Also, with complex systems, the configuration models increase in complexity making manual calculation very time consuming, thus a computational tool will be developed to interface with a purpose-specific functional model drawing tool to assist a designer through the propagation of faults along flow paths following the connectivity between the configurations. The hierarchy of the functional and process models will be built into the tool to (1) provide active feedback on a fault's PFL and SPS and (2) provide linking to a system's functional model at faulted configurations so that existing function-based failure propagation and risk tools can be applied.

5. ACKNOWLEDGEMENTS

This work has been performed in conjunction with the United States Air Force Academy and has been funded by the Joint IED Defeat Organization.

REFERENCES

[1] Nagel, R.L., Stone, R.B. and McAdams, D.A. A Process Modeling Methodology for Automation of Manual and Time Dependent Processes. *ASME International Design Engineering Technical Conferences*, DETC2006-99437, Philadelphia, PA, 2006).

[2] Hutcheson, R.S., McAdams, D.A., Stone, R.B. and Tumer, I.Y. A Function-Based Methodology for Analyzing Critical Events. *ASME International Design Engineering Technical Conferences*, DETC2006-99535, Philadelphia, Pennsylvania, 2006).

[3] Kumamoto, H. and Henley, E.J. *Probabilistic Risk Assessment and Management for Engineers and Scientists*. 1996 (IEEE Press, New York).

[4] *Procedures for Performing a Failure Mode, Effects and Criticality Analysis*. (MIL-P-1629A), 1980 Department of Defense.

[5] Automotive Industry Action Group (AIAG) Potential Failure Mode and Effects Analysis (Fmea) Reference Manual. 1993 (Automotive Industry Action Group).

[6] Russomanno, D.J., Bonnell, R.D. and Bowles, J.B. Functional Reasoning in Failure Modes and Effects Analysis (Fmea) Expert System. *Reliability and Maintainability Symposium*, 1993, pp. 339-347).

[7] Kmenta, S. and Ishii, K. Advanced Fmea Using Meta Behavior Modeling for Concurrent Design of Products and Controls. *ASME Design Engineering Technical Conference*, DETC98/CIE-5702, Atlanta, GA, 1998, (ASME).

[8] Kmenta, S., Fitch, P. and Ishii, K. Advanced Failure Modes and Effects Analysis of Complex Processes. *ASME Design Engineering Technical Conference, Design for Manufacturing Conference*, DETC99/DFM-8939, Las Vegas, NV, 1999).

[9] Wilkinson, P.J. and Kelly, T.P. Functional Hazard Analysis for Highly Integrated Aerospace Systems. *Certification of Ground/Air Systems Seminar*, Ref. No. 1998/255, 1998, pp. 4/1-4/6).

[10] Stone, R.B., Tumer, I.Y. and Van Wie, M. The Function-Failure Design Method. *Journal of Mechanical Design,* 2005, 127(3), 397-407.

[11] Roberts, R., Stone, R. and Tumer, I.Y. Application of Function-Failure Similarity Method to Rotorcraft Component Design. *Submitted to Journal of Engineering Design,* 2002.

[12] Vesely, W.E. and Goldberg, F.F. *Fault Tree Handbook*. 1981 US Nuclear Regulatory Commission.

[13] Blanchard, B.S. and Fabrycky, W.J. *Systems Engineering and Analysis*. 2006 (Prentice-Hall, Upper Saddle River, NJ).

[14] Voland, G. *Engineering by Design*. 2004 (Pearson Prentice Hall, Upper Saddle River, NJ).

[15] Bedford, T. and Cooke, R. *Probabilistic Risk Analysis: Foundations and Methods*. 2001 (Cambridge University Press, Cambridge).

[16] Stamatelatos, M. *Probabilistic Risk Assessment: What Is It and Why Is It Worth It?* 2000 NASA Office of Safety and Mission Assurance.

[17] Krus, D. and Grantham Lough, K. Applying Function-Based Failure Propagation in Conceptual Design. *ASME International Design Engineering Technical Conference*, Las Vegas, NV, 2007).

[18] Kurtoglu, T. and Tumer, I.Y. A Graph-Based Framework for Early Assessment of Functional Failures in Complex Systems. *ASME International Design Engineering Technical Conference*, DETC2007-35421, Las Vegas, NV, 2007).

[19] Padalkar, S., Karsai, G., Biegl, C., Sztipanovits, J., Okuda, K. and Miyasaka, N. Real-Time Fault Diagnostics. *IEEE Expert: Intelligent Systems and Their Applications,* 1991, 6(3), 75-85.

[20] Marshall, C.W. *Applied Graph Theory*. 1971 (Wiley-Interscience, New York).

[21] Bowman, R.A. Efficient Sensitivity Analysis of Pert Network Performance Measures to Significant Changes in Activity Time Parameters. *Journal of Operational Research Society,* 2007, 58, 1354-1360.

[22] Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.H. *Engineering Design: A Systematic Approach*. 2007 (Springer Verlag).

[23] Nagel, R.L., Hutcheson, R.S., Stone, R., McAdams, D. and Donndelinger, J. Function Design Framework (Fdf): Integrated Process and Function Modeling for Complex System Design. *ASME International Design Engineering Technical Conference*, DETC2008-49369, New York, 2008, (ASME).

[24] Stone, R., Wood, K. and Crawford, R. A Heuristic Method for Identifying Modules for Product Architectures. *Design Studies,* 2000, 21(1), 5-31.

[25] Grantham Lough, K., Stone, R.B. and Tumer, I.Y. The Risk in Early Design Method (Red). *Journal of Engineering Design,* 2007, 18(1).

Contact: R.L. Nagel
School of Mechanical, Industrial and Manufacturing Engineering
204 Rogers Hall
Oregon State University
Corvallis, OR 97331-6001 USA
rlnagel@DesignEngineeringLab.org
http://www.designengineeringlab.org/rlnagel

Robert L. Nagel is a graduate student working toward his Ph.D. in Mechanical Engineering at Oregon State University in Corvallis, OR where Robert also works as a graduate research assistant in the Design Engineering Lab. Robert holds a B.S. in Mechanical Engineering from Tri-State University (known now as Trine University) in Angola, IN and an M.S. in Mechanical Engineering from the University of Missouri-Rolla (known now as Missouri University of Science and Technology) in Rolla, MO. Robert's research includes functional and process modeling and their applications to engineering design and system's analysis.