METHODOLOGY FOR THE DESIGN OF UNMANNED AIRCRAFT PRODUCT FAMILIES

Dane Freeman , Dongwook W. Lim , Elena Garcia , Dimitri N. Mavris
Aerospace Systems Design Laboratory, School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, USA

Keywords: *Product Family Design, Agent Based Simulation, Data Mining*

Abstract

To remain competitive in the marketplace, companies attempt to minimize their expenses while satisfying diverse customer requirements. If several different products are produced to service related but different market niches, then there may be implied similarities between this set or family of products. The company can then attempt to leverage those similarities, reducing waste and improving overall family performance. One major difficulty arises when designing a family of products. The sharing of components between the different products is critical to the success of the family, but is a hard combinatorial problem that has implications with all aspects of the design. This paper introduces a product family design methodology to leverage product commonalities found using data mining techniques on a generated database of potential designs. To begin to demonstrate this methodology, a modeling environment was created integrating agent-based models as well as several system models capable of analyzing a family of unmanned aerial vehicles operating under two different scenarios: aerial firefighting, and maritime surveillance.

1 Introduction and Motivation

The objective of product family design is to create a set of products that satisfy specific market niches while minimizing overall family costs[14]. Because products will be servicing sufficiently similar markets, there may be some component similarities. Ideally, sharing components between similar products should reduce the cost of the family by lessening the duplication of effort.

Using these shared components, called platforms, between different products can minimize waste by utilizing one shared component instead of two or more. This decreases engineering design effort and increases manufacturing efficiency thereby lowering family costs. Families also have the ability to consider reconfigurable systems that maintain a high level of performance by changing its components to meet multiple functional requirements or a change in operating conditions[17].

However, product families do have drawbacks. There is a tradeoff to consider when making commonality decisions is the penalty to an individual product's performance. Once a component is shared, there is less design freedom and that platform may compromise the end product. For example, if the respective products' requirements are too dissimilar, sharing components may result in unsatisfactory results for both end products. Failing to meet customer expectations either through lower end products being overdesigned and/or higher end products can result in lower market shares. Furthermore, design complexity also increases in families because formerly independent products are now coupled and require interfaces between the platforms and the rest of the product[8].

It is important to identify which components are similar enough, so that sharing does not compromise the individual products' performances. When considering a number of family design alternatives, balanced and informed decisions require a methodology to systematically trade-off the cost savings and performance losses.

Many existing product family design methods make decisions a priori about how components can be shared; constraining them across every product in the family, or not at all. Methods that simultaneously optimize component sharing and design variable settings have the potential to find better families because some subsets of products may be more alike than others[7]. However, allowing components to be shared between any subset of products results in a large combinatorial problem, and thus considering large product families can be computationally prohibitive.

Figure 1, provides a holistic view of the different levels in product family design. The figure illustrates the different domains in product family design as well as the combinatorial mappings between the different domains[5]. For example the product portfolio is the mapping between customer needs and the individual functional requirements. The product portfolio describes the numbers of products and which product and set of functional requirements are necessary for each market niche. The mapping between the design parameters and functional requirements is the platform specification. Here different components are either unique to a particular product or are shared among one or more products. There are lower levels of mappings and finer details as the product family design process progresses into process and logistic design however for the purposes of this paper only the product definition and prod-

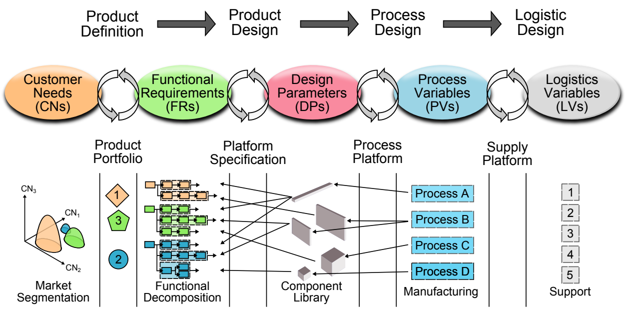

Fig. 1 A Holistic View of Product Family Design and Development[5]

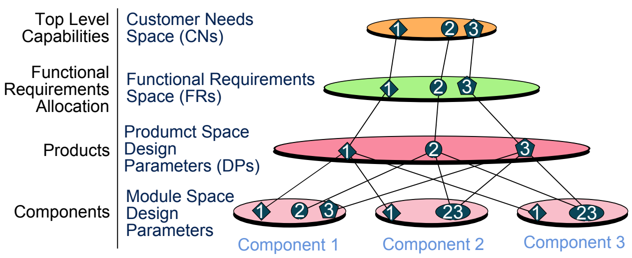

Fig. 2 Product Family Hierarchy

uct design are considered.

To help guide the formulation of a new methodology, we will begin by investigating part of the breakdown of the different aspects in product family design. Figure 2 presents hierarchically the domains in product family design. The top space of this hierarchy depicts the customer need niches that products must target to be successful. Below the customer needs space is the functional requirements space that each product must meet. The third layer is the product space where each product is defined by physical design parameters. The lowest level is the module spaces. Here each module space encompasses their own subset of design parameters relevant from the product design parameters. Like in the holistic view, the mapping between the customer needs space and the functional requirements space is the product portfolio; mapping between the functional space and the product space is the product architecture; mapping between the module subspaces and the product space is the platform sharing configuration.

This paper proposes a method that can identify component commonalities which will attempt to lessen the combinatorial problems by identifying possible sets of product family platforms enabling a more efficient exploration of different product families. This is done by inspecting the domain subspaces for natural groupings. These domains exist across the entire scope of product family design, but this method focuses on module subspaces. If components from different products are similar enough to be grouped together in these subspaces, then those components could possibly become the same platform, because the performance tradeoff should be smaller than grouping dissimilar components. Using these natural groupings should lessen the combinatorial problem in the family design allowing for more alternatives to be considered. The methodology needs to balance cost savings and performance compromises due to component commonality and should address product growth potential, technology evolution, and uncertainty.

To aid in the demonstration of the methodology a modeling environment needs to be created that captures the complexities of the scenarios the family will operate. Furthermore, this environment will need to contain enough details to allow the different component subspaces to be explored and to quantify their impact through to the customer needs level. A family is being considered that fulfills two distinct scenarios: aerial firefighting, and maritime monitoring. Because the different members of the family will be interacting with each other, a system of system (SoS) model will need to be implemented. The SoS nature of these scenarios also further complicates the family design process.

2 Family Design Methodology

The proposed methodology attempts to address the design of a product family from the beginning problem definition stage though to the establishment of all of the physical design parameters of both the unique components and the various product platforms. The complete methodology is illustrated in figure 3. This proposed methodology identifies the trade-off between product performances and costs by identifying family commonalities in a robust and systematic way. It does this using pattern recognition in the context of the larger product family design process.

This methodology begins similarly to the general outline of axiometric design[19]. The first steps relate to the definition of customer needs and requirements and to the generation of alternative architectures. Using this information, detailed models can be created that capture the design parameter's impacts on high level customer requirements. Those models can then be explored to generate a database of potential designs that are subsequently filtered at different levels using various constraints until only the best subset remains. Product platform decisions can then be guided by identifying similarities between the different products in the component subspaces. A final product family can be identified using the down selected platform options.

The first step is to identify the set of high level capabilities the customer desires. The goal is to understand the problems that need to be solved. This step can be done by background research as well as other sources like detailed market analysis and customer interactions. There have been several studies for performing market segmentation at this customer needs level[20, 6, 2]. Ultimately, the designer should be able to describe the targeted market niches as well as translate the customers' needs into Measures of Effectiveness (MoEs). In this step environmentally driven noise parameters are also identified which can drive the products performances away from the desired customer

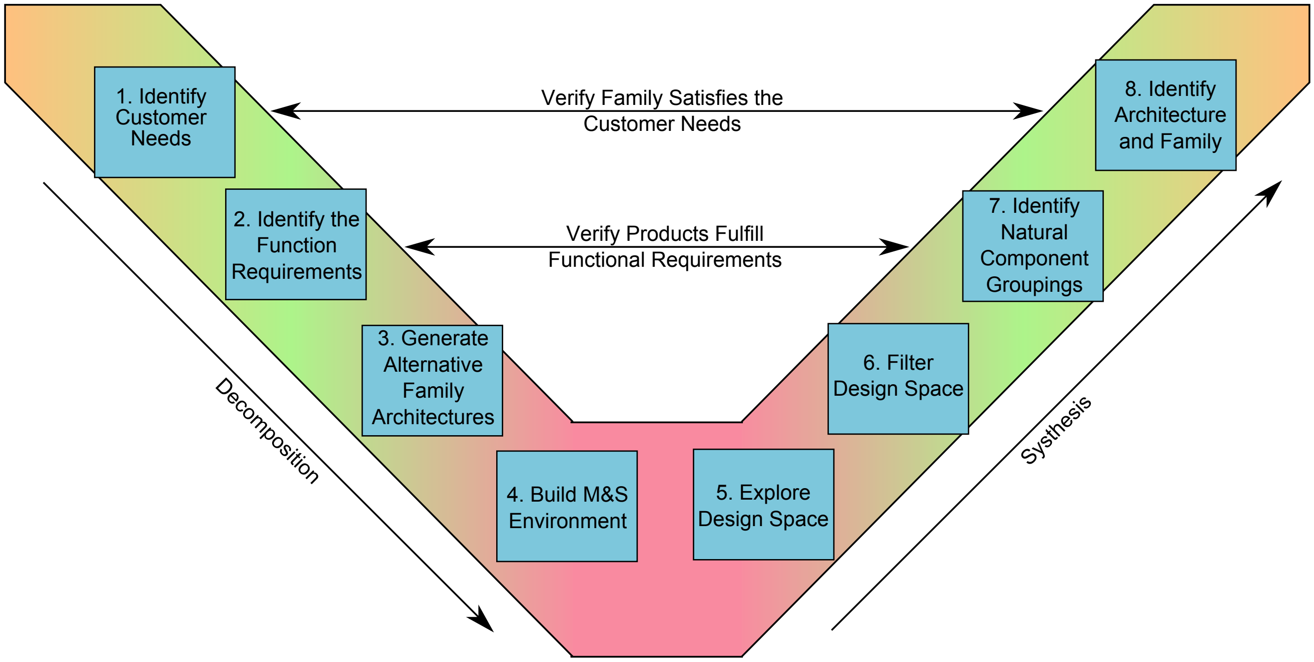

Fig. 3 Product Family Design Proposed Methodology

requirements.

Next these customer needs must to be translated into engineering requirements. After the customer needs are identified and codified, a functional breakdown is used to determine the functions necessary to satisfy the customer requirements. A functional breakdown is performed by analyzing similar existing systems and by applying engineering intuition and experience to the problem. Stone et al. [18] introduces three heuristics to create a systematic module identification approach 1) a function flow may pass through a product unchanged, 2) a flow may branch, forming independent function changes, or 3) a function flow may be converted to a different form. According to their study, the choice of which module to implement is not obvious and these rules help to keep modules easily identifiable with a particular function.

Once the functional breakdown is completed different family architectures can start to be developed. The family architecture relates the different functions back to physical systems and is concerned with modularity and the commonality between the different products[5]. For complex systems there is no unique mapping between functions and systems. In addition to creating the function to system mappings, individual product's modules should be identified. A module is a subsystem that can be treated as an individual product block and will be used later when looking for similarities between the different products in the family[15, 16].

Then, after alternative architectures are generated, a modeling and simulation environment is built to evaluate how well the different architectures meet the customer needs. When creating the models of the system, care should be taken to address appropriate levels of fidelity. Typically, as physics codes increase in fidelity, their computation time also increases greatly requiring the use of surrogate models. Surrogate models, like neural networks or response surface equations, are tuned to the underlying physics based models, capturing the relevant variability and can be executed in a fraction of the time. This speedup allows for a more densely sampled database for the data mining. Dense sampling is important to ensure that the database adequately represents the product spaces. The end result of this step is to have a modeling and simulation environment that is able to capture the impacts of product components on high level customer MoEs.

After the modeling environment is created, the database of each product is populated either using a design of experiments or a random sampling of the design space. When exploring the design space of each product, there is no assumed platform configuration, i.e. the products are evaluated independently without component sharing.

This exploration step is critical because it generates the database of product design alternatives. If the database proves to be too sparse and fails to capture feasible points then additional sampling schemes must be employed. To improve the quality of each product's design alternatives, local optimizers could be used. However, it is important to have an even exploration of the design space, so that any groupings found later are due to the physics of the problem rather than biases in the optimizer.

The database generating process up to this point can be vizualized returning to the hierarchical product family domains with the dots representing different possible designs, figure 4. On the figure, the lowest domain is the individual

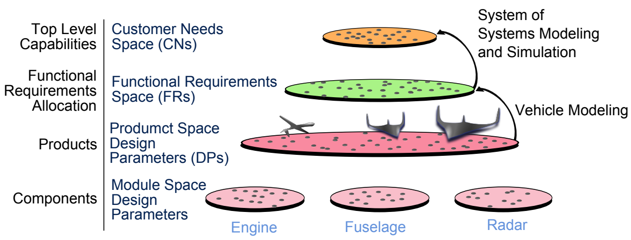

Fig. 4 Product Family Domains Diagram

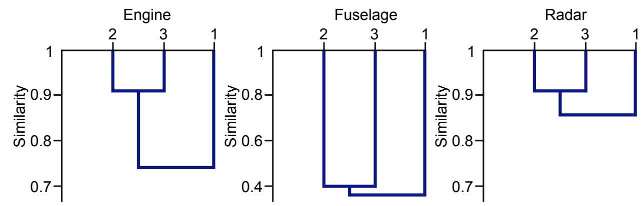

Fig. 5 Component Commonality Dendrogram

module spaces for each component in all of the products. For example, only the engine, fuselage, and radar is shown but in reality there can be very many components considered. The next higher domain is the product space where each product is composed of all of their shared components as well as any unique components. Products are mapped up to the next level into the functional requirements space using various performance models like vehiclle models or sensor models. Examples of performances that belong to the functional requirements space would be each vehicle's speed, range, and payload capacity. Finally, the highest domain is the customer needs space. This requires another level of modeling to be able to calculate how well different sets of products are able to accomplish their respective requirements.

Once the database has been populated it can be post processed with different constraints to narrow the design range. These requirements can represent customer needs, feasibility constraints, viability constraints, and/or technical constraints. Products that fail to meet their required customer needs are pruned and not considered in the pool of points to be clustered. The database is also biased to the physics of the problem as well as the feasibility constraints.

Once the database is filtered, the module spaces can be inspected to look for natural groupings in the design parameters. The goal is to be able to, for each module space, identify if any products are similar to one another which would indicate potential family commonalities.

Identifying possible sets of product component commonalities for the feasible designs is done on the separate subspaces. Each subspace represents an individual component extracted from the database of generated design points for the whole family. If components from different products are similar enough to be grouped together, then those components could possibly become a single component.

There are several possible data mining techniques that could be employed to extract the component similarities. Freeman[4] demonstrated using fuzzy equivalence relations extracted from fuzzy c-means cluster membership functions to show the binary relationship between one product's component to a different product's component. The study shows pattern recognition analysis was able to identify similarities between the different components when compared to a alternative product platforming method. Fuzzy equivalence relations can be used to generate a heuristic for further product family investigation.

These relations can be visualized in a dendrogram, figure 5. As the decision parameter increases only products with a strong relationship continue to be grouped together. At the extreme low, products all share the same platform and at the high end are no sharing occurs. For example, in the engine module space engine aircraft 2 and aircraft 3 are more similar to each other than to aircraft 1. Likewise, for the radar and fuselage aircraft components but the difference is smaller between the fuselages.

Now the heuristic can then be used to guide exploration in determining the component partitioning that yields the ideal cost savings as well and how much performance penalties there are in the products. This final step explores the family and architectural design space using different component sharing suggested by the preceding pattern recognition step. The end goal is to identify which architecture and family should be carried forward into more detailed design and analysis. Because the commonality decisions are guided by naturally grouped points, there should be less wasted computational effort on poor areas in the design space allowing better searching in the preferred regions. For example from the notional dendrogram above, sharing and engine, fuselage, or radar between aircraft one and aircraft two and three is not recommended. Again this is based off of clustering those module spaces and finding that product one was more different than product two and three.

There is also an iterative procedure that can occur between steps 6 and 7. Because the database has already been generated in step 5 it is simple for the designer to interact with the data and re-filter based on different constraints. When different design constraints are used, different sets of design alternatives are left in each module space. Using data mining on the spaces can yield alternative groupings which then can be compared with patterns found from the other criteria. This captures the component commonality sensitivities. For example, if two products continue to have similar engines even as different constraints are placed on the products, then it strengthens the support that those two products should share a common engine.

It is critical to verify these suggested commonalities by evaluating the precise tradeoff between cost and performance presented by sharing. The final objective of this is to arrive at a complete product family portfolio that maximizes the amount of cost savings while minimizing the performance penalties. Ultimately, this tradeoff is what decision makers will use to select the family.

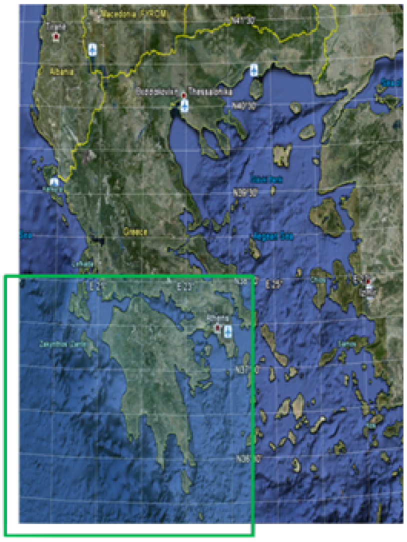

Fig. 6 Area of Greece under Consideration

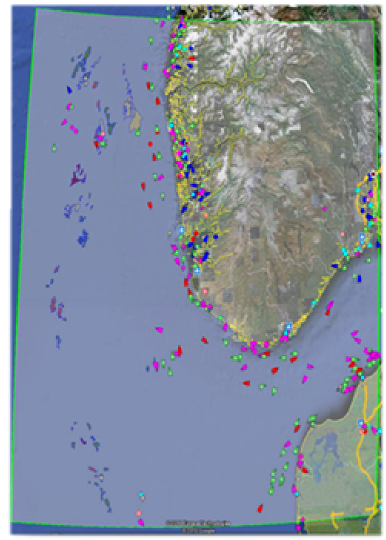

Fig. 7 Norway Coast

3 System of Systems Mission Scenarios

The remainder of this paper focuses on the implementation of the modeling and simulation environment able to evaluate the performance of a product family of aerial vehicles for two different scenarios. These scenarios are different enough to require distinct product solutions, but are similar enough to allow family considerations. The first scenario explored the possibility of using a system of UAVs for firefighting in the remote Greek islands. The second scenario explored the use of a family of UAVs for maritime surveillance and monitoring off the coast of Norway.

The firefighting scenario was developed around the Greek islands because the islands off the coast of Greece are small, and dispersed. So housing local fire units would be cost prohibitive. This combined with hot summers, dry brush, lighting strikes, and human negligence, causes frequent fires posing a threat to the natural beauty and infrastructure of Greece. To prevent this damage, the focus of the scenario is to monitor the region with real time surveillance of the forest regions with high altitude aircraft; and, when the fires do appear, provide continuous coverage of the fire location and fire extinguishing support with fire retardants. Figure 6 shows the portions of the Greek islands used to develop the modeling and simulation environment.

The second scenario is the patrolling of the Norwegian maritime Exclusive Economic Zone (EEZ). Maritime traffic is very dense in the North Sea, especially in the south of Norway near the strait leading to the Baltic Sea. Also, several Norwegian offshore oil and natural gas platforms lie in the same area, around 70 nm off the coast. These platforms require persistent surveillance by the coast guard to monitor the release of oil into the sea as well as protect the rigs themselves. Large accidental discharge is unlikely however smaller intentional discharge due to shipping cleaning out tanks or oily bilge is much more frequent[3]. Many European countries have agreements for monitoring, dealing, and prosecuting illegal oil discharge at sea[1].

In addition to environmental monitoring and response, this scenario requires vehicles suitable for Search and Rescue (SAR) of distressed ships. Accidents happen at sea and it is the responsibility of the coast guard to offer aid. Having a persistent aerial surveillance allows the tracking of polluters such as tankers cleaning out their tanks, monitoring of trawlers to ensure they stay in the fishing areas, and possibly preventing terrorist attacks on the platforms. Maintaining the safety and health of this region requires a mechanism to be in place which is capable of monitoring the illegal discharge of oil slicks, and the required dispersion and clean-up of these slicks; and the searching for any distressed vessels in need of rescuing.

Figure 7 shows a real time evaluation of the vessels off the coast of Norway collected using the Automatic Identification System, AIS[10]. The AIS monitors different types of ships, represented by different colors) and reports their position. As can be seen, the traffic can be high in specific areas near the coast. These regions can be home to many different services that require ships to dock and make this a high traffic region of transient ships; many without much interest vested in the local environment. Whenever there is high traffic without much private interest, coupled with a small likelihood of being caught, there will be violations of local laws.

Both of these problems exhibit long endurance monitoring and surveillance missions which are better conducted at higher altitudes with improved fuel efficiencies, and also low altitude time critical delivery missions. Because of the long endurance times required and high mission risk in the case of the firefighting unmanned aircraft are ideal[9]. Also because there is not a current existing of unmanned aircraft, the ideal component sharing between the different aircraft is unknown making these scenarios an interesting design opportunity.

3.1 Step 1 Customer Needs

The customer needs for the example problems can be identified through background literature review and the description of the scenarios. In the firefighting scenario, the customer needs to prevent damage cause by the fire. To that end, the high level MoEs identified are: the time it takes to detect fire, the time to respond to the fire, the size of the burnt land, and the cost of the associated systems.

For the maritime monitoring scenario, the customer needs are to limit ecological damage from maritime oil discharge as well as prevent loss of life from accidents at sea. These needs are translated into the following MoEs: time to identify oil spills, time to clean spills, successful identification of the polluter, average time ships go unmonitored by any aircraft, time to search for any distressed boats, time to respond to any distressed boats, and the costs of the program. In addition to these MoEs for both scenarios, they also have noise factors including the geographic locations of the oil spill, fire, or distressed boats. These noise factors a important for identifying the robustness of any resulting aircraft family.

3.2 Step 2 Functional Breakdown

For the scenarios' functional breakdown, various necessary missions were identified. These missions would need to be completed for the family to successfully meet the customer needs.

Two missions were identified for the firefighting scenario: fire monitoring, and firefighting. The fire monitoring mission is to provide coverage of the fire prone land. Once a fire is detected a firefighting mission is performed to drop fire retardants.

The maritime monitoring scenario has six identified missions: patrol (fisheries, oil platforms), ship tracking, environmental monitoring, oil spill response, and search and rescue. The fishery patrol and ship tracking missions help to provide awareness to the Norwegian Coast Guard in its task of maintaining security and enforcing the Norwegian EEZ. Environmental monitoring is critical also to the protection of the Norwegian coastline from maritime polluters who dump bilge and wash oil tanks in the open sea. The primary goal of this mission is to detect and verify oil slicks while collecting sufficient evidence for legal action. Once a spill is detected, and if it is of appropriately large size, the system moves to an oil spill response mission capable of delivering tools to quickly combat the spill; like dropping oil dispersant before the slick can interfere with wildlife. Occasionally, there are ships that require assistance due to the harsh weather. At these times, the system needs to be able to search for and locate distressed vessels / people and monitor the situation until additional assistance arrives. Once finding the distressed vessels / people, a rescue mission is performed to deliver emergency aid.

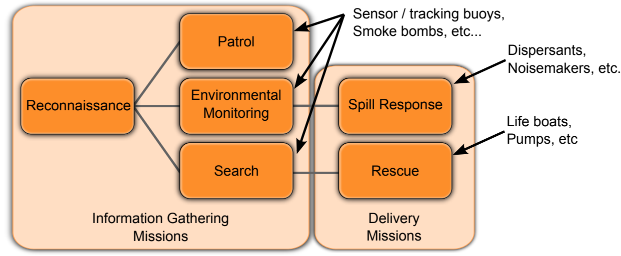

Fig. 8 Mission Hierarchy

These missions cause induced functions for their successful completion. For example, in the firefighting scenario, fires must be able to be sensed with enough accuracy to successfully delivery fire retardants in a location that maximizes their effectiveness. Induced functions of for the maritime monitoring missions include, the aircraft being be able to sense the locations of ships, oil spills, and distressed boats.

3.3 Step 3 Family Architectures

The next step is to use the functional requirements to formulate possible solutions. To simplify the kinds of subsystems involved both a radar and infrared sensor will be used to satisfy the induced functions of the aircraft being able to sense and track their environment. The kinds of aircraft will also be limited to traditional tube and wing configurations to allow for better benchmarking against existing alternatives.

The firefighting scenario only has two missions, reconnaissance and delivery of a large payload of fire retardant. There are two possible main solutions for performing these missions: having one aircraft searching and dropping, or having unique aircraft types for each role.

Because the maritime monitoring scenario contains more functional requirements the missions are first categorized into two types: information gathering, and delivery. The ship tracking mission can be treated as a superset of the patrol, environmental monitoring, and search missions. Information gathering missions will trigger the delivery missions of spill response and rescue once their respective targets are found.

After inspecting the mission hierarchy, different high level family architectures are developed. Figure 9 shows the first architecture being considered. Here, there is one aircraft that performs all of the information gathering missions and a different aircraft that is called to perform the special delivery missions. Simple mission profiles are included for each of the different missions and are used to help develop mission performance models.

The second architecture that could be considered is shown in Figure 10. Here, the information gathering missions are combined with their unique delivery segments. For example, for the search and rescue missions, one aircraft performs the searching and then is capable of drop-

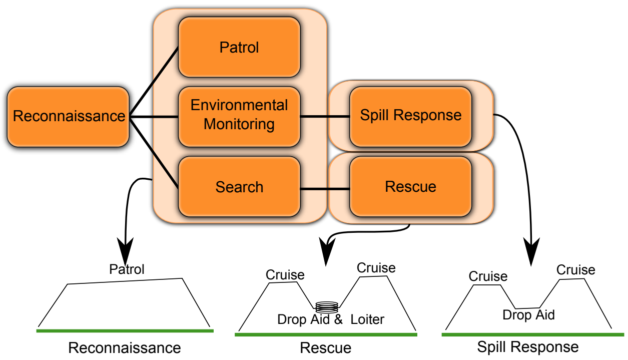

Fig. 9 Maritime Architecture One

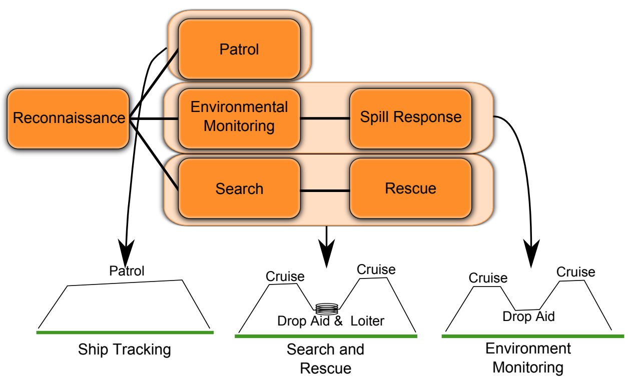

Fig. 10 Maritime Architecture Two

ping aid to the distressed vessels / people.

3.4 Step 4 Modeling and Simulation

These scenarios requires complex modeling environments to be created capable of generating the required design database which captures everything from the component level to the customer needs level. Two environments are created coupling with their respective system level models to understand the impact of the various components at higher levels. For the two simulations, it is optimal to maximize the code reused between the two environments while still changing the scenario specific elements of the environment.

To be able to capture the customer needs level a System of Systems (SoS) model needs to be created. There are several methods to model and analyze SoS each with their own strengths and weaknesses. The most common methods for exploring these complex design spaces are discrete event simulations, and agent based models (ABMs). Agent-based models were found to be suitable for these two problems to more flexibly capture interactions between the vehicles and the scenarios' environments. Below, figure 11, is an image of the modeling and simulation environment for the firefighting scenario.

To create the full integrated modeling and simulation environments, several sub-models are required. For example, sensor models and an aircraft sizing model which will

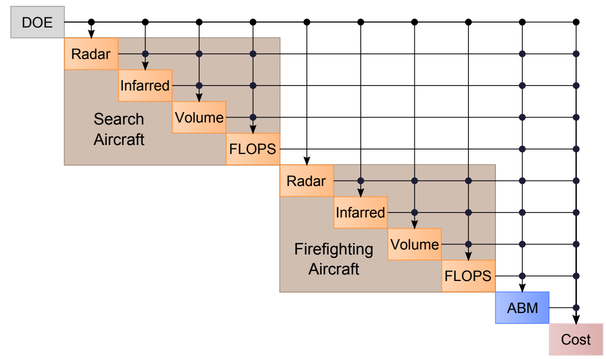

Fig. 11 Integrated Modeling and Simulation Environment for the Firefighting Scenario

be input into the ABM. The sizing and synthesis of aircraft for this integrated environment, based off of a notional medium altitude long endurance (MALE) aircraft, requires several various code disciplines.

The system level models yield vehicle level Measures of Performance (MoPs) and the agents used in the SoS models create a mapping from the MoPs to overall mission-level Measures of Effectiveness (MoEs). The modeling environment is used to populate a large database full of potential product designs that satisfy all of their requirements and then use that database to find opportunities for commonality. The base system level models were created using first principles for the various mission sensors (radar, infrared sensors) as well as simple cost estimating relationships to understand how expensive certain components are.

The appropriate initial inputs are selected from the design of experiments, which is combined with each aircraft, and sized independently (depending on the needs of the payload volume and the sensor package). Following the sizing, the SoS simulation is implemented and metrics are stored.

For these two examples radar and infrared sensor models are developed from first principles and implemented in Matlab. In the maritime problem radar can be used to detect ships at sea, while for the Greek problem radar can be used to detect fires by the return from smoke. For the radar modeling, the basic radar equation is implemented which is based off of models from Mahafza[12]. Infrared sensors are needed to be able to identify polluters in the Norwegian simulation and to provide additional information about the fire in the Greek simulation. The infrared sensor model is from Lomheim[11].

The volume sizing of the aircraft is performed in Matlab and captures the dependency of the geometry and payload volume to ensure that the firefighting aircraft and oil spill response/rescue aircraft are large enough for their payload. With this volume sizing approach the internal placement of each of the components is ignored. Since this is a conceptual design study it is important to find the subclass of vehicles that are capable of completing the mission before specific locations are considered. Using the assump-

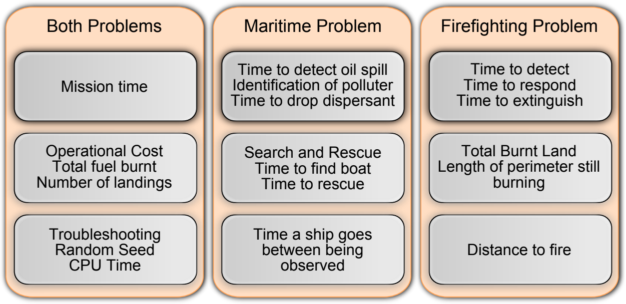

Fig. 12 Tracked Measures of Effectiveness for Both Family Simulations

tions about the traditional aircraft configuration: the aircraft has a cylindrical fuselage, and the avionics and other internal components will not increase or decrease in volume from a baseline aircraft. These imply that the changes to the fuselage length directly impact the size of the payload the aircraft is capable of carrying. The inputs for this sizing method are the payload volume, the percentage of the payload carried internally, and the notional MALE aircraft geometry. The outputs are the fuselage length and the payload water weight.

The aircraft sizing is then conducted using a code known as FLOPS[13] and uses all of the relevant coupling parameters from the sensor models and volume sizing. The sizing of the aircraft is conducted as a rubber aircraft from a notional MALE model. This aircraft has an endurance which is superior to 24 hours of loitering and a payload of up to 3,000 lbs. On the larger side, this aircraft is scaled to be similar to the CL-215 with a 12,000 lb payload.

FLOPS models were created for the four separate configurations: two for the Greek problem, and two for the Norwegian problem. Each test problem has a patrol aircraft and a drop aircraft with slightly different missions. All of the aircraft models use a turboprop engine and are based on a calibrated rubberized engine model for a notional MALE aircraft. The FLOPS model is used to calculate optimal mission conditions, components and gross weights.

After all the sub models are executed the SoS ABM can use the metrics of performance to conduct an investigation of the vehicles' effectiveness of completing the mission. In order to accomplish measuring the effectiveness many, but not all, of the outputs from the system level simulations are required for input into the SoS level simulation. These inputs are then combined with simulation specific inputs.

There are several metrics which require investigation, figure 12. As can be seen in the list there are many similar metrics for both of the simulations.

The execution of the SoS agent based models are similar for both scenarios. First, the simulation is initialized at a steady state and an allocation of the tasks is conducted. An example of this allocation in the Greek simulation is after a fire is detected, the simulation identifies retardant delivery is needed; however, within the simulation only a subset of the aircraft are capable of completing this mission, and thus one of the drop aircraft is selected to complete the mission in lieu of the searching aircraft. Upon selecting the aircraft, a check of the aircraft's ability to successfully complete the mission is also measured. This check determines whether or not the aircraft has enough fuel to complete the mission. If it does not, a different aircraft is selected.

After a proper asset has been selected, the mission is conducted and the environment updated for the new conditions. This process repeats itself until an exit criteria has been reached. The only difference between the two simulations in terms of resource allocation is in the maritime problem, a search and rescue mission may take precedence over the oil slick identification and dispersant missions. The reason for this is obvious: the possibility of saving lives is significantly more important.

The specific patrol algorithm for these simulations has been developed by ASDL to search the area in a systematic matter. For this search function each pixel is set to a cost function for each of the searching aircraft. The cost function indicates the last time the pixel was seen, the closeness of the pixel to other aircraft and the closeness of the pixel to the specified aircraft. This cost function pushes searching aircraft to explore areas which have not been seen as well as prevents aircraft from pursuing identical search regions.

This environment combines the elements necessary to generate the product database which will be populated in the future by evaluating the effectiveness of the aircraft family in performing these complex system of systems missions.

4 Summary

Companies competing across a variety of market niches should attempt to identify any product similarities. The company can then attempt to leverage those similarities to streamline design, manufacturing, and maintenance of the overall family. Care must be taken to not employ unnecessary sharing to prevent large decreases in performances or a large increases in overall system complexity.

There is a gap from a methodology standpoint in addressing the complex and combinatorial nature of family commonality coupled with family design in system of system scenarios. This paper introduced a product family design methodology addressing the platform combinatorial problem by using pattern recognition on a database that has been biased to the physics of the problem. It is believed that this methodology will improve over existing methods because 1) pattern recognition could be used to help identify those points that are more alike guiding platform selection, 2) sensitivity to changing requirements can be captured by comparing the clustering results from different design databases that have been filtered with different requirements, and 3) knowing potential family platforms can enable better product family exploration because poor platform combinations can be excluded.

To begin demonstrating the methodology on a real world test case, a modeling environment was first created that captures the complexity of the scenarios involved with sufficient detail to be useful in determining the impact different components have on the overall systems.

Research progress to date has been to understand the general product family design process, as well as details into both test scenarios so that the proposed product family design methodology is tested on problems that reflect reality. These models were created to map the component level physical characteristics into progressively higher performance spaces. The sensor models are based on first principles, the aircraft model was implemented using the validated aircraft modeling code of FLOPS for overall vehicle performance, and overarching agent based simulation to determine mission performances were developed. The fully integrated simulation environment enables the impact of low level component settings to be mapped into how well customer needs are met.

The next steps are to explore the integrated model design space for the subsystems. After a validation is completed and a baseline point analyzed, the next steps are to apply pattern recognition techniques on the database of the different component subspaces to identify similarities that could eventually become a single common component. It is believed that by reducing the combinatorial problem due to the identification of these commonalities, potential quality product families can be found more efficiently than had other product family design methods been used.[3]

5 Acknowledgments

This work has been supported by Saab and Dassault Aviation. Dane Freeman is a Saab fellow through the Aerospace Systems Design Laboratory at the Georgia Institute of Technology. Special thanks to Gunnar Holmberg, Michel Ravachol, Phillipe Ranque, Kemp Kernstine, and Remi Coisnon. Their support is gratefully acknowledged.

6 Contact Author Email Address

For further information contact: dane.freeman@asdl.ae.gatech.edu

References

[1] *Bonn Agreement Aerial Surveillance Handbook 2004*, 25 edition, 2007.

[2] Agard B and Kusiak A. Standardization of components, products and processess with data mining. *Proc International Conference on Production Research Americas*, Santiago, Chile, 2004.

[3] Ferguson S. *Aerial Obervation of Oil Pollution at Sea*. Cedre, english translation edition, 2006. The Centre of Documentation, Research and Experimentation on Accidental Water Pollution.

[4] Freeman D, Lim D, Garcia E, and Mavris D. Identification of product family platforms using pattern recognition. *Proc ASME International Design Engineering Technical Conferences - Design Automation Conferences*, Washington DC, 2011.

[5] Jiao J, Simpson T. W, and Siddique Z. Product family design and platform-based product development: a state-of-the-art review. *Journal of Intellegent Manufacturing*, 2007.

[6] Kazemzadeh R, Behzadian M, Aghdasi M, and Albadvi A. Integration of marketing research techniques into house of quality and product family design. *International Journal of Advanced Manufacturing Technology*, Vol. 41, pp 1019–1033, 2009.

[7] Khajavirad A, Michalek J. J, and Simpson T. W. An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. *Structural & Multidisciplinary Optimization*, Vol. 39, No 2, pp 187–201, 2009.

[8] Krishnan V and Grupta S. Appropriateness and impact of platform based product development. *Management Science*, Vol. 47, No 1, pp 52–68, 2001.

[9] Kumar M and Cohen K. Wild land fire fighting using multiple uninhabited aerial vehicles. *Proc AIAA Infotech@Aerospace Conference*, April 2009.

[10] Lekkas D. Marine traffic, http://www.marinetraffic.com/ais, Retrieved Last (May 2012). Hosted by University of the Aegean.

[11] Lomheim T, Milne E, Kwok J, and Tsuda A. Performance sizing relationships for a short-wave mid-wave infrared scanning point-source detection space sensor. *Proc Aerospace Conference*, pp 113–138, 1999.

[12] Mahafza B. R and Elsherbeni A. *Matlab Simulations for Radar Systems Design*. Chapman & Hall/CRC, 2004.

[13] McCullers L. A. *FLOPS Flight Optimization System*. ATK Space Division, Mail Stio 442 NASA Langley Research Center, 8.11 edition, October 2009.

[14] Meyer M. H and Lehnerd A. P. *The power of product platform - building value and cost leadship*. New York: Free Press, 1997.

[15] Miller T. D and Elgaard P. Structuring principles for the designer. *International Design Seminar: Integration of Process Knowledge into Design Support Systems*, 1999.

[16] Newcomb P. J, Bras B, and Rosen D. W. Implications of modularity on product design for the life cycle. *Journal of Mechanical Design*, Vol. 120, pp 483–490, 1998.

[17] Olewnik A, Brauen T, Ferguson S, and Lewis K. A framework for flexible systems and its implementation in multiattribute decision making. *ASME Journal of Mechanical Design*, Vol. 126, pp 412–419, 2004.

[18] Stone R. B, Wood K. L, and Crawford R. H. A heuristic method for identifying modules for product architectures. *Design Studies*, Vol. 21(5), pp 5–31, 2000.

[19] Suh N. P. *Axiomatic Design: Advances and Applications*. Oxford University Press, 2001.

[20] Zhang Y, Jiao J, and Ma Y. Market segmentation for product family positioning based on fuzzy clustering. *Journal of Engineering Design*, Vol. 18, No 3, pp 227–241, 2007.

Copyright Statement