ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

New Orleans JAZZED about Engineering Education

Paper ID #16145

Modifying Model-based Systems Engineering for Undergraduate Students

Dr. Ashley Bernal, Rose-Hulman Institute of Technology

Ashley Bernal is an Assistant Professor of Mechanical Engineering at Rose-Hulman Institute of Technology. She received her PhD from Georgia Institute of Technology in 2011. She was an American Society of Mechanical Engineers (ASME) teaching fellow and Student Teaching Enhancement Partnership (STEP) Fellow. Prior to receiving her PhD, she worked as a subsystems engineer at Boeing on the Joint Unmanned Combat Air Systems (JUCAS) program. Her research areas of interest include piezoelectrics, nanomanufacturing, optical measuring techniques, and intercultural design.

Dr. Scott Kirkpatrick, Rose-Hulman Institute of Technology

Scott Kirkpatrick is an Assistant Professor of Physics and Optical Engineering at Rose-Hulman Institute of Technology. He teaches physics, semiconductor processes, and micro electrical and mechanical systems (MEMS). His research interests include heat engines, magnetron sputtering, and nanomaterial self-assembly. His masters thesis work at the University of Nebraska Lincoln focused on reactive sputtering process control. His doctoral dissertation at the University of Nebraska Lincoln investigated High Power Impulse Magnetron Sputtering.

Dr. Anneliese Watt, Rose-Hulman Institute of Technology

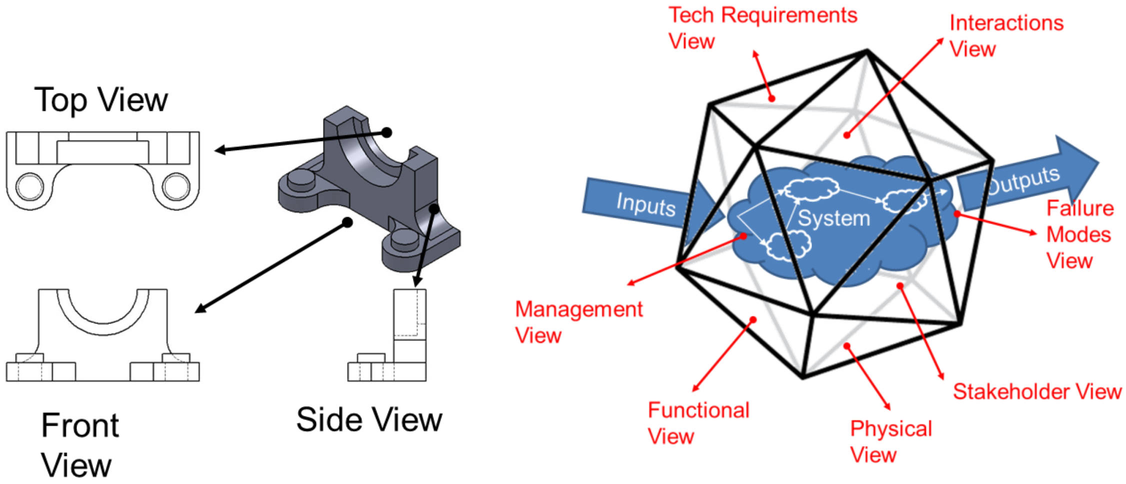
Anneliese Watt is a professor of English at Rose-Hulman Institute of Technology. She teaches and researches technical and professional communication, rhetoric and composition, medicine in literature, and other humanities elective courses for engineering and science students. Her graduate work in rhetoric and literature was completed at Penn State, and her recent research often focuses on engineering and workplace communication.

Modifying Model-based System Engineering for Undergraduate Students

Abstract
In this complex global society, developing products that fit stakeholder needs has become extremely challenging. In order to prepare undergraduates to deal with complexity, especially in a global context, systems engineering approaches have been taught for the past 3 years in a Summer Grand Challenge Program in which the students design, build, test, and communicate a humanitarian design for a developing country. We have boiled down the entire system's approach into four essential models: a stakeholder and feature model, an interactions model (black-box), and logical architecture (white-box). To create each of these models, every component is given a set of grammatical rules to help establish a broader definition of the requirement of the subcomponent. This paper will focus on how we have adapted the traditional model-based system engineering content to make it more easily accessible and understandable by undergraduate students.

Introduction
A faculty team (professors of professional communication, mechanical engineering, and physics) at Rose-Hulman Institute of Technology have collaboratively designed and taught an intensive multidisciplinary design program [1] in which undergraduate engineering and science students tackled one of the National Academy of Engineering's Grand Challenges[2] during a 12-credit-hour 10-week summer program. The program is centered around designing a system to use in a less developed country with major components of systems engineering integrated throughout the experience in the form of practice of model-based systems competencies.[3-4] For instance, students were required to identify stakeholders and analyze their needs via the development of feature models, to generate system domain models and feature definitions and attributes, and to develop system logical and physical architectures. Over the past three years of teaching this material, we have made changes in the way that we have presented this material which have yielded much improved models and better designs. In this paper we will discuss what changes we have made in terms of what models we have the students develop as well as grammatical rules that help guide the students through the model development process.

Background
The goal of creating systems models is to help students make sure all perspectives of the system are articulated in a clear and consistent manner. This is similar to creating CAD models. In a CAD drawing there are 2D orthographic views. Three orthographic views are required to completely capture all of the features. These 2D views should also be consistent with one another. Developing a system is much more complex and nebulous than a 3D CAD drawing (three dimensional computer-aided design drawing), and more views are required to fully capture all of the views (Figure 1). The different views of a system are known as models and each model has a specific name associated with it. The objective of developing these models is for students to develop system competencies that students can utilize well beyond one single design problem or class.



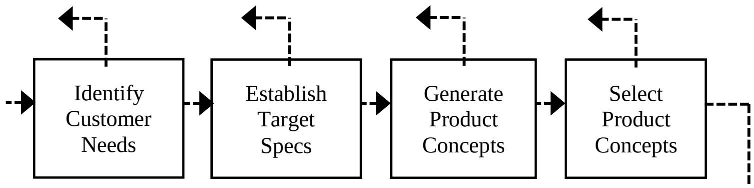
(a) (b)

Figure 1- (a) CAD model demonstrating the 2D views that fully represent the design shown in the isometric view (b) Image demonstrating the numerous views that are required to fully describe an entire system.[5]

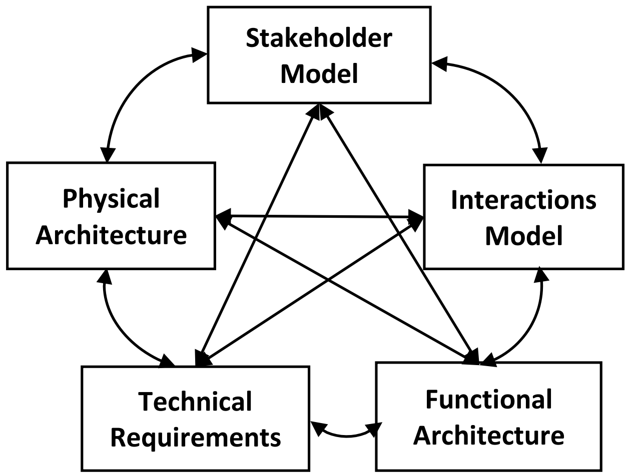
The system competencies defined by Schindel et al [3] include the following:
1. Describing the target of innovation from a **systems perspective**;
2. Applying a **system stakeholder view** of value, trade-offs, and optimization;
3. Understanding system's **interactions and states** (modes);
4. Specifying system **technical requirements**;
5. Creating and analyzing **high level design**;
6. Assessing solution **feasibility, consistency, and completeness**;
7. Performing system **failure mode and risk analysis**;
8. Planning system **families, platforms, and product lines**;
9. Understanding **roles and interdependencies across the innovation process**.

The Summer Grand Challenge Program primarily consists of rising sophomore and rising juniors from various majors. In this one course we focus on a subset of these competencies to broaden students' perspectives when considering a design. We included portions of stakeholders view, high level design, interactions and states models. These models will be discussed in the next section of this paper.

In a traditional design class, students traverse through the various number of steps of a design sequence and each step should be informed from the previous step. Because of the linear nature in which these steps are presented, often students' final step of the process is no longer clearly related to their first step (e.g. the decision matrix criteria is not connected to customer needs). Although the students are encouraged to go back and review previous steps, the linearity of the process makes them unlikely to "go back to step 1" as it is assumed completed and no longer necessary to develop their final design (Figure 2a). The beauty of the system model is that you can start anywhere and your system is not complete until all the models are consistent with each other, requiring a continuous improvement to all models (Figure 2b). In the next sections of the report, we will describe each model in detail.



(a)



(b)

Figure 2-(a) Traditional steps in developing a product taught during an undergraduate design course.[6] (b) Model based system engineering which allows for a less linear approach.[5]

Stakeholder and Feature Model

The stakeholder and feature model is a model the students prefer to complete first as it is the most familiar model to the students. They can relate to being a customer, and thus can begin to create a set of requirements that may be of interest to certain stakeholders. The completeness of this model will still likely require additional iterations as stakeholders unfamiliar to the students are discovered. To help students identify stakeholders other than customers the following questions can be asked:

- Who is dissatisfied with the current situation?
- Who is affected by changing the situation?
- Who will authorize a change in the situation?
- Who benefits from a solution?
- Who benefits from no solution being enacted?
- Who is affected if we fail to enact a solution?
- Who is affected if our solution fails?

The course has been taught three times and we have not seen a need to change the stakeholder feature model either in style, or in approach to teaching the model. Examples of student work from the first year and the third year of the course are shown in Table 1 and Figure 3. It can be seen that the students were able to grasp the concept of stakeholder features easily in the first year. For instance, the student team from the first course offering was able to recognize several stakeholders, ranging from the local tribe to the humanitarian organization, and was able to develop a relatively complete set of features, but are missing some of the features that the maintainer and the installer would care about--such as replicability and repairability. In addition, a few features were mapped incorrectly. For instance, the maintainer would not care about the visual appeal of the product. For the student group from the third offering of the course, only four main stakeholders were included, thus the list was not as complete. Each of the stakeholders were mapped to specific features, which was similar to the first team; however, this student group added features such as ease of repair, distributable, and ease of construction. The second model included both good descriptive titles for each feature as well as a clear definition. This prevented ambiguity in the meaning of a certain feature and led to better mapping of the stakeholders to their desired features.

Table 1-Stakeholder and feature model from a student team from the 1st offering of the course that was interested in providing clean water to a local tribe in Kenya

STAKEHOLDERS	FEATURES & ATTRIBUTES	MAPPING
1. Current water suppliers	a. Efficiency	1. a,b,c,d,e,f,g,h,i,m,n,o
2. Domestic Animals	b. Hours of operation	2. g,i,o
3. Entrepreneurs	c. Lifetime	3. a,c,d,e,f,g,h,i,j,m,n,o
4. Farmers	d. Local Manufacturability	4. b,c,e,g,h,i,j,m,n,o
5. Hospital staff	e. Price	5. a,b,c,e,g,h,i,k,m,o
6. Humanitarian organization	f. Product cost	6. b,c,d,e,f,g,h,i,j,k,l,m,n,o
7. Installation personnel	g. Quality of purification	7. c,d,i,j,l
8. Luo Culture	h. Reliability	8. d,j,k,m,n
9. Luo people	i. Safety	9. b,c,d,e,g,h,i,j,k,
10. Maintenance personnel	j. Size/portability	10. a,b,c,d,g,h,i,n,o
11. Masai government	k. Sustainability	11. a,c,d,e,f,g,h,i,k,l,n,o
12. Other locals	l. Type of power input	12. b,c,d,e,g,h,i,j,k,
13. Pastoralists	m. User-friendliness	13. b,c,d,e,g,h,i,j,m,n,o
14. Product designers	n. Visual Appeal	14. a,b,c,d,e,f,g,h,i,j,k,l,m,n,o
15. Students	o. Water Yield	15. b,c,e,g,h,i,m,n,o
16. Teachers		16. b,c,e,g,h,i,m,n,o
17. Tourists		17. a,b,d,g,h,i,n
18. Wild Animals		18. g,i,o

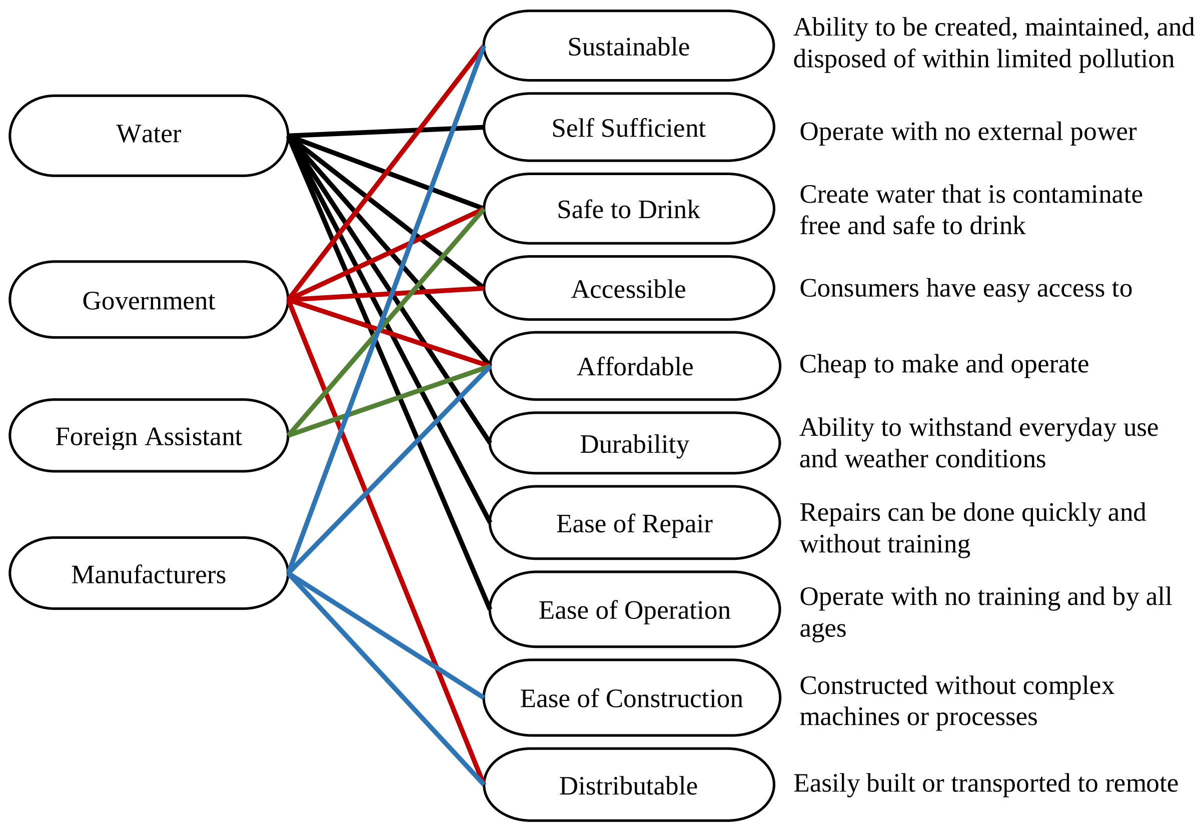


Figure 3-Stakeholder and feature model from a student team from the third offering of the course that was interested in providing clean water to the people of Haiti.

Domain Model & Logical Architecture

During the first offering of the Summer Grand Challenge Program, we had the students develop a domain model, followed by a logical architecture model (Figure 4). A domain model is intended to show persons or things that interact directly with the system. The domain model provided challenges to the students in recognizing what was an interaction with the environment. For instance, initially they forgot to include items such as weather conditions, including wind and rain. In addition stakeholders were included that did not directly interact with the device, such as the government. Also, features were accidentally included such as lifespan which should be captured in the stakeholder/feature model, whereas things that reduce the lifespan could have been included. An example would be wild animals that accidentally step on the device due to its use in the Savannah or birds that defecate on the device, reducing its reflective properties, ultimately diminishing its capability of purifying water. Students also had difficulty with the directionality of various interactions. Next, we had the students develop the logical architecture. It seemed fairly obvious to us that the domain model that was developed would influence the internal components that comprise the logical architecture; however, the students basically neglected the initial domain model and developed another set of external interactions. The students successfully developed several internal logical descriptions of possible components such as water pasteurizer and debris blocker, but still include some components that are better described as mass flows into and out of a component such as “large debris” which should not be considered logical components.

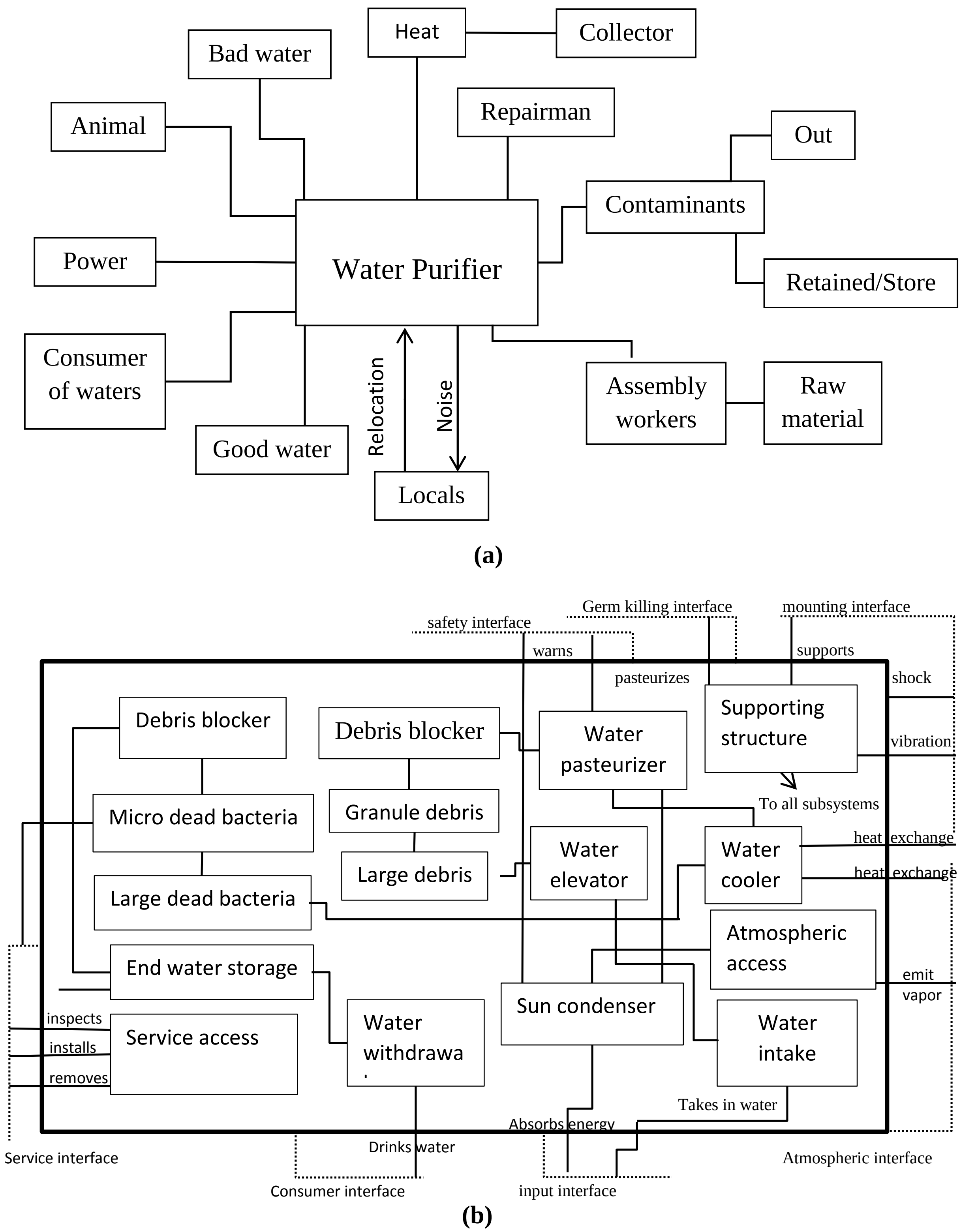


Figure 4-(a) Domain model and (b) logical architecture developed separately in two distinct exercises.

As can be seen, the students were unfamiliar with the concept of a domain model diagram, especially when a separate activity from the development of the logical architecture. To help with this issue, the model was renamed to a more familiar term--"black box"--which they are exposed to in their circuit classes. In addition, all components of the models were defined using grammatical states such as inputs/outputs (I/Os) were defined as nouns that describe what is being transferred between the actors and the system which includes power, energy, mass, and information. In order to identify objects that interact with the device other than simply stakeholders, we defined a new term "actors". Actors were defined as nouns that interact with their device. Some of the stakeholders listed may be actors but actors include more than simply people: actors can include environmental factors such as wind, rain, and/or other devices that physically come into contact with the system being designed. We also defined logical components as verbs followed by a noun. This helped ensure that the student no longer listed features as part of their logical architecture and prevented physical components from being named too early in the design process. This definition enabled better initial model development with much less iteration required. For instance in the first iteration students were able to define appropriate actors and I/Os. Features were no longer confused as actors and I/Os were no longer nebulous undefined connections. In addition, the directions of the I/Os were included in the diagram. We also saw an improvement of the model by having the students directly build upon their "black box" by adding the logical "white box" components to their model (Figure 5). Although the model is still not perfect as can be seen with the "transportation forces" that appears to interact with the "collect impurities", this model is more complete and easier to follow than the 1st time teaching the course.

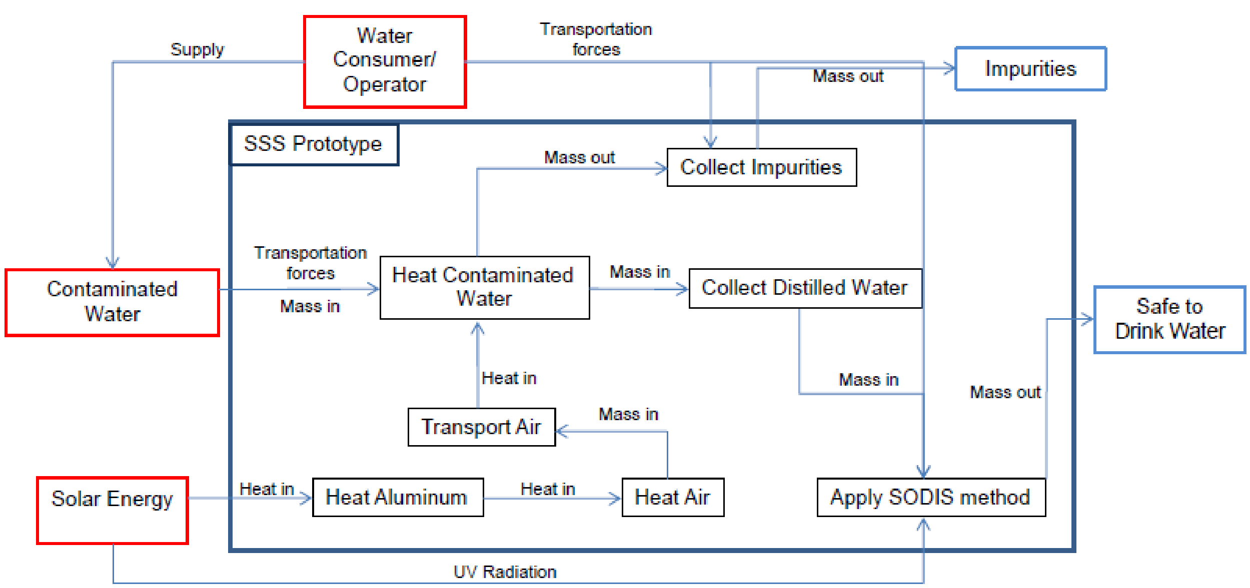


Figure 5-Students' "black box" model along with "white box" components that detail a water purification system that utilizes solar energy.

Conclusion

Through having the students develop models that describe their thought process regarding the system they planned on building later in the course, there was less communication errors as all students appeared to "be on the same page." This was evident through their ability to produce a final product much sooner during the class than the previous year such that they actually had time to perform additional testing. They also more regularly considered stakeholders when making design decisions. For instance, initially during the design process they tried to eliminate the use of wood because of deforestation issues in Haiti. In previous years, material considerations weren't considered until much later in the design process. They also used these models to ensure all aspects of the device were satisfying the initial features. For instance, portability was continuously discussed rather than waiting until the last minute as in previous years. In addition, design changes were incorporated such that one single person could setup and use the device, rather than requiring 8 people to lift a 200 lb awkward device as in the previous year's design. Overall, by incorporating the definitions of various components of the model based system approach with grammatical terms improved not only the systems models but also resulted in an overall improved final design. This is further demonstrated by the students being invited to submit an *IEEE Potentials*[7] magazine article and various other accolades.

Bibliography

[1] A. Bernal, S. Kirkpatrick, A. Watt, "What's in the Soup? Reflections from an Engineer, a Physicist, and an English Professor on an Interdisciplinary Summer Grand Challenge Program", *Proceedings of the 2014 Conference of the American Society for Engineering Education (ASEE),* June, 2014.
[2] http://www.engineeringchallenges.org/
[3] Schindel, W., "Requirements statements are transfer functions: An insight from model-based-systems engineering", *Proceedings of INCOSE 2005 Symposium*, July, 2005.
[4] W. Schindel, S. Peffers, J. Hanson, J. Ahmed & W. Kline, "All Innovation Is Innovation of Systems: An Integrated 3-D Model of Innovation Competencies", *Proceedings of the 2011 Conference of the American Society for Engineering Education (ASEE)*, June, 2011.
[5] M. Simoni, E. Andrijcic, B. Kline, A. Bernal, (2016, July), "Helping Undergraduate Students of Engineering Discipline Develop a Systems Perspective", *2016 Annual INCOSE International Symposium*, July, 2016. (accepted)
[6] K. Ulrich and S. Eppinger, Product Design and Development, Fifth Edition, McGraw-Hill, New York, 2012.
[7] K. Piens, A. Schultz, R. Tanaka, J. Atzinger, and C. Miannan, "Engineering relief for Haiti", IEEE Potentials, Vol. 34, No. 1, 2015.