Open Hardware Creators in Academia Fellowship

Unlocking the Potential:
A Course Curriculum Applying Model-Based Systems Engineering (MBSE) for Open Source Hardware (OSHW) in Academia

Applications of MBSE for Developing and Promoting the Use of OSHW in Academia

Dahl Winters
Steven J. Simske
Colorado State University
Department of Systems Engineering

OSHWA

Course Synopsis

Benefits:

1. Accelerate the adoption of OSHW in academia
2. Provide a comprehensive understanding of model-based systems engineering (MBSE) and its application to OSHW
3. Address key issues impeding the widespread use of OSHW in academia
4. Enable professors in engineering and related fields to leverage OSHW on high-value research projects

Expectations:

By the end of this course, participants will:

1. Understand the importance of MBSE in addressing challenges related to OSHW adoption
2. Gain knowledge on the 11 key issues hindering the use of OSHW in academia and how MBSE can overcome them
3. Be equipped with the necessary skills to implement MBSE techniques for OSHW projects
4. Feel confident in their ability to create high-quality documentation and overcome language barriers
5. Explore collaborative development and remixing possibilities for open-source hardware designs
6. Understand licensing and intellectual property rights related to OSHW and its documentation

Course Q&A

Q: Who is this course for?
A: This course is primarily designed for university professors in engineering and related fields who are interested in incorporating open-source hardware into their research projects.
Q: Do I need prior experience with model-based systems engineering or open-source hardware?
A: While prior knowledge is beneficial, this course is structured to cater to participants with varying levels of experience. We will start from the basics and gradually build upon the concepts.
Q: Will I receive a certificate upon completion?
A: Yes, participants who successfully complete the course will receive a certificate of completion to showcase their newfound expertise.
Q: Can I access the course materials after the course ends?
A: Yes, you will have lifetime access to the course materials, including videos, worksheets, and additional resources.
Q: How much time should I allocate each day for the course?
A: We recommend setting aside approximately one hour per day to fully engage with the course content and complete the assigned tasks.

The Course

1. Introduction

Title: Applications of Model-Based Systems Engineering for Developing and Promoting the Use of Open Source Hardware (OSHW) in Academia

Subject: The lack of widespread use of OSHW in academia is likely due to a lack of awareness of its contributions outside of academia to academically relevant fields, one of which is sustainability. This curriculum describes how to apply model-based systems engineering (MBSE) to address several key issues preventing more widespread use of OSHW in academia. This is a living document meant for continual improvement, thus it is a work in progress.

We will demonstrate the use of a free and open source program for accomplishing this goal. This curriculum contains 11 modules, each presenting an issue impeding the broader adoption of OSHW in academia and how MBSE is well suited to overcome these issues.

The list of 11 issues are as follows:

1. There are no unifying standards or best practices for creating high quality documentation.
2. Documentation for OSHW projects is dispersed across many platforms, websites, wikis and blogs
3. No clear definition of scope exists
4. Lack of standard formats, clear organization, and technical jargon makes it difficult for the layperson to understand existing documentation
5. Language is a barrier for the dissemination of open hardware plans
6. There's no simple way to remix and mashup hardware
7. Derivative work is difficult to track
8. Lack of appropriate software for designing, displaying and sharing plans makes collaborative development difficult
9. It's difficult to update and evolve open source hardware designs due to documentation dependencies
10. Documentation is time-consuming
11. Unclear licensing and fear of infringement of intellectual property (IP) rights discourage people from producing documentation

Target Audience: University professors primarily in engineering and related fields

Benefits: Accelerating the adoption of OSHW in academia by making it much easier to understand, adopt, use, and produce OSHW on high-value research projects. OSHW can also encourage community involvement while providing improved hands-on learning experiences.

Outcome: Upon successfully completing this curriculum on applying MBSE to address issues impeding the adoption of OSHW in academia, university professors should gain a deep understanding of the challenges and solutions related to OSHW documentation, collaboration, and licensing. As a result, they will be empowered to introduce OSHW into their research projects, effectively communicate its benefits to students, and contribute to the broader adoption of OSHW in academia.

2. Successful case studies and impactful projects

An area where OSHW has added great value to an area of strong academic interest is sustainability, to solve challenges such as improved environmental sensing and carbon dioxide removal. Below are case studies of open hardware projects that have successfully addressed sustainability challenges, as well as examples of how MBSE can be used to design sustainable open hardware systems.

1. OpenAir Collective: VIOLET Decarbonizer, CYAN Decarbonizer, Thursday - open hardware projects that use low-cost, open-source technology to capture carbon dioxide (VIOLET, Thursday) and store it in a stable mineral form (CYAN). These projects aim to address the challenge of reducing greenhouse gas emissions by providing a scalable and sustainable solution for carbon capture and storage.
2. Safecast: an open hardware project that uses a network of radiation sensors to collect data on environmental radiation levels. The project was started in response to the Fukushima nuclear disaster in 2011, and has since expanded to cover other areas around the world. The sensors are based on open source designs and are built using off-the-shelf components, making them easy to reproduce and deploy in a wide range of environments.
3. AirCasting: an open hardware project by HabitatMap, that uses a wearable air quality monitor to collect data on air pollution. The project was started in response to the growing concern over air pollution in urban areas, and is designed to empower citizens to monitor and report air pollution levels in their communities.
4. Open Source Ecology is a project around the development of a set of 50 open source machines that can be used to build a sustainable, self-sufficient community, which make up the Global Village Construction Set (GVCS). The project has been used by communities in over 10 countries to build homes, grow food, and generate energy.
5. The Open Source Beehives project is developing open source hardware and software for beekeepers to monitor their hives and track the health of their bees. The project has been used by beekeepers in over 50 countries to improve the health of their bees and increase honey production.
6. The FarmBot project is around the development of an open source, automated farming machine that can be used to grow food in small spaces, such as backyards or community gardens. The FarmBot has been used by farmers in over 20 countries to grow food more sustainably and efficiently.

7. There are several projects for Arduino Weather Stations - these are simple, open source weather stations for monitoring the weather in remote locations. Such projects have been used by scientists, farmers, and hobbyists to track weather data and make informed decisions about their activities.
8. AirGradient provides low-cost, open source and open hardware air quality monitors that can be used to track air pollution in real time. The project has been used by citizens, scientists, and policymakers to monitor air quality and advocate for cleaner air.
9. Ribbit Network provides open source sensors called “Frogs” that are open hardware for climate observation. Live sensor data are made visible from their website, demonstrating how a sensor network can be useful to teachers, makers, and scientists.

3. Exploring existing OSHW documentation and identifying challenges

Open Source Ecology: documentation of their 50 machines in its Global Village Construction Set (GVCS) is hosted on Dozuki, which uses an open, XML-based document standard for instruction manuals called oManual. Each of the 50 machines has its documentation broken down into about 10 modules and each module is broken down into about 75 development points listed in a spreadsheet for development. Trovebox is used to store all related pictures, and videos are uploaded to YouTube in real time.

OpenAir Collective: documentation of their open hardware carbon removal projects, both Violet and Cyan, are primarily on GitHub. The OpenAir Forum using Discourse provides a platform for discussion and file review (images, data on spreadsheets). Newer projects such as the Thursday direct air carbon capture device have also been added to Hackster.io to improve integration with the broader open hardware community there. Videos are also posted to YouTube after editing.

Commonalities: Both OSE and OAC emphasize the importance of sharing documentation openly, leveraging online platforms to host and share their documentation, and engaging with communities to foster collaboration and knowledge exchange in OSHW. Both use a means of documenting their development points, so that documentation can be organized and tracked – spreadsheets are used at OSE and the OpenAir Forum also provides such a structure. Media related to projects must also be stored, and such media may become a part of community engagement efforts on popular OSHW platforms (such as Hackster.io).

OSE additionally employs oManual, allowing for standardized and accessible documentation stored in an XML-based format that can be easily understood and used. UML Designer by Obeo is an open-source program under the Eclipse Public License (EPL), available at https://github.com/ObeoNetwork/UML-Designer/tree/master. It can also be downloaded for installation on Windows, MacOS, and Linux at https://www.umldesigner.org/download/.

4. Challenges - Course Modules

Each module examines one of 11 issues impeding the broader adoption of OSHW in academia and how MBSE is well suited to overcome these issues. ("Open Source Hardware Documentation Jam » The Challenge")

1. There are no unifying standards or best practices for creating high quality documentation.
2. Documentation for OSHW projects is dispersed across many platforms, websites, wikis and blogs
3. No clear definition of scope exists
4. Lack of standard formats, clear organization, and technical jargon makes it difficult for the layperson to understand existing documentation
5. Language is a barrier for the dissemination of open hardware plans
6. There's no simple way to remix and mashup hardware
7. Derivative work is difficult to track
8. Lack of appropriate software for designing, displaying and sharing plans makes collaborative development difficult
9. It's difficult to update and evolve open source hardware designs due to documentation dependencies
10. Documentation is time-consuming
11. Unclear licensing and fear of infringement of intellectual property (IP) rights discourage people from producing documentation

Issue 1: There are no unifying standards or best practices for creating high-quality documentation.

Solution: Develop a comprehensive guide on creating standardized documentation for OSHW projects, including templates and examples. Emphasize the importance of clear and concise documentation to facilitate understanding and replication.

Issue 2: Documentation for OSHW projects is dispersed across many platforms, websites, wikis, and blogs.

Solution: Create a centralized platform or website specifically dedicated to hosting and organizing OSHW project documentation. Implement search functionalities and tagging systems to make it easy for users to find relevant documentation.

Issue 3: No clear definition of scope exists.

Solution: Offer guidance on defining the scope of OSHW projects, including their intended purpose, target audience, and expected outcomes. Provide examples of well-defined scopes to serve as references.

Issue 4: Lack of standard formats, clear organization, and technical jargon make it difficult for the layperson to understand existing documentation.

Solution: Develop a module that focuses on simplifying technical jargon and explaining complex concepts in a more accessible manner. Encourage the use of standardized formats and clear organization structures in documentation.

Issue 5: Language is a barrier for the dissemination of open hardware plans.

Solution: Provide resources for translating OSHW documentation into different languages. Collaborate with volunteers or language professionals to create translated versions and encourage the community to contribute translations.

Issue 6: There's no simple way to remix and mashup hardware.

Solution: Introduce tools and techniques for remixing and integrating OSHW components. Highlight successful examples of hardware remixing and provide step-by-step guides on how to approach and execute such projects.

Issue 7: Derivative work is difficult to track.

Solution: Present methods and tools for tracking derivative works of OSHW projects. Explore the use of version control systems or project management platforms to document and track modifications made to original designs.

Issue 8: Lack of appropriate software for designing, displaying, and sharing plans makes collaborative development difficult.

Solution: Identify and showcase software tools that support collaborative OSHW development, design, and visualization. Provide tutorials and walkthroughs on using these tools effectively for team-based projects.

Issue 9: It's difficult to update and evolve open-source hardware designs due to documentation dependencies.

Solution: Outline strategies for managing documentation dependencies and version control in OSHW projects. Highlight the benefits of maintaining up-to-date documentation and provide guidelines for updating designs while preserving backward compatibility.

Issue 10: Documentation is time-consuming.

Solution: Offer time-saving tips and techniques for creating OSHW documentation efficiently. Provide templates, checklists, and automation tools that can streamline the documentation process without sacrificing quality.

Issue 11: Unclear licensing and fear of infringement of intellectual property (IP) rights discourage people from producing documentation.

Solution: Educate users on different licensing options for OSHW and their implications. Explain how to properly license and protect intellectual property rights. Showcase success stories of individuals or organizations that have effectively utilized OSHW licenses.

5. Establishing documentation standards and best practices inspired by MBSE

Why use MBSE for documenting OSHW

MBSE can contribute valuably to the design of sustainable open hardware systems and can help with designing sustainable open hardware systems in several ways:

1. Designing modular systems that can be easily modified and adapted to different applications, reducing the need for new hardware designs and minimizing waste.
2. Optimizing the energy efficiency of open hardware systems by modeling the energy consumption of different components and identifying ways to reduce energy usage.
3. Optimizing the sustainability of open hardware supply chains by modeling the environmental impact of different manufacturing processes and identifying ways to reduce the carbon footprint of the production process.

Use a set of standardized formats and structures for OSHW documentation

Standard system modeling languages like SysML offer a rigorous and formal way to make a system design easily understood and modified by others. This offers the ability to organize the documentation as the system is organized, with a modular design that makes the system easy to modify and adapt to new uses. Modular designs that separate the system into distinct components makes it easier to modify or replace individual components without affecting the rest of the system.

For broad use in open hardware, diagramming approaches similar to MBSE using languages like SysML but that do not require expensive diagramming software or a significant learning curve may be desirable, as in the case where projects require the help of partners outside a university that may not have the budget or time to learn. There are three diagramming approaches - flowcharts, block diagrams, and mind maps - that still allow a visual representation of components within a complex system and can support organizing documentation in a modular way.

Through the use of these three diagramming methods, the principles of MBSE can be more broadly applied beyond systems engineering, while retaining the documentation simplification ability of MBSE which improves understanding of systems and subsystems, how they interact and interface with each other, and thus how they can interface with other systems to rapidly iterate and innovate on new designs.

Putting Diagrams Into Practice

Visual diagramming is useful in complex system designs where, for example, a variety of environmental sensors providing data for model training and subsequent inference are used to signal the actuation of a set of responses (e.g. removing carbon or supplying irrigation). **Flowcharts** offer shapes and connections between them that can be labeled with one or more interactions or data flows. **Block diagrams** use blocks to represent the different components of the system, and also lines to represent the connections between them. **Mind maps** use a central image to represent the main idea or function of a system, and the central image is then surrounded by branches that represent the different components of the system.

All three of these are easy to create and understand, and can be used to communicate complex ideas in a clear and concise way by representing the structure of a system, the flow of data, or the relationships between different components of that system.

Follow guidelines on creating clear, concise, and comprehensive documentation

The **system requirements** should be documented early in the design process and these should be traced throughout the design and testing phases. This helps provide support when the system is validated, to ensure that it meets its intended goals. The **rationale behind design decisions** should also be documented, such as why a particular component or interface was chosen. This helps others to understand the design and to modify it for their own purposes. And to track design changes and to collaborate effectively with others on making those changes, **version control software** such as Git is necessary. This helps to not only keep track of changes, but changes can be merged from multiple contributors and rolled back if necessary.

6. Collaboration and Remixing in OSHW

MBSE has already been used in the past to successfully design, document, and test open hardware systems. To get started using it, NASA offers a good video resource: How to Get Started Using MBSE on a Project: The Basics of What, How and Who | NESC Academy Online.

1. OpenROV is an open source underwater robot that can be used for scientific research, exploration, and education. The OpenROV team used MBSE techniques to design and document the hardware, software, and control systems for the robot, which helped them to produce a highly functional and customizable system that can be adapted to a wide range of applications.
2. ArduSat is a satellite platform that can be used for scientific research and education. The ArduSat team used MBSE techniques to design and document the hardware, software, and data acquisition systems for the satellite, which helped them to produce a flexible and modular system that can be adapted to a wide range of scientific experiments.
3. The Open Source Medical Supplies library contains a large number of projects with open source designs for medical devices. The project uses a modular approach, with each module representing a specific function, such as a surgical instrument, a diagnostic tool, or a patient monitoring device.

7. How MBSE can benefit OSHW implementation in academic research

The following benefits are possible by applying MBSE to develop and innovate on sustainable open hardware designs:

1. Open hardware can facilitate the development and deployment of new renewable energy technologies, such as solar panels, wind turbines, and energy storage systems. By making designs and specifications freely available, open hardware can lower the barriers to entry for individuals and organizations looking to develop and commercialize these technologies.
2. Relationship diagramming can be used to model complex systems and identify areas where improvements can be made to increase efficiency, reduce waste, and lower emissions. By creating detailed diagrams of energy systems, for example, it may be possible to identify areas where energy is being lost and design interventions to reduce this waste.
3. MBSE, through SysML, can be used to model the energy system at a higher level, helping to identify dependencies and potential areas for optimization. By creating a model of the entire energy system, it may be possible to identify areas where small changes could have a significant impact on overall energy consumption and emissions.
4. Open hardware can also help to reduce the environmental impact of manufacturing and production by enabling more efficient use of resources and reducing waste. By making hardware designs freely available, it may be possible to reduce the number of iterations required to develop a product and to minimize the use of materials and resources throughout the manufacturing process.
5. Finally, open hardware can facilitate greater collaboration and knowledge-sharing between individuals and organizations working on renewable energy and carbon reduction initiatives. By enabling more people to contribute to the development and improvement of these technologies, it may be possible to accelerate the pace of innovation and make progress towards these goals more quickly.

Raising Course Awareness (Pre-Registration)

Pre-Launch

Emails

Pre-Launch Email 1: “Unlock the Power of Open Source Hardware in Academia”

- Introduce the course and highlight the benefits of adopting OSHW in academia
- Share testimonials or success stories from OSHWA Trailblazer Fellows and previous participants when we have those
- Encourage prospective attendees to join the course waitlist or register early

Pre-Launch Email 2: “Overcoming Challenges: MBSE for OSHW Documentation”

- Highlight the key issues faced in OSHW documentation and the role of MBSE in addressing them
- Emphasize the practicality and relevance of the course content
- Offer a sneak peek into the curriculum and what participants can expect to learn

Pre-Launch Email 3: “Limited-Time Offer: Early Bird Enrollment for OSHW Course”

- Create a sense of urgency by announcing limited-time early bird enrollment
- Highlight exclusive bonuses or discounts for early sign-ups
- Remind potential attendees about the benefits of the course and the positive impact it can have on their research

Pre-Launch Video

“Welcome, future innovators! Are you ready to unlock the potential of open-source hardware in academia? Our upcoming course on “Applications of Model-Based Systems Engineering for OSHW in Academia” will equip you with the skills and knowledge to overcome the barriers holding back the widespread adoption of OSHW. Join us as we dive into 11 key issues and explore how MBSE can revolutionize the way we approach OSHW documentation, collaboration, and more. Enroll now and be part of a movement that accelerates research and innovation in academia!”

Pre-Launch Posts

Success stories of universities and/or researchers who have already implemented OSHW using MBSE techniques – Open-Source Ecology at UM-Columbia for example.

A visually appealing infographic highlighting the benefits of OSHW and MBSE.

A live Q&A session on social media, answering questions about OSHW and MBSE.

During the Course

Instructional Emails (one per day)

- Each instructional email should provide valuable content related to the day's module
- Include clear instructions for the student's daily tasks and activities
- Incorporate interactive elements such as quizzes, challenges, or group discussions to enhance engagement

Pre-Launch Social Media Posts

- "Are you ready to revolutionize how academic research is done? Our upcoming course will show you how to leverage open-source hardware in your research projects. Stay tuned!"
- "Attention all professors in engineering and related fields! Discover the untapped potential of OSHW and learn how to overcome key challenges. Our course is coming soon!"
- "Calling all academic innovators! Join our waitlist and be the first to know when our course on MBSE for OSHW launches. Don't miss out on this opportunity to learn how to cut time while accelerating your research!"