SafeWalk: a Simulation Tool Kit for Exploring Software Requirements in a Safety-Critical Product Line

James I. Lathrop*, Robyn R. Lutz†, Cameron Brecount‡, Hugh Potter§,
Kathryn Rohlfing ¶, Jesse Slater ‖ and Joshua Wallin **
Department of Computer Science, Iowa State University, Ames, Iowa, USA
Email: *jil@iastate.edu, †rlutz@iastate.edu, ‡brecount@iastate.edu §hdpotter@iastate.edu
¶kathrynr@iastate.edu ‖jcslater@iastate.edu **wallin.j@northeastern.edu

***Abstract*—SafeWalk is a tool kit designed for simulation of a safety-critical product line of astronaut jetpacks. It provides (1) a Unity-based simulation development environment and (2) software artifacts inspired by a real jetpack used by astronauts during spacewalks. SafeWalk has been developed to be a readily extensible and real-time configurable product line for use in research and education. It provides a rich environment conducive to empirical research into safety-critical requirements and to visualization of safety-related human-cyberphysical interactions.**

***Index Terms*—Requirements engineering, Simulation toolkit, Safety-critical, Product Line**

I. Introduction

SafeWalk is a tool kit designed for simulation of a safety-critical product line project. It consists of the simulation development environment and the project's software artifacts. SafeWalk is inspired by a real NASA backpack propulsion unit ("jetpack") used by astronauts to return safely to a space station's airlock if their tether fails during a spacewalk [2]. SafeWalk has been developed as a configurable and extensible product line with multiple versions and realistic variants to simulate. It is designed to provide a rich environment conducive to empirical research into safety-critical requirements and to visualization of safety-related human-cyberphysical interactions.

SafeWalk provides a simple software simulator of a NASA-inspired astronaut jetpack, extended by our project into a product line, together with the software artifacts created in its development. Its creation was motivated by the lack of readily available safety-critical software product lines for research and education. This lack of artifacts impedes both researchers and educators. SafeWalk has been developed to help fill this gap. It supports research and teaching for safety-critical software product lines' requirements engineering.

Intended users. The intended users are researchers exploring the requirements engineering of safety-critical product lines, especially those involving safety-critical human-cyberphysical interactions, and instructors of RE project-based courses, especially those involving student-team projects to create and simulate alternative and variant requirements.

RE challenge addressed. Simulation tools are a powerful mechanism to explore and demonstrate new ideas, including in requirements engineering [5], but can be time-consuming and tricky to use. To be practical, our SafeWalk tool kit needs to be easy for a researcher or student to add new features to the simulated model; the simulation must enable visualization and exploration of alternative requirements, e.g., to mitigate hazards; and the simulated model must have traceability among its evolving software development artifacts. The simulation solution must be configurable, including at runtime; include both the user who engages in the simulated activity and the operator who configures and observes the simulation; and invoke a simplified but adequately realistic physics engine.

Proposed solution. SafeWalk is a new simulation toolkit to specify, design and simulate software requirements and configuration interfaces for the user (the "astronaut") and the future developer (the researcher or student extending the SafeWalk product line with additional capabilities) so that the results are elegant, easy to understand and maintain, and suitable for open-sourcing, allowing others to build on them.

II. SafeWalk methodology and tool kit

The methodology underlying SafeWalk is goal-oriented requirements engineering, with a focus on obstacle analysis [6]. We developed the requirements for a simple software simulator of an astronaut's self-rescue using a jetpack in order to safely return to a space station airlock if their tether to the space station fails during a spacewalk. Hazards are thus obstacles to achieving the safety requirements. The available software artifacts for SafeWalk include the jetpack product line's feature model, safety hazards, software requirements, architectural design, sequence diagrams, and source code.

SafeWalk's simulation tool kit harnesses rich visualization capabilities through its use of the Unity game and physics engine to process user inputs and sensor inputs, control simulated thrusters, monitor at runtime for safe operation, and update the operator (simulation controller) and user (astronaut) displays in order to reach the target airlock in the reference frame. The Unity-based simulation development environment

2332-6441/22/$31.00 ©2022 IEEE
DOI 10.1109/RE54965.2022.00038

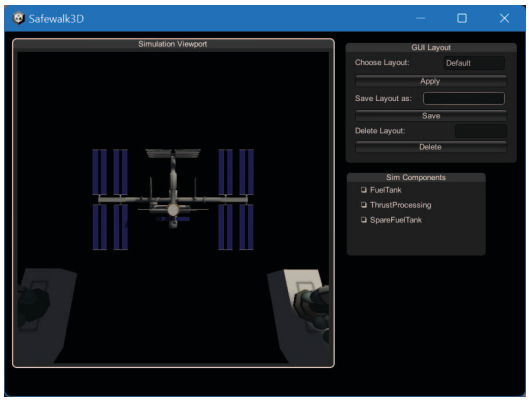

Fig. 1. An example of the SafeWalk simulation tool from the operator's point of view, as the astronaut/user being monitored approaches the space station.

is used for visualizing and playing out various safety-critical mission scenarios.

Example scenario. We designed SafeWalk so that it is easy for users to add their own features to the baseline product line. The tool demonstration video provided with this paper (see link at end) shows us adding a new product-line feature and then simulating it. The example *safety-critical scenario* shown is that an astronaut runs out of fuel (propellant) before reaching the space station. Runtime monitoring gives information about the fuel remaining. To mitigate the *hazard*, we show how we can add a *new feature*: a Backup Fuel Tank. This, in turn, enables a *new software requirement*: "When SafeWalk detects low fuel, it shall switch over to using the backup tank." The tool demonstration video shows how we then *simulate this safety-critical scenario* in order to verify that the safety requirement is met in the new product and to validate that the new requirement achieves its intended behavior.

III. Architecture and Implementation

Our tool uses the Unity game engine to implement an entity component architecture and a modular design. The architecture supports two roles: the *operator* (researcher or developer) who selects the product-line features to enable and configures the simulator controller, and the *user* (player/astronaut) who executes the mission simulation. The baseline product runs the simulation on a computer. The implementation allows SafeWalk to be extended later to remote operators, to be run in a browser, and to interface with a VR headset.

Implementation. SafeWalk is implemented in Unity and C#, and developers extend the baseline product using the C# language. Figure 1 shows a screenshot of an example simulation of an astronaut maneuvering toward the International Space Station (ISS) with the jetpack's arms visible in the foreground and the ISS visible in the background. The figure shows the default configuration display that the operator sees.

SafeWalk evaluation. We performed unit and interface tests on our jetpack model using the simulations. We also used an early version of SafeWalk for student projects in an RE course, and describe our experience in [4]. Current limitations include simplifications of the thruster behavior and the physics engine.

Availability. Our project plans to release the SafeWalk development environment for product-line research on the jetpack, which enables addition and simulation of features and products, as well as our development software artifacts. SafeWalk requires Unity to be installed, and Unity provides a royalty-free license for educational and personal use.

Related industrial and research efforts. Gamification, as in SafeWalk, uses gaming engines for development of non-gaming applications such as training or investigation, and has proven effective for multiple software engineering purposes [1], [5]. We also benefited from public information about two NASA simulators for its SAFER backpack propulsion device, one a complex, physical simulation for astronaut training and the other a VR simulation on the space station [3].

IV. Future Plans and Conclusion

Support for RE research and instruction. SafeWalk gives RE researchers the opportunity to visualize and evaluate their ideas in a configurable simulation environment with product-line software artifacts and development infrastructure already built-out. Examples of RE research questions that might be explored in SafeWalk are exploring preferences among alternative requirements; identifying unwanted interactions between product-line features; gauging the acceptability of eliciting requirements using game-like scenarios; using simulation visualizations to verify safety requirements; and probing the future role of simulation in safety arguments. For educators, an example assignment for student team projects might be to extend SafeWalk with a new feature that, given a safe path from the starting position of the user (the astronaut) to the space station airlock, automatically synthesizes parameters to reduce time, battery use, or propellant, while maintaining the required safety properties.

In summary, SafeWalk provides a simulation environment that supports focused exploration of software requirements elicitation, analysis, and validation questions for a safety-critical jetpack product line.

V. Acknowledgments

This research is supported in part by NSF grants 1513717 and 1900716.

Tool Video: https://iastate.box.com/v/SafeWalkToolDemo

References

[1] A. Bucchiarone, A. Cicchetti, and A. Marconi, "Towards engineering future gameful applications," in *ICSE-NIER*. ACM, 2020, pp. 105–108.
[2] J. Crow *et al.*, *NASA Formal Methods Specification and Verification Guidebook, NASA-GB-001-97, Vol. 2*, 1997.
[3] A. D. Garcia, J. Schlueter, and E. Paddock, *Training Astronauts Using Hardware-in-the-Loop Simulations and Virtual Reality*, 2020.
[4] R. R. Lutz, J. I. Lathrop, C. Brecount, K. Gast, K. Rohlfing, and J. Wallin, "Using an astronaut jetpack project to teach human-CPS requirements engineering," in *REET@RE 2020*. IEEE, 2020, pp. 9–10.
[5] H. Samin, N. Bencomo, and P. Sawyer, "Pri-aware: Tool support for priority-aware decision-making under uncertainty," in *RE 2021*. IEEE, pp. 450–451.
[6] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009.