

User Interface Design for Multi-Objective Decision Making

Ira WINDER^{a,b,1} and Kazuo HIEKATA^b

^a*Massachusetts Institute of Technology, Engineering Systems Laboratory, USA*

^b*University of Tokyo, Industrial Systems Laboratory, Japan*

Abstract. We seek to accelerate the adoption of multi-objective decision making (MODM) methods within transdisciplinary engineering. To this end, we specify a generic user interface that makes computational systems models more accessible to non-technical decision makers. The collection of user stories presented in this paper allude to minimum viable features to include in the future development and testing of a generic user interface for multi-objective decision making.

Keywords. Multi-Objective Decision Making, Decision Support Systems, User Interface Design, Systems Modeling, Transdisciplinary Engineering

Introduction

Research engineers (i.e. model developers) who create high quality computational models for multi-objective decision making (MODM) rarely have time nor resources to consider the design of thoughtful user interfaces to go along with them. This is especially painful for decision-makers who might benefit most from such computational models. To help bridge the gap between decision makers and model developers, we specify the design of a generic user interface that allows decision makers to more easily configure and evaluate engineering solutions using computational models. We present the design of this interface via a set of user stories that are compatible with agile software development techniques, as we intend to expedite future software development based on our findings.

Multiple Objectives in Transdisciplinary Engineering

Transdisciplinary projects are implicitly complex endeavors, if only due to the diversity of stakeholders that they necessitate. Each stakeholder in a transdisciplinary engineering endeavor indeed brings one or more unique perspectives from which they may judge an outcome's success. It is common to describe judgements with an objective function that dispassionately reduces some aspect of an outcome's performance to a single number. For instance, an environmental engineer might judge an outcome by its total carbon footprint, while a financial manager might judge the same outcome by its cost [1]. Therefore, we might expect that most transdisciplinary endeavors are especially burdened by multiple objective functions.

¹ Corresponding Author, Mail: jiw@mit.edu.

Multi-Objective Decision Making

Multi-objective decision making (MODM) is the science of picking one or more engineering solutions that satisfy two or more objective functions. A valid solution is described as “non-dominated” when there exists no solution that performs better in all objectives [2]. The sum of such solutions is described as the “solution space,” or the set of all solutions that one might argue are acceptable given known alternatives. A common phenomenon in MODM, however, is that the number of non-dominated solutions can increase dramatically as one incorporates more objectives into the process of decision making. Without other methods to narrow down the solution space, this can lead to decision paralysis [3]. Therefore, we believe that the narrowing of solutions, while considering multiple objectives, is a principal pain point in the realm of transdisciplinary engineering.

Decision Support Systems

A decision support system (DSS) is a computerized interactive system that utilizes data and models to support a decision maker (DM) [4, 5]. As such, the typical components of a DSS include a (1) database, (2) model, (3) user interface, and (4) decision maker. In their comprehensive survey of DSS applications, Eom identifies an ongoing yet urgent need to bridge the gap between practitioners and DSS researchers. Despite much progress discovering the general components of DSS, they argue, there is not enough focus given to generic methods and techniques that would give practitioners better access to DSS in a number of underdeveloped subspecialties.

DSS and MODM Integration

Integrating methods of MODM with DSS is not terribly new. In fact, Razmak and Aouni published a study of multi-criteria decision support systems (MCDSS), observing dozens of use cases across multiple fields [6]. Weistroffer et al observed, however, that the vast majority of software in this subfield is “quasi-experimental, developed by academic researchers to test specific algorithms or to solve a specific problem on an ad hoc basis” [2]. While there are some examples of generic MCDSS, such as the OMOptim tool in OpenModelica, we believe that much work can be done to create generic MCDSS that is sufficiently user-friendly to be readily adopted by contemporary non-technical decision makers [7].

Decision Makers and Model Developers

Prior research by Mieg clarifies the bifurcated roles of “system experts” (i.e. model developers) and “decision making experts” in transdisciplinary engineering. While Harald identifies a need for decision makers to “professionalize themselves” so that they can more adeptly consider multiple systems domains, the research declines to offer concrete means of such professionalization [8]. As such, we believe that a thoughtfully designed generic user interface for MCDCC will help decision makers participate more “professionally” by allowing them to interface more smoothly with one or more systems models in any given transdisciplinary engineering problem.

Though the primary benefactor of our proposed generic user interface would be quintessential decision makers (e.g. non-technical executives), we must also consider the needs of model developers (e.g. engineers). For instance, imagine a logistics team and a financial analysis team are developing a multi-criteria computational model to help an adjacent executive team make a difficult decision about how to deploy a logistic network for their company (Figure 1) [9].

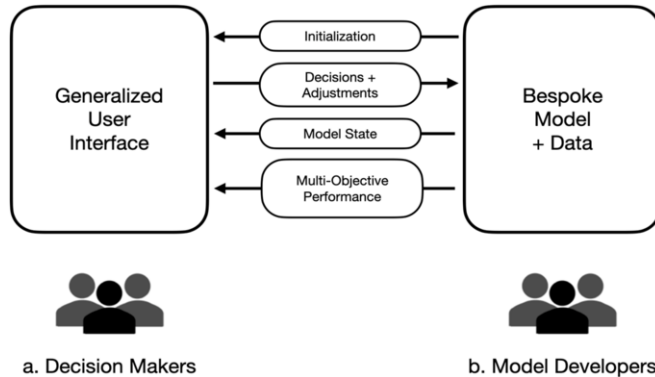


Figure 1. This diagrams shows the interactions between users and different DSS components.

In this case, the executive team are the (a) decision makers, while the logistics team and financial team working together are the (b) model developers. Model developers have proficiency in both their domain of expertise (e.g. logistics and finance) as well as computation. In practice, however, small teams of model developers are usually not well-versed in the nuances of user interface design. Therefore, we would like to specify a generic user interface to bridge this gap. Such an interface connects directly to a multi-objective systems model encoded by systems engineer(s) with specialized knowledge.

1. Objective

We seek to accelerate the adoption of multi-objective decision making methods for transdisciplinary engineering by specifying a generic user interface that makes computational systems models more accessible to decision makers. Our principal objective in this paper is to specify key features and capabilities of a generic user interface that might make it easier for researchers to effectively deploy multi-objective decision models to decision makers.

2. Method

We present our generic interface design as a set of user stories, a term borrowed from agile software development. A user story is an informal description of a feature from the point of view of the eventual user of a software system [10]. By adopting the language of user stories, we hope to expedite the development of future software that incorporates our findings.

We identify key user stories for a generic MCDSS for decision makers via qualitative analysis of patterns observed across a set of already-completed bespoke user interface case studies. The case study set includes pieces of bespoke MCDSS software developed during the course of the author’s prior research. While we considered case studies outside of our own research, this proved difficult for two reasons. Firstly, the vast majority of user interfaces developed for MODM seem to be meant for model developers, rather than decision makers. OpenModelica, for instance, may practically be used by a model developer to generate a report for the benefit of decision makers, but is not meant to be used in the board room, per se. Secondly, despite incidents of bespoke MCDSS software in literature, documentation of user experience is lacking.

We expect to be among the first users of this new tool, so operationalizing features that we have consistently implemented across multiple projects and domains should also expedite future research within our laboratory. We hope, however, that our conclusions will be applicable to other researchers who desire a standard user interface for their own multi-criteria decision models.



Figure 2. Case studies of bespoke multi-criteria decision support systems built by the author. They include (A) maritime shipping simulation, (B) tangible interactive urban planning, (C) pharmaceutical manufacturing simulation, (D) campus design, (E) autonomous vehicle impact analysis, and (F) real estate financing.

3. Results

3.1. Case Studies

Our set of MCDSS case studies includes examples from the following domains: maritime shipping [11, 12], urban master planning [13, 14], pharmaceutical manufacturing, campus design [15], autonomous vehicle impact analysis [16], and real estate investment financing (Figure 2).

3.2. User Stories

We summarize our results as user stories within ten broad categories of user experience. In brackets, we identify which of our six case studies {A, B, C, D, E, or F} inspired each category. For each story, we distinguish decision makers and model developers.

3.2.1. Logical and Quantitative Decision Making {A, B, C, D, F}

1. A decision maker can configure one or more decisions related to the specification of a solution using intuitive means such as toggles, radio buttons, and sliders. A set of decisions constitutes a solution.
2. A model developer can specify the nature of decisions available in the user interface.
3. A model developer's model can receive decisions from the user interface formatted as a series of booleans, integers, or floating point numbers.

3.2.2. Geographic Decision Making {B, F}

1. A decision maker can specify geo-located points and polygons via a map or abstract surface.
2. A model developer can specify the extent and nature of geo-located decisions available in the user interface.
3. A model developer can receive geo-located inputs as a series of coordinates from the interface.

3.2.3. Adjusting for Uncertainty {E, F}

1. A decision maker can adjust assumptions inherent to the model, including probabilistic distributions. Assumptions are clearly distinct from decisions.
2. A model developer can make certain data and assumptions in the model changeable via the user interface.
3. A model developer's model can receive updates to assumptions, including any probabilistic uncertainty.

3.2.4. Model State Diagram {A, B, C, D, E, F}

1. A decision maker can view an abstraction of a model as a 2D or 3D diagram of its state.
2. A model developer can specify the content of model's state, which is then automatically rendered as a 2D or 3D diagram by the interface

3.2.5. Save, Recall, and Viewing of Solution Iterations $\{A, D, F\}$

1. A decision maker may save, preview, and recall any number of solution configurations from memory. Multiple solutions are automatically organized as a tree of iterations that reveal the evolution of a user's decisions over time.

3.2.6. Multi-Objective Plots $\{A, B, C, D, E, F\}$

1. A decision maker can view a plot of solution performance according to multiple objectives. They may also assign weights and thresholds to specific objectives, or hide them altogether. They may also automatically filter for non-dominated scenarios.
2. A model developer can specify any number of objectives, their description, their units, and any minimum or maximum thresholds.
3. A model developer's model can simulate the performance of a solution and send them back to the user interface as absolute quantities or probability distributions.

3.2.7. Deployability $\{A, B, C, D, E, F\}^2$

1. A decision maker can easily access a pre-configured interface (e.g. via secure web application)
2. A model developer can easily generate and share their model to decision makers as an interactive MCDSS.

3.2.8. Adoptability $\{A, B, D\}$

1. A model developer can easily understand the application programming interface (API) for the interactive simulation.

3.2.9. Sensitivity Analysis $\{F\}^3$

1. A decision maker may see decision sensitivities before they are even made. For instance, a slider might indicate which direction it may be moved in order to improve the outcome of one or more objectives.

3.2.10. Solution Generation $\{ \}^4$

1. A decision maker user may leverage algorithms (e.g. genetic algorithms) to help them discover local optimizations within a solution space.

4. Discussion and Future Work

This paper leaves us with at least two major tasks ahead. First, there is the issue of building a minimum viable piece of software that satisfies all of our user stories.

² The authors believe that all six case studies were generally lacking in the area of deployability, which may explain why all of them inspired a need for improvement.

³ Sensitivity analysis is a feature only recently developed, so it is only inspired by a sole case study.

⁴ Solution generation is the only user story without any prior precedent among our case studies. However, we believe it is necessary to include some form of automated solution generation in order to test the performance of the interface during development. We explain this further in the section on future work.

Secondly, we must formulate a method for testing the effectiveness of our generic interface compared to some status quo condition.

We expect to build out the software as an open source project made freely available on a platform such as GitHub using the MIT Licence. While initial work will be conducted jointly by the UTokyo Industrial Systems Laboratory and the MIT Engineering Systems Laboratory, we welcome contributions from our peers.

As for testing and validating the generic user interface, we will develop an experimental method for measuring a user's ability to satisfactorily discover non-dominated solutions in a series of toy models. Over a number of statistically significant trials, we will compare human decision maker performance to that of a genetic algorithm that is designed to search for local optimizations within these toy models. To be considered successful, a decision-making human should be able to perform at least as well as a locally-optimizing algorithm. Ideally, we may find that a human can sometimes be better than their algorithmic counterpart at finding a global optimum.

5. Conclusion

The user stories in this paper define minimum viable research objectives to kick off the long-term development and testing of a generic user interface for multi-objective decision making. While the authors of this paper intend to subsequently embark on this endeavor, we welcome members of the research community to review our preliminary findings and, furthermore, we invite them to join us in any capacity. Ultimately, we hope this work will lead to increased adoption of multi-objective decision making methods in transdisciplinary fields.

Acknowledgement

The author of this paper is supported by a generous grant from Mercari Japan via University of Tokyo's Research Institute for an Inclusive Society through Engineering (R.I.I.S.E.), which is directed by Professor Yoshihiro Kawahara. The case study for pharmaceutical manufacturing was disclosed with permission from GlaxoSmithKline, and the case study for real estate finance development was disclosed with permission of Lend Lease, LLC.

References

- [1] F. Ascione, N. Bianco, De Masi, R. F., Mauro, G. M., & Vanoli, G. P. Energy retrofit of educational buildings: Transient energy simulations, model calibration and multi-objective optimization towards nearly zero-energy performance, *Energy and Buildings*, 2017, Vol. 144, pp. 303–319.
- [2] H.R. Weistroffer, C.H. Smith and S.C. Narula, Multiple criteria decision support software. In J. Figueira, S. Greco and M. Ehrgott (eds.) *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer New York LLC, 2005, Vol. 78, pp. 989–1018.
- [3] G. Montibeller, H. Gummer and D. Tumidei, Combining scenario planning and multi-criteria decision analysis in practice, *Journal of Multi-Criteria Decision Analysis*, 2006, Vol. 14(1–3), pp. 5–20.
- [4] S.B. Eom, S.M. Lee, E.B. Kim and C. Somarajan, A survey of decision support system applications (1988-1994), *Journal of the Operational Research Society*, 1998, Vol. 49, pp. 109-120.
- [5] S. Eom and E. Kim, A survey of decision support system applications (1995-2001), *Journal of the Operational Research Society*, 2006, Vol. 57(11), pp. 1264–1278.

- [6] J. Razmak and B. Aouni, Decision Support System and Multi-Criteria Decision Aid: A State of the Art and Perspectives, *Journal of Multi-Criteria Decision Analysis*, 2015, Vol. 22(1–2), pp. 101–117.
- [7] P. Fritzson, A. Pop, K. Abdelhak, A. Ashgar, B. Bachmann, W. Braun, D. Bouskela, R. Braun, L. Buffoni, F. Casella, R. Castro, R. Franke, D. Fritzson, M. Gebremedhin, A. Heuermann, B. Lie, A. Mengist, L. Mikelsons, K. Moudgalya, ... P. Östlund, The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *Modeling, Identification and Control: A Norwegian Research Bulletin*, 2020, Vol. 41(4), pp. 241–295.
- [8] H.A. Mieg, System experts and decision making experts in transdisciplinary projects, *International Journal of Sustainability in Higher Education*, 2006, Vol. 7(3), pp. 341–351.
- [9] I. Winder, *Lego Logistics: Tangible Interactive Matrix Meets Last Mile Logistics Simulation — Interactive Simulation*. Massachusetts Institute of Technology, 2016, <https://ira.mit.edu/blog/lego-logistics>
- [10] S. Dimitrijević, J. Jovanović, V. Devedžić, A comparative study of software tools for user story management, *Information and Software Technology*, 2015, Vol. 57, pp. 352–368.
- [11] L. Pelegrin, B. Moser, S. Wanaka, M.-A. Chavy-Macdonald and I. Winder, Field Guide for Interpreting Engineering Team Behavior with Sensor Data. In E. Bonjour, D. Krob, L. Palladino, F. Stephan (eds.) *Complex Systems Design & Management, CSD&M 2018*, Springer, Cham, 2019, pp. 203–218.
- [12] I. Winder, D. Delaporte, S. Wanaka and K. Hiekata, Sensing Teamwork during Multi-objective Optimization. *IEEE World Forum on Internet of Things, WF-IoT 2020 - Symposium Proceedings*, 2020, pp. 1–6.
- [13] C.M. Rose, E. Saratsis, S. Aldawood, T. Dogan and C. Reinhart, A Tangible Interface for Collaborative Urban Design for Energy Efficiency, Daylighting, and Walkability. *Proceedings of the 14th Conference of International Building Performance Simulation Association*, 2015, <https://dspace.mit.edu/handle/1721.1/106597>, Accessed July, 1 2021.
- [14] J.I. Winder and K. Larson, (2017). Bits and Bricks Tangible Interactive Matrix for Real-time Computation and 3D Projection Mapping, *Future Technologies Conference (FTC) 2017*, 29-30 November 2017, Vancouver, Canada, pp. 1113-1116, https://saiconference.com/Downloads/FTC2017/Proceedings/160_Paper_469-Bits_and_Bricks.pdf, Accessed July, 1 2021.
- [15] P. Manandhar, K. Rong, K. Carroll, R. De Filippi, I. Winder, J. Dieffenbach and B. R. Moser, Sensing systemic awareness and performance of teams during model-based site design. *IEEE World Forum on Internet of Things, WF-IoT 2020 - Symposium Proceedings*, 2020, June 1, DOI: 10.1109/WF-IoT48130.2020.9221406.
- [16] J.F. Finn III, D. Vasquez, A. Starr, K. Kusina, K. Silvester, I. Winder, *Autonomous Vehicles and Their Impact on Real Estate*, Gensler Research Institute, 2018.