The *Unlimited Rulebook*
Architecting the Economy Mechanics of Games

O *Unlimited Rulebook*
Arquitetando as Mecânicas de Economia de Jogos

Wilson Kazuo Mizutani[1], **Fabio Kon (advisor)**[1]

[1]Departamento de Ciência da Computação — Universidade de São Paulo (IME-USP)
Caixa Postal 05508–090 — São Paulo — SP — Brazil

{kazuo,kon}@ime.usp.br

Abstract. *Economy mechanics involve resource management, combat, object interactions, and power progression systems in games. There is a lack of development tools to reduce their implementation cost while still being game- or genre-agnostic. The Unlimited Rulebook is a reference architecture designed to fulfill this purpose, especially for games that lean heavily on self-amending rules and continuous releases. Developed using a consolidated process for reference architectures, our solution effectively reduces implementation effort in the long run in exchange for a thoughtful upfront investment to design a game system.*

Keywords. *Digital games, software architecture, economy mechanics, reference architecture.*

1. Introduction

In game development, the creative process of designing mechanics is often at odds with the technical process of implementing those mechanics [Wang and Nordmark 2015]. When a designer comes up with new mechanics for the game, either the existing game has to be already developed with similar possibilities in mind or new features must be written to support that idea — each alternative with its own downsides. A standard solution the game industry has employed over the years is *software reuse*, a practice most evident in the widespread adoption of *game engines* [Gregory 2019]. Thanks to them, developers can often focus more on what makes their game unique, and that has allowed thousands of games to exist that would cost too much to write from scratch otherwise.

When it comes to game mechanics, however, some are much more easily reusable than others. For instance, one will find multiple options among libraries that implement physics mechanics [Gregory 2019], but will struggle to find a general-purpose implementation for tactical combat or farming simulation. In this thesis, we studied what types of mechanics are less reusable, why they are so, and how we can further promote the creative process of making games beyond what engines and other tools already provide.

1.1. Motivation

We found that **economy mechanics** [Adams and Dormans 2012] are among the hardest to design general-purpose, reusable implementations for, because of how idiosyncratic they are. This type of mechanics comprises mainly resource management, combat, object

interaction, and power progression systems. One of the key factors that reduce the opportunity for reusability in them is that they often include what we call **self-amending rules** — dynamic rules that overrule each other [Peter Suber 1982]. Though a single, general-purpose solution that is *flexible* enough to support self-amending rules while also being *reusable* might not always be practical, an *extensible* approach is something our research has shown to effectively reduce development effort.

1.2. Relevance

Economy mechanics challenge developers to find solutions unique to each game while keeping technical costs under control. This is especially true in game products that follow a continuous release model [Zhong and Xu 2021] since architectural decisions affect productivity for months or years. Our thesis formalizes this still sparsely explored domain and identifies an initial solution based on the state-of-the-art and state-of-the-practice.

2. Proposal

Besides engines and frameworks, there are other possible approaches to reduce the effort of implementing economy mechanics. Solutions found in both academic and industry literature include design patterns [Nystrom 2014], model-driven development [Zhu 2014], and software product lines [Furtado 2012]. The approach we chose in our thesis is a **reference architecture** [Nakagawa et al. 2011]. It consists of *reusable knowledge* (as opposed to reusable implementations) that provides a model to analyze economy mechanics and a series of guidelines to design architectures tailored to specific games based on that model. The project includes an associated reference implementation to illustrate the proposed solution — see Section 3.1. We named our reference architecture the ***Unlimited Rulebook***.

2.1. Originality

Compared to other approaches, this solution is unique as it provides a general-purpose tool that does not neglect the idiosyncrasies of economy mechanics. All other solutions in this domain that offer reusability are designed with only one particular genre in mind, such as the RPGMaker series of engines[1]. At the same time, more general solutions, like design or architectural patterns (e.g. Entity-Component-System [West 2018]), provide no more support for economy mechanics than they do for other types of mechanics.

2.2. Methodology

Our work followed the ProSA-RA method [Nakagawa et al. 2014], a four-step process for designing and evaluating reference architectures. The resulting solution provides two main tools for game architects. The first is a set of **domain concepts** that systematically map the requirements of a game onto a **reference model**. The second is a series of **architectural viewpoints** which, in turn, maps the specific model of each game into an architecture that best fits its needs in terms of economy mechanics implementation.

3. Results

Following the method above, we investigated software architectures applied to economy mechanics. That involved carrying out semi-structured interviews, performing a systematic literature review [Mizutani et al. 2021], and gathering information from multiple publications from academic sources and gray literature alike. We compiled that knowledge into ten domain concepts that served as the foundation for our design.

[1] `https://www.rpgmakerweb.com/` (last accessed July 23rd, 2022)

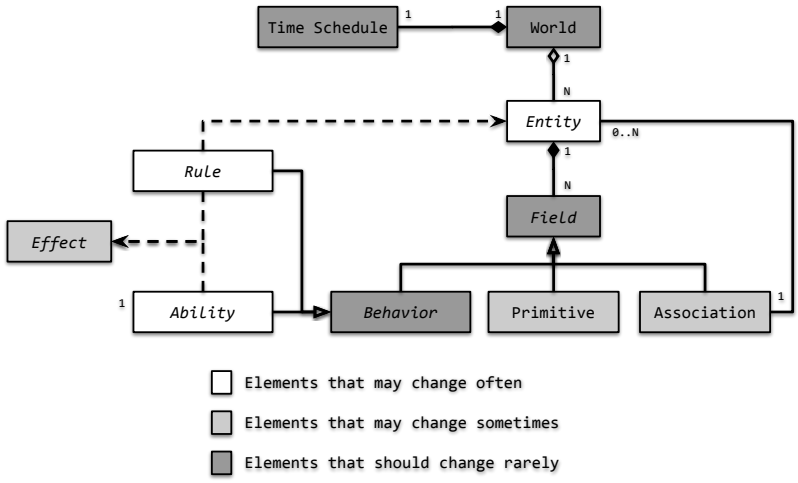

Figure 1. The mechanics model part of the reference model.

With that, we proposed a reference model for economy mechanics in games (partly shown in Figure 1) and detailed the three architectural viewpoints that make up the bulk of the *Unlimited Rulebook*. They are the crosscutting viewpoint, the runtime viewpoint (one of its diagrams can be seen in Figure 2), and the source code viewpoint. The design process of the *Unlimited Rulebook* was iterative and consisted of three cycles. At the end of each, we evaluated the reference architecture using two methods: systematically-conducted quasi-experiments [Campbell and Stanley 1963] and the development of proofs-of-concept in the form of both simplified simulators and actual games.

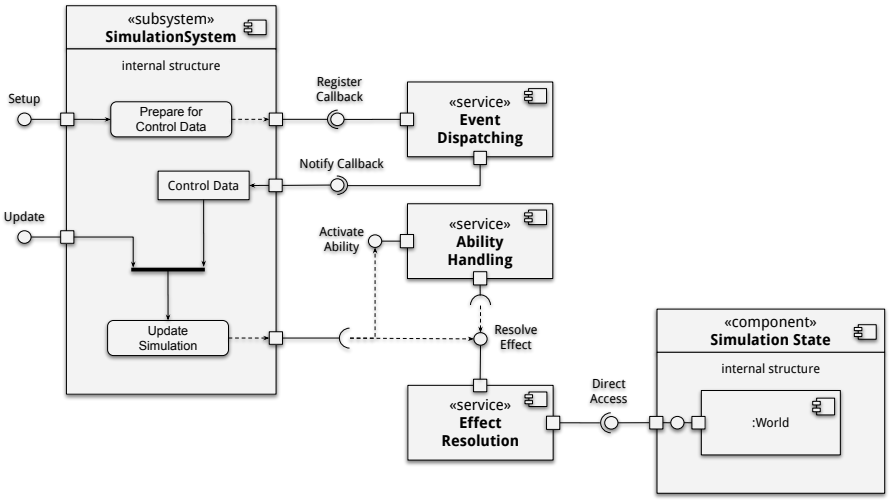

Figure 2. Part of the runtime viewpoint.

3.1. Byproducts

Our research produced three publications — two conference papers discussing preliminary results of the research [Mizutani and Kon 2019], [Mizutani and Kon 2020] and one article published in the Entertainment Computing journal, which details our systematic literature review [Mizutani et al. 2021]. A fourth publication is planned for submission to an internation journal within the next months. Also as part of our domain investigation, we compiled a dataset from our semi-structured interviews (available in Appendix

A of the thesis). In the evaluation steps, our results include the design for the quasi-experiments, the data collected from them, and two new game programming courses at University of São Paulo (during which the quasi-experiments took place). Furthermore, one of the proofs-of-concept was *Grimoire: Ars Bellica*, a small open-source game designed to evaluate the benefits of implementing self-amending rules with the *Unlimited Rulebook* and fulfill the role of reference implementation[2]. At the heart of the *Unlimited Rulebook*, one of the key byproducts of our research is the Ability-Effect-Rule design pattern, responsible for supporting self-amending rules in a flexible and extensible manner.

4. Conclusion

We reached a solution that reduces the cost of economy mechanics under the appropriate circumstances. Not all genres and games have a volume of economy mechanics large enough to compensate for the upfront cost of designing a specialized architecture using the *Unlimited Rulebook*. Among the ones that do, the most common genres include role-playing, strategy, management simulation, and 4X games. Software architects and developers make the most out of the *Unlimited Rulebook* when their games lean heavy on self-amending rules — like in card games or rogue-likes — and/or a continuous release model — such as competitive games, which must keep the gameplay fresh to retain players.

4.1. Main Challenges

One challenge in our research is that, while economy mechanics is a somewhat known term in the field of game design, it is practically unspoken in computer science applied to games. Thus, finding references in this domain required much more effort than more typical topics. Another challenge was evaluating the *Unlimited Rulebook*. Though our quasi-experiments were carried out as designed, the time available during a single college semester is not enough to reach the point where the benefits of our solution surpass its costs. The proof-of-concepts fared better in that regard, but are more biased since there is no randomness and all developers involved already had experience with the *Unlimited Rulebook*. We intend to further validate and improve the reference architecture as we see it used in more projects in both academia and the industry.

4.2. Contribution

We gathered evidence from both academic and gray literature to model and tackle the challenges of economy mechanics in software architecture. Given the previous lack of coordinated effort in this field, our thesis offers a formal and unprecedented starting point for researchers and practitioners alike. Through concepts such as the economy simulation state and rule adjudication services, we can now delve into this matter in a more systematic fashion. Our results also establish the limitations of empirical studies in academic contexts for this domain and suggest a number of improvements in future validation methods.

References

Adams, E. and Dormans, J. (2012). *Game Mechanics: Advanced Game Design*. New Riders.

[2]Playable at `https://kazuo256.itch.io/grimoire-ars-bellica`.

Campbell, D. T. and Stanley, J. C. (1963). *Handbook of research on teaching.*, chapter Experimental and Quasi-Experimental Designs for Research, pages 1–71.

Furtado, A. W. B. (2012). *Domain-Specific Game Development*. PhD thesis, Universidade Federal de Pernambuco.

Gregory, J. (2019). *Game engine architecture, third edition*. CRC Press.

Mizutani, W. K., K. Daros, V., and Kon, F. (2021). Software architecture for digital game mechanics: A systematic literature review. *Entertainment Computing*, 38:100421.

Mizutani, W. K. and Kon, F. (2019). Toward a reference architecture for economy mechanics in digital games. In *Proceedings of the Brazilian Symposium on Games and Digital Entertainment (SBGames)*, pages 623–626.

Mizutani, W. K. and Kon, F. (2020). Unlimited rulebook: a reference architecture for economy mechanics in digital games. In *Proceedings of the IEEE International Conference on Software Architecture (ICSA)*, pages 58–68.

Nakagawa, E. Y., Guessi, M., Maldonado, J. C., Feitosa, D., and Oquendo, F. (2014). Consolidating a process for the design, representation, and evaluation of reference architectures. In *Proceedings - Working IEEE/IFIP Conference on Software Architecture 2014, WICSA 2014*, pages 143–152.

Nakagawa, E. Y., Oliveira, P., and Becker, A. M. (2011). Reference Architecture and Product Line Architecture: A Comparison. In *European Conference on Software Architecture (ECSA)*, pages 2–5.

Nystrom, R. (2014). *Game Programming Patterns*. Genever Benning.

Peter Suber (1982). Nomic: A Game of Self-Amendment. Available online at http://legacy.earlham.edu/ peters/nomic.htm (last accessed July 23th, 2022).

Wang, A. I. and Nordmark, N. (2015). Software Architectures and the Creative Processes in Game Development. In *International Conference on Entertainment Computing*.

West, C. (2018). Using Rust For Game Development.

Zhong, X. and Xu, J. (2021). Game updates enhance players' engagement: a case of dota2. pages 117–123.

Zhu, M. (2014). *Model-Driven Game Development Addressing Architectural Diversity and Game Engine-Integration*. PhD thesis, Department of Computer and Information Science Faculty, Norwegian University of Science and Technology.