Bridging the Gap: Conceptual Modeling and Machine Learning for Web Portals

Dadhichi Shukla,
Eugen Lindorfer
STRG GmbH, Vienna, Austria
firstname.lastname@strg.at

Sebastian Eresheim,
Alexander Buchelt
UAS Saint Poelten, Austria
firstname.lastname@fhstp.ac.at

Abstract. In recent years e-commerce has become a significant part of the economy by turning into a multi-billion dollar industry. Modern web-tools and services enable business owners to create online portals within a short period. In most cases, the business owners design their online shops based on existing user journey templates offered by webpage service providers. As a result, it limits their analysis of the user journey of their shop portal prior to the launch (go online). It becomes extremely challenging to understand what the experience of the visitors following the template user journey could be. In this paper, we propose a system to leverage the fundamentals of conceptual modeling and reinforcement learning to model web portals and analyze their structure. We employ two reinforcement learning methods, namely, Q-learning and SARSA, to train agents and navigate in a simulated environment representing the web portal. The paper is an attempt at creating a bridge between conceptual modeling and reinforcement learning by taking an empirical study approach.

Keywords: Conceptual modeling, Reinforcement learning, Agents, Online Shop Simulation, Web Portals, Empirical Study

1 Introduction

The field of conceptual modeling has vastly advanced in understanding complex system by applying levels of abstraction. It facilitates fragmenting a system with complex functions in a way to make it human-interpretable. Despite the well-known merits associated with conceptual modeling (CM), it has yet to see extensive adoption in the field of machine learning (ML). Both of the domains have seen limited overlap with each other. It is perhaps due to the applicative nature of modern-day machine learning methodologies.

Today, machine learning is used in every aspect of our digital presence. Thanks to the vast amount of digital data available, the ML models can be trained to identify user patterns while browsing the net. The training data consists of user interactions such as tracking data, sharing pictures, likes and comments on articles, viewed videos, online shopping, etc. The ML models attempt to understand patterns of user behavior, primarily used for personalized content recommendation.

Works by Terragni et. al [17] and Sarirah et al. [14] propose approaches by apply process mining techniques to web server logs of user journey in order to investigate user behaviors, identify clusters of different users, and improve user journey. In light of online portals adhering to particular templates for the user journey (or sitemap) during webpage design, an interesting question arises: How can we analyze a website's user journey prior to its launch? Given that web server logs help identify various user behaviors, there is no-way to gather real user data prior to the launch. To address the question, we propose to employ reinforcement learning (RL), a subfield of machine learning, to analyze complex web portals, particularly online shopping portals, characterized by a multitude of interconnected pages.

We can utilize fundamentals of conceptual modelling to describe specific objectives to describe web portals. Successively, implement RL techniques, incorporating the objectives, and then performing an empirical study. We designed a simulated environment consisting of nodes and edges representing an online shopping portal. The RL agents were trained to navigate through the environment. One of the challenges in the analysis of web portals is identifying if there is a correlation with users online behavior and user journey. However, user's online behavior is complex in nature comprising various web actions. The forthcoming sections explain the ability of our current approach to emulate user behaviors.

Next, Section 2 describes various research efforts aiming to bridge the two vast domains of conceptual modeling and machine learning. Additionally, we highlight papers applying reinforcement learning methods for web-based challenges. In Section 3, we explain our framework to model web portals and how to apply reinforcement learning to the analysis of the portals. Section 4 illustrates results from our experiments, where an agent learns to navigate the simulated environment. Lastly, in Section 5 we present our concluding arguments, emphasizing how conceptual modeling and machine learning can complement each other.

2 Related Work

Maass et al. [7] provide an extensive study addressing how conceptual modeling and machine learning can support each other in the development of better solutions. Within the context of the proposed paper, we attempt to address the above by using virtual RL agents to navigate a simulated online store to emulate user behavior. The agent's behavior is established based on RL methods such as Q-learning [18] and SARSA (State-action-reward-state-action) algorithm [13].

Previous research has applied RL in online environments where agents operate on human-created artifacts. Shi et al. [16] presented the World of Bits (WoB) platform, where agents accomplish tasks on an online page by simulating actions such as key presses and mouse clicks. Other study proposed by Gur et al. [6] was along the same lines, where an agent learns to navigate a webpage. A RL agent learns to solve the tasks by mastering a set of primitive skills, such as filling up appropriate text in a form or deciding a date, and combining those to form and solve complex, compositional tasks such as booking an airline ticket, logging in to a website, etc. The key similarity of our work and the aforementioned studies is the modeling of web portals and having a RL agent to perform various tasks. It aligns with the fundamentals of: (i) conceptual modeling, by having a web portal model to describe aspects of the real-world for the purpose of simplifying its understanding, and (ii) reinforcement learning, by implementing an agent to perform real-world tasks, analyzes of empirical results, and drawing conclusions about the modeled web portals.

Often, it appears the fields of conceptual modeling and machine learning are weakly connected [8], however, numerous research works has exhibited otherwise. A RL-based tool called MORTAL proposed by Yuan et al. [19] to learn a relational schema by interacting with a relational database management system (RDBMS). In a recent study, Bork et al. [3] focused on systematic mapping between the two interdisciplinary fields i.e., conceptual modeling (CM) and artificial intelligence (AI), and the advantages of strategically utilizing one another. Reinhard et al. [12] propose a conceptual model for interactive labeling with the human-in-the-loop method using RL.

A study by Feltus et al. [5] investigated the role of AI in helping with domain conceptualization. The authors provide a review of symbolic and subsymbolic AI techniques. Reinforcement learning is categorized as a subsymbolic approach since the agent learns to act and react in an environment with the objective of solving a task by accumulating rewards. Conceptual modeling of web portals, intertwined with training RL agents to perform human-style tasks, enables a new way of analyzing user journeys on webpages.

3 Modeling

If we look at nature, we will observe animals explore and interact with their environment to survive by traveling around, gathering food, etc. Similarly, the paradigm of reinforcement learning captures an agent's behavior in the environment in search of a reward. Barreto et al. [1] provide an extensive explanation of model-based and model-free agents. The authors propose an algorithmic model that strikes a balance between model-free and model-based approaches, resembling human-like methods of describing the world and selecting strategies for interaction. Comparably, when it comes to web portals, we have to consider the human factor in the design of the user journeys of the web pages. Our work primarily focuses on online shopping portals.

Consider a retail business owner who wishes to start an e-commerce portal. While a physical store entails numerous expenses like rent and maintenance, its benefit lies in enabling firsthand customer experience through direct observation of their interaction with products. Take, for instance, a boutique clothing store. Despite the costs, the physical store permits customers to engage with fabrics, try on outfits, and evaluate their appearance in mirrors, offering a tactile shopping experience. Having the experience as a predicate, the owner can exercise changes in the shop, such as redesigning aisles, placement of mirrors, speedy customer service, enhancing the shop aesthetics, etc.

In comparison to a physical store, an online shop deploys existing user journeys. A user journey template is analogous to aisle design in a physical shop. Having an online shop enables the owner to scale the customer base far wider than the reach of the physical shop [2, 9]. Although the owner benefits from cost reduction and scalability by having an online shop, it is difficult to gauge the shopping experience of an online customer. Most web portals use an existing user journey template to create an online shop, thereby leaving minimal room for experimenting and understanding the customer experience. Unbeknownst of its merits and demerits, an existing user journey template is used frequently. Moreover, given today's tools, the owner can analyze a user journey only after the online shop goes online and not during the offline development phase. We attempt at solving this problem by having virtual RL agents, emulated as real users with various behaviors, generate empirical data prior to the launch of a website.

Conceptual modeling plays a crucial role in designing an empirical approach by providing a framework to represent the various components and objectives of the user journey. It aids in breaking down the complex system of an online shopping portal into manageable parts, making it easier to analyze and understand. Through conceptual modeling, we can identify the different stages and touchpoints in the user journey, such as product browsing, adding items to the cart, and completing a purchase. We propose to model web portals, consisting of webpages, as a graph structure, with a webpage as a node and edges representing all links accessible from the webpage. Modeling web portals with a graph structure provides a simplistic overview of all the connections among pages.

The graph can be exported in any of the standard formats, like XML, JSON, etc. We use the JSON format as it enables seamless integration with popular RL libraries [4,11]. The ease of integration with RL-libraries facilitates the process of experimenting with new environments into the training pipeline, making experimentation more efficient. The JSON-based RL environment enables the creation of diverse and customizable scenarios for training an agent. Overall, a structure consisting of nodes and edges, can be exported from all kinds of web pages, especially e-commerce shopping portals where each node consists of elements such as add to cart, review, like, comment, favorites, etc.

Reinforcement learning, allows us to train an agent to make decisions and take actions within the online shopping environment. The agent's objective is to maximize a reward, defined according to desired outcomes. The RL agent learns through trial and error, adjusting its actions based on the observed rewards and the environment's responses. In the context of analyzing an online shopping portal, we use reinforcement learning as an empirical approach, where a user is modeled as an agent and the online shopping portal as the environment.

By combining the principles of reinforcement learning and conceptual modeling, we can simulate user behaviors, study the actions of an agent on a user journey, and eventually optimize the web portals. To implement the empirical approach towards conceptual modeling of online portals and to put reinforcement learning to use, we propose the following steps:

1. **Conceptual Model**: Develop a conceptual model to represent the user journey of the online shopping portal. The model should capture the different stages, touchpoints, and possible actions a user can take. It may include components such as product discovery, search functionality, product details, reviews, cart management, and checkout.
2. **State Representation**: Define the states of the environment where an RL agent can observe and interact within the conceptual model. States are represented by variables to capture relevant information, such as the current page, items in the cart, search history, time allowed for shopping, amount of money to spend, or user

preferences. The state representation should reflect the context of a task and enable the agent to make informed decisions.

3. **Action Space**: Define the possible actions the agent can take in each state. Actions can include browsing different products, adding items to the cart, applying filters, using search functionalities, or proceeding to checkout. The action space of the agent should encompass a range of choices reflecting interactions by real-world users in the online shopping portal.
4. **Reward Design**: Design a reward function providing feedback to the agent based on its actions. The reward function can consider metrics such as successful purchases, time and money spent on the portal, etc. Its role is to incentivize actions leading to desired task outcomes and discourage actions diverging from task completion.
5. **Training and Evaluation**: To train the agent based on the specified conceptual model, use RL algorithms such as Q-learning, SARSA, Deep Q-Networks (DQN) [10], Proximal Policy Optimization (PPO) [15], etc. The agent interacts with the environment by selecting actions based on its current state, receiving rewards, and updating its policy to improve future decision-making. Iteratively exploring the state space and adjusting the agent's behavior are part of the training process.
6. **Empirical Study**: Conduct experiments using the trained agent to simulate user behaviors and study their impact on the online shopping experience. Vary the agent's behavior by adjusting parameters related to the state space, reward design, and training RL model. Evaluate the effectiveness of different user journey variations.
7. **Analysis and Optimization**: Analyze the empirical study results to gain insights into the relationship between the user journey and user behavior. Identify patterns, trends, and influential factors contributing to positive user experiences and desired outcomes. The results would provide insights into suboptimal user journey designs, diminishing the conversion rate of a shopping portal. Use these insights to optimize the conceptual model and refine the user journey design, improving the online shopping portal's performance and user satisfaction.

By combining the empirical approach of reinforcement learning with the principles of conceptual modeling, we can iteratively improve the user journey design in an online shopping portal. This method facilitates data-driven decision-making and optimization, leading to enhanced user experiences, increased conversion rates, and improved business outcomes, where the data is generated by RL agents simulated as real users instead of using the data from a real user. Moreover, the method also provides a framework for understanding the complex dynamics of user behavior and enables targeted interventions to guide users towards desired actions within the online shopping environment free from collecting real users web log data.

4 Experiments and Results

We simulated an online shopping portal consisting of webpages such as "jeans", "shoes", and "shirts". The simulated shopping portal starts with the *homepage* node, with edges connected to three level-1 nodes (or main category): *bestsellers*, *the essentials*, *trending*. Each level-1 node has three level-2 nodes namely *jeans*, *shoes*, and *shirts*, also called as wardrobe capsules. Lastly, each wardrobe capsule has two articles or level-3 nodes, for example, the *shoes* capsule consists of *loafers* and *trainers*. A partial view of the simulated shopping portal modeled as a JSON structure is described in listing 1.1, where ellipsis (...) indicate the rest of the nodes.

```
{
  "nodes":
  [
    {
      "node_label": "homepage",
      "product_page": "False",
      "nodes_list": ["homepage", "bestsellers",
      "the essentials", "trending"],
      "action_elems": ["jump_to"]

    },
    {
      "node_label": "the essentials",
      "product_page": "False",
      "nodes_list": ["homepage", "jeans", "shoes", "shirts"],
      "action_elems": ["jump_to"]
    },
    {
      "node_label": "shoes",
      "product_page": "False",
      "nodes_list": ["homepage", "loafers", "trainers"],
      "action_elems": ["jump_to"]
    },
    {
      "node_label": "loafers",
      "product_page": "True",
      "nodes_list": ["homepage", "shoes"],
      "action_elems": ["jump_to"]
    },
    {
      "node_label": "trainers",
      "product_page": "True",
      "nodes_list": ["homepage", "shoes"],
      "action_elems": ["jump_to"]
    },
    {
      "...": "..."
    }
  ]
  }
```

Listing 1.1: Shopping portal JSON representation.

In total there are 31 nodes modeled as the environment for the RL agent. All the nodes have an edge connecting to the *homepage* node. Each node consists of four keys:

1. **node_label**: Name of a node.

2. **product_page**: To identify whether a node consists of a product/article to be bought.
3. **nodes_list**: List of connected nodes, namely, nodes an agent can jump to.
4. **action_elems**: List of actions an agent can perform.

Currently, the agent can execute only one type of action, i.e., *jump_to*, to go to one of the connected nodes. We are working towards incorporating other types of action such as *add_to_cart*, *review*, *like*, *comment*, etc. Introducing additional types of actions would also involve a substantial action space, manageable through reinforcement learning methods like DQN, PPO, etc.

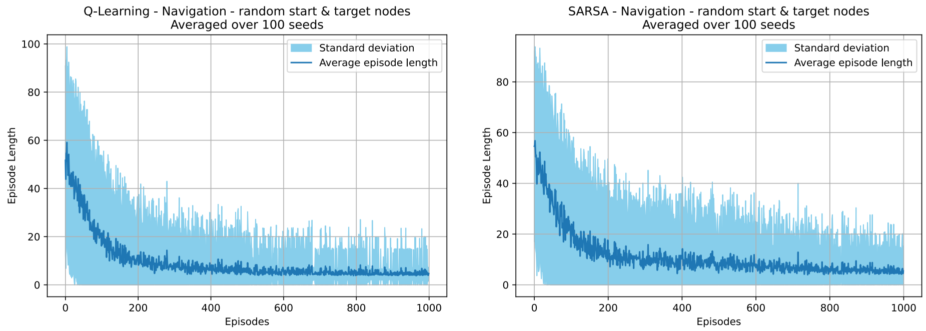
The state space of the agent is a two-dimensional tuple consisting of discrete values. The first dimension represents the node label where the agent is currently present, whereas the second dimension tells the agent where it is supposed to go. The total number of states is computed by N^s; where N is the number of nodes and s is the dimension of the state space. Therefore, in our particular shopping environment, there are $31^2 = 961$ states.

The action space of the agent is a one-dimensional discrete value, representing the node label where the agent decides to *jump_to*. Since the environment consists of 31 nodes, the action space consists of 31 actions. However, not all actions are permitted in each state. The number of actions the agent can choose from is based on the actions stated in *action_elems* and *nodes_list* of the node where the agent is currently present.

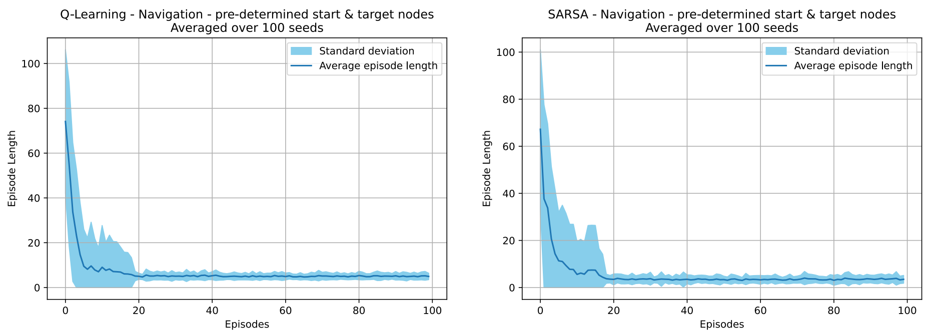
In the navigation task, the agent is provided with a node label where it has to start and find its way to a target node. It selects one of the nodes from the *nodes_list* of its current node using ε-greedy method and executes the *jump_to* action. The selection of the start and target nodes was made in two ways: (i) **randomized**: the start and the target node are randomly selected, and (ii) **pre-determined**: the start and the target nodes are selected manually ensuring a minimum of 2 edges between the nodes. The reward function of the agent is defined as follows: the agent receives a reward of (i) **1** if the targeted node is reached, (ii) **-1** if the maximum amount of steps during a training episode is reached, or (iii) **0** when an agent jumps to a node other than the targeted node.

The results for the navigation task are shown in fig 1. The y-axis of the plots represents the length of an episode, i.e., the number of steps the agent takes to navigate to the target node, whereas the x-axis is the number of episodes i.e., the number iterations to train an agent. The dark blue line shows the average episode length per episode and the light blue background area shows the standard deviation, over 100 different seeds. Figure 1a and fig. 1b illustrate results from randomized and pre-determined start and target nodes, respectively. A noticeable difference between the two node selections ways is the number of episodes required by an agent to converge. The randomized method begins to converge at around 650 episodes, whereas the pre-determined method requires less than 20 episodes to reach the target node. Both the RL approaches continue to drop their episode length of the training process as the number of episodes increase.

If we observe closely the result of the pre-determined nodes fig. 1b, the SARSA agent converges comparably faster with respect to Q-learning. In other words, SARSA helps the agent solve the navigation task relatively quickly. When dealing with randomized nodes, SARSA exhibits slower convergence in contrast to Q-learning, as a result of less knowledge gained during each iteration due to its non-exploratory nature.



(a) Agent navigating from a randomly selected start node and a target node. **(l)** Agent trained with Q-learning, **(r)** Agent trained with SARSA.



(b) Agent navigating from a pre-determined manually selected start node and a target node. **(l)** Agent trained with Q-learning, **(r)** Agent trained with SARSA.

Fig. 1: Comparing Q-learning with SARSA for the navigation task. Dark blue line indicates average episode length and light blue background shows standard deviation over these episodes each with 100 seeds.

One can deduce, through observation, an agent using the SARSA approach displays risk-averse tendencies or is focussed, consequently limiting its exploratory behavior. The Q-learning agent, however, demonstrates risk-taking and exploratory behavior. Although the differences in the convergence values are comparably close, they will eventually be significant with addition of newer nodes, edges, and agent actions to the environment.

5 Conclusion and Future Work

Given the extensive amount of research in the fields of conceptual modeling and machine learning, it is reasonable to expect they would complement each other. We addressed creating a bridge between the two domains by taking an empirical approach due to the applicative nature of machine learning, especially reinforcement learning.

We chose the problem of analyzing whether user journeys of online portals, specially the ones yet to be launched, engender specific user behaviors online. Using a JSON structure to model web portals in our conceptual modeling phase undeniably facilitates easy addition and removal of nodes, connections among the nodes, and available actions for an agent on a node. The simplicity and expressiveness of the environment as JSON, makes the graph's topology, node attributes, and edge properties a step towards AI explainability.

Expanding upon the work presented in this paper, we intend to introduce a broader range of types of action an agent could perform at each node. Consequently, we would have to exercise sophisticated learning methods such as DQN, PPO, masked-PPO, etc., due to their ability to handle large action space. By employing various RL methods and changing reward functions, we could potentially simulate different types of complex user behaviors on a web portal. Consequently, the analysis of user journeys of a web portal can be completely simulated to identify emphasizing and de-emphasizing segments of the portal. The integration of empirical methods and conceptual modeling with machine learning not only enhances the validity of the models but also facilitates a deeper understanding of the underlying processes. In conclusion, empirically-informed conceptual modeling practice in conjunction with reinforcement learning represents a powerful and rigorous approach to understanding complex systems such as web portals.

6 Acknowledgement

The research leading to these results are produced under the STRG.Agents project funded by the Austrian Research Promotion Agency (Die Österreichische Forschungsförderungsgesellschaft FFG) under the grant agreement number 45196538 and the FFG number 898104.

References

1. Barreto, A., Hou, S., Borsa, D., Silver, D., Precup, D.: Fast reinforcement learning with generalized policy updates. Proceedings of the National Academy of Sciences (2020)
2. Bjerkan, K.Y., Bjørgen, A., Hjelkrem, O.A.: E-commerce and prevalence of last mile practices. Transportation Research Procedia (2020)
3. Bork, D., Ali, S.J., Roelens, B.: Conceptual modeling and artificial intelligence: A systematic mapping study. arXiv preprint arXiv:2303.06758 (2023)
4. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym (2016)
5. Feltus, C., Ma, Q., Proper, H.A., Kelsen, P.: Towards ai assisted domain modeling. In: Advances in Conceptual Modeling: ER 2021 Workshops CoMoNoS, EmpER (2021)
6. Gur, I., Jaques, N., Miao, Y., Choi, J., Tiwari, M., Lee, H., Faust, A.: Environment generation for zero-shot compositional reinforcement learning. Advances in Neural Information Processing Systems (2021)
7. Maass, W., Storey, V.C.: Pairing conceptual modeling with machine learning. Data & Knowledge Engineering (2021)
8. Maass, W., Storey, V.C.: Why should machine learning require conceptual models? (2021)

9. Maat, K., Konings, R.: Accessibility or innovation? store shopping trips versus online shopping. Transportation Research Record (2018)
10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature (2015)
11. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research (2021)
12. Reinhard, P., Li, M.M., Dickhaut, E., Reh, C., Peters, C., Leimeister, J.M.: A conceptual model for labeling in reinforcement learning systems: A value co-creation perspective. In: International Conference on Design Science Research in Information Systems and Technology (2023)
13. Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. University of Cambridge, Department of Engineering Cambridge, UK (1994)
14. Sarirah Husin, H., Ismail, S.: Process mining approach to analyze user navigation behavior of a news website. In: Proceedings of the 4th International Conference on Information Science and Systems (2021)
15. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
16. Shi, T., Karpathy, A., Fan, L., Hernandez, J., Liang, P.: World of bits: An open-domain platform for web-based agents. In: International Conference on Machine Learning (2017)
17. Terragni, A., Hassani, M.: Analyzing customer journey with process mining: From discovery to recommendations. In: 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud) (2018)
18. Watkins, C.J., Dayan, P.: Q-learning. Machine learning (1992)
19. Yuan, G., Lu, J.: MORTAL: A tool of automatically designing relational storage schemas for multi-model data through reinforcement learning. In: Proceedings of the ER Demos and Posters 2021, International Conference on Conceptual Modeling (ER 2021) (2021)