Playing with Recipes: Encouraging Exploration of a Model of Taste

Johnathan Pagnutti
University of California, Santa Cruz
jpagnutt@ucsc.edu

Jim Whitehead
University of California, Santa Cruz
ejw@ucsc.edu

ABSTRACT

The sense of taste is a highly complex phenomena, and has not been tackled by recipe generators. However, the techniques used in modeling taste and generating recipes can inform a class of playful interfaces that help users explore, experiment with and gain mastery over the manipulation of highly complex models. This abstract introduces an architecture for a model of taste that users interact with by playing with and rating recipes generated from the model.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems.

Keywords

taste, content generation, computational creativity

1. INTRODUCTION

Several domains have been explored for Procedural Content Generation (PCG)[21] (sometimes referred to as generative methods), however there are still plenty of open domains that can benefit from content generation. Creativity in the culinary arts has been linked to creativity in scientific domains[3], and so techniques developed to help encourage exploration and experimentation in the culinary space can be mapped back over to the scientific space. In addition, people often feel intimidated and afraid of cooking[6]. Mixed initiative systems are useful for creating constructivist learning environments[11], which are built around experimentation on ill-structured problems (such as the way to bake a perfect cookie). Games like PicBreeder[17] present an easy to explore spaces that people feel comfortable experimenting in. This sort of game has come out of an exciting interdisciplinary space between Computational Creativity (CC) and PCG. The result of this trend is that design guidelines exist for not only creating content generators, but also designing games that allow players to explore the creative potential of those generators. We propose using an interdisciplinary framework in order to develop a computational model of taste, a generator based on this model and an interface that allows for users to explore that space and be creative.

Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015), June 22-25, 2015, Pacific Grove, CA, USA. ISBN 978-0-9913982-4-9.

2. PAST WORK

PIERRE[13] uses a genetic algorithm to generate crock pot recipes from a corpus gathered from various websites. Case-based AI planners have also generated recipes[8, 9]. None of these generators are built from a model of taste; all of them only consider other recipes. There have also been several mixed initiative recipe generators that construct a knowledge base form a large corpus of recipes, then work with a user to create new dishes[14, 15]. The authors have experience in recipe generation, having designed a cocktail recipe generator. That work helped highlight the need for a more comprehensive generator and informed much of the proposed goals and design.

3. RESEARCH GOALS

The research goals of this work are threefold- build a computational model of taste, build a generator on top of that model, and build an interface that encourages exploration and experimentation with the model. The first goal is the most ambitious; thus far, no computational model of personal taste has been created. Investigations into taste have shown it to be a highly complex phenomena[18, 12, 7]. Parts of taste have tractable models: past work on flavor shows trends in flavor compounds and ingredient selection[1], and these trends have informed an evaluator[22]. Analytics have been performed about the mouthfeel and texture of various chemical compounds (such as in [23]), but these analytics have yet to be incorporated in a model or generator.

The next research goal is designing a generator to create recipes based on the taste model. It is important that the generator exist in dialogue with the model: if we want the generator to be creative, then it needs to be able to evaluate artifacts and react to them[20], as in figure 1. This critique is important, because it changes how we present the generated recipe to the user, as well as allowing our model to update itself in order to incorporate the critique.

Finally, this abstract proposes an interface over the generator that allows people to explore and experiment with recipes. Generator control and generative space exploration are good reasons to use PCG in games[19]. The entire project can be considered a simulation of experimenting in a kitchen, while providing warnings to users when they have

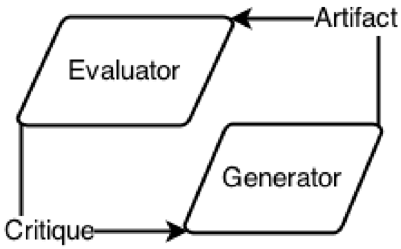

Figure 1: Generators create artifacts that are passed to an evaluator. The evaluator generates a critique that changes how the generator creates new artifacts.

deviated from what the model predicts will be successful. This is similar to games like Stellar that model a process and allow players to play with it[4]. The generator and model need to be flexible enough so that its recommendations can be overridden. This also makes the project similar to autotelic creativity games, like the Spore Creature Creator or PicBreeder. Much like those games (or parts of games), the act of modifying and creating new recipes exists for its own sake, because experimenting with food is fun. There is just a high chance that participants will learn a little bit about cooking along the way.

4. PROPOSED APPROACH

The architecture of the software system are a series of layers, with the model sitting at the bottom, the generator in the middle and the interface at the top, as seen in figure 2. Recipes will be broken into two parts: the ingredient list and the preparation steps. Each ingredient can be broken down into flavor compounds based on known flavor databases. Past work has had success in finding trends in cooking by treating flavors as linear combinations of flavor elements[2], and we continue the use of that model here. By simulating how these compounds react when mixed and how they transform as various preparation procedures are applied to them, we can derive a vector of features present in the final dish. Recipes should come from some publicly available database, which will allow us to assign a cook time, number of ingredients, and other relevant scores. By starting with some initial set of scored recipes, we can estimate scores for new recipes based on the features of what the model has already scored. We can also figure out which features are important to a high quality/novelty recipe and which are not, then weight them accordingly.

The scores themselves are under pressure for some weighting, as some scores will be more important to some users than others. For example, some users may not really care too much about novelty, but really are looking for a recipe that they can make quickly and that they're familiar with. Other users may want to take an entire day in the kitchen working on something that is good and novel but requires them to use every single bowl they own. Also, users are different people– what one person feels is a novel concept, another might have seen already. There might be parts of the ingredient or preparation space that someone never wants to see, because they have a food allergy and/or do not feel comfortable attempting to flambé something.

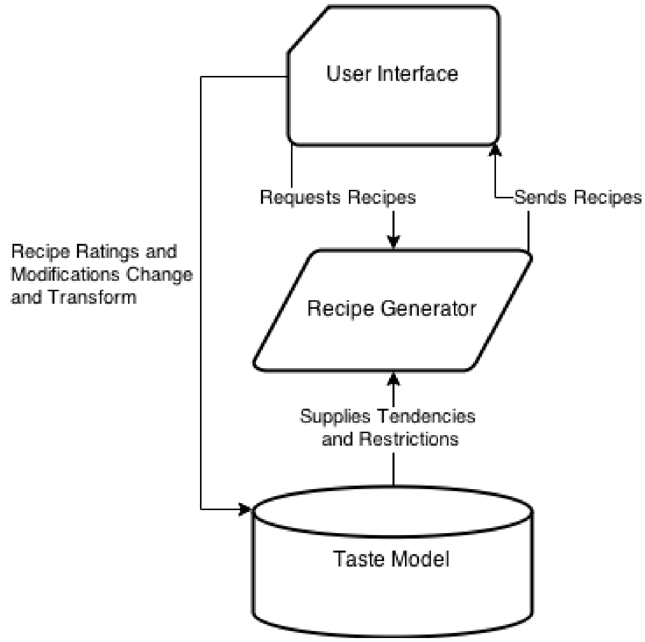

Figure 2: Generators create artifacts that are passed to an evaluator. The evaluator generates a critique that changes how the generator creates new artifacts.

These requirements indicate a bounded genetic algorithm (GA), by using logic programming to identify the bounds for the GA. The constraints for the logic program may be set by a user (said user is allergic to peanuts or doesn't want to flambé) or follow a rule of thumb for cooking. The fact that a GA can be loaded with many different heuristics to optimize for all kinds of different qualities allows for the flexibly needed to appeal to many different users. The bounds can be considered as the properties that a solution (in this case, recipe) must or must not have in order to be valid. The model evaluates each generated recipe, returning a relevant set of scores. As someone interacts more with the system, these novelty and quality scores will be more and more aligned with their own personal likes and dislikes.

We want to encourage the concept of exploring a vast world of recipes, while allowing the player control over the generator. Direct manipulation is key- players will not understand how actions impact recipes in an overly abstract interface. When the player gets close to the type of recipes they are looking for, they can drill in and see a more detailed view of the recipe. This view gives finer information (such as a more precise nutrition information), and allows for the recipe to be tweaked. Tweaking updates the recipes scores, which may change how the generator feels about a particular recipe. After making the dish (and tasting it, hopefully), they can respond back to the system and rate the recipe. This affects the model, as the score re-triggers the machine learning step and rebuilds the entire model, now incorporating the new data point. This approach lends itself towards machine learning, as new examples need to tweak how the model behaves.

This sort of process tends to be its own reward– coming up with successful substitutions or tweaking flavors on a dish is, in and of itself, a playful activity that allows a user to feel ownership over the result[5]. Games, such as [16], attempt to map creative exploration of a possibility space into formal game mechanics. In addition, "idea generation" games for game developers offer a number of design patterns and concepts where the 'win condition' is the generation of a good idea [10]. By analyzing games such as these, we aim to also develop a 'safe sandbox' where users can be creative with recipes.

5. EXPECTED ACHIEVEMENTS

The first expected achievement is building a flexible computational model of taste that can reflect personal inputs into the evaluation process. The use of logic programing to identify GA bounds is novel. This is a large scale AI project that combines techniques from both the CC community and the PCG community, especially those in PCG interested in mixed initiative interfaces. This project tries to present cooking in a more accessible way and foresees its user base as people who are interested in learning how to cook and people who already like to cook, but do not yet feel confident enough to start working entirely on their own.

6. CONCLUSION

By combining an understanding of taste with machine learning techniques, we can develop systems that have the benefit of both. We can tweak a model to take into account fuzzy influences, such as culture or personal taste while also showing our work when we do understand the underlying processes, such as the chemical reactions that take place while baking. This mixed approach allows for the development of flexible content generators and design interfaces that let users explore and understand deeply complex phenomena and processes.

7. REFERENCES

[1] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1, 2011.

[2] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási. Flavor network and the principles of food pairing. *Scientific reports*, 1, 2011.

[3] S. H. Carson, J. B. Peterson, and D. M. Higgins. Reliability, validity, and factor structure of the creative achievement questionnaire. *Creativity Research Journal*, 17(1):37–50, 2005.

[4] K. Compton and A. Grow. Stellar. *http://www.stellarsupernova.com/*.

[5] M. Daniel, Y. Guttmann, and A. Raviv. Cooking and maslow's hierarchy of needs: A qualitative analysis of amateur chefs' perspectives. *International Journal of Humanities and Social Science*, 1(20):86–94, 2011.

[6] R. Engler-Stringer. Food, cooking skills, and health: a literature review. *Canadian journal of Dietetic practice and research*, 71(3):141–145, 2010.

[7] J.-X. Guinard and R. Mazzucchelli. The sensory perception of texture and mouthfeel. *Trends in Food Science & Technology*, 7(7):213–219, 1996.

[8] K. J. Hammond. Chef: A model of case-based planning. In *AAAI*, pages 267–271, 1986.

[9] T. R. Hinrichs. *Problem solving in open worlds: A case study in design*. Lawrence Erlbaum Hillsdale, NJ, 1992.

[10] A. Kultima, J. Niemelä, J. Paavilainen, and H. Saarenpää. Designing game idea generation games. In *Proceedings of the 2008 conference on future play: Research, play, share*, pages 137–144. ACM, 2008.

[11] J. C. Lester, B. A. Stone, and G. D. Stelling. Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. In *Computational Models of Mixed-Initiative Interaction*, pages 185–228. Springer, 1999.

[12] J. Lim and M. B. Johnson. The role of congruency in retronasal odor referral to the mouth. *Chemical senses*, 37(6):515–522, 2012.

[13] R. G. Morris, S. H. Burton, P. M. Bodily, and D. Ventura. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity*, pages 119–125, 2012.

[14] F. Pinel and L. R. Varshney. Computational creativity for culinary recipes. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 439–442. ACM, 2014.

[15] F. Pinel, L. R. Varshney, and D. Bhattacharjya. A culinary computational creativity system. In *Computational Creativity Research: Towards Creative Machines*, pages 327–346. Springer, 2015.

[16] J. P. Rosenstock. Free play meets gameplay: igotband, a video game for improvisers. *Leonardo Music Journal*, 20:11–12, 2010.

[17] J. Secretan, N. Beato, D. B. D Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1759–1768. ACM, 2008.

[18] G. M. Shepherd. Smell images and the flavour system in the human brain. *Nature*, 444(7117):316–321, 2006.

[19] G. Smith. Understanding procedural content generation: a design-centric analysis of the role of pcg in games. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 917–926. ACM, 2014.

[20] D. Stokes. Minimally creative thought. *Metaphilosophy*, 42(5):658–681, 2011.

[21] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):172–186, 2011.

[22] K. R. Varshney, L. R. Varshney, J. Wang, and D. Myers. Flavor pairing in medieval european cuisine: A study in cooking with dirty data. *arXiv preprint arXiv:1307.7982*, 2013.

[23] S. Vidal, L. Francis, A. Noble, M. Kwiatkowski, V. Cheynier, and E. Waters. Taste and mouth-feel properties of different types of tannin-like polyphenolic compounds and anthocyanins in wine. *Analytica Chimica Acta*, 513(1):57–65, 2004.