

CS 279: Final Project Proposal

MACHINE-ASSISTED COOKING

Miriam Cha, Michelle Deng, and Melih Elibol

SUMMARY

Cooking can be a source of enjoyment and pride and a forum for creative experimentation, and compared to prepackaged and most restaurant meals, homemade meals are both healthier and more economical. Unfortunately, cooking takes time, and the tools, ingredient balance, motor techniques, and use of heat and cold, pressures of precise timing, and many other factors may seem overwhelming to novices. Particularly in a world where people are on the run and prepackaged meals, restaurants, and near-instant food delivery abound, it seems that many people perceive cooking as too stressful and time-consuming and simply not worth the effort to learn.

Few resources ease the cognitive load well. We find currently available printed cookbooks too cumbersome and verbose. There are many cooking blogs (e.g., [1]-[2]) that share recipes to users but these websites are merely a digital version of printed cookbooks. A few developers attempted to address the issue by developing a cellphone application [3] with interactive interfaces, but the app is hard to read and certainly not designed to be used while handling stoves and ingredients.

Hence in this project, we want to design a digital cooking assistant for novice cooks that achieves the following:

- Reduces the cognitive stress of coordinating multiple sensory inputs and motor tasks under time constraints
- Makes instructions in recipes more accessible
- Helps the user learn cooking techniques

Potential features of our design include timers for managing multiple simultaneous tasks; reminders to coordinate oven and stove temperatures; easy-to-read interfaces that display one step at a time with large pictorial, audio, and/or video aids, somewhat reminiscent of turn-by-turn navigation of a GPS; and lifelines to contact other users who have previously tried the recipe.

We will base our approach from user-centered design in HCI research. We first plan to survey both novice and expert cooks to identify key differences in the levels of cooking expertise—particularly those that seem “blockers” for novices. As such, our survey will include what novice cooks find hard, what prevents them from cooking more often, and what motivates experts to cook.

After identifying key issues from the preliminary survey, we will iterate through several cycles of prototyping and user studies and questionnaires or interviews to create our design. We will begin with multiple quick paper prototypes to gain early insights into the design before we invest time into developing a digital design. After couple rounds of iterating the paper prototypes, we will start implementing the digital design, leveraging the off-the-shelf recipe API BigOven [4]. To assess the digital prototypes, we will compare user performance on recipes with and without our prototypes and observe the effects on surveyed user ratings for mental demand, physical demand, temporal demand, performance, effort, and frustration using NASA Task Load Index. Ultimately, through repeated rounds of usability testing, we hope to arrive at a user-friendly cooking interface that achieves our stated goals.

MILESTONES

M1: Related Work / Surveys / "Assistant" (October 28th)

- Establish relevant related work:
 - Concrete related work: Actual applications that attempt to solve the issues our application seeks to solve.
 - Related research: HCI research related to solving problems related to distribution of cognition (e.g. papers leading up to inventions such as vehicle GPS devices).
- Establish Surveys
 - Survey for people who don't cook to help identify why people don't cook.
 - Survey for people who do cook to help identify what they enjoy about cooking.
 - Survey/Interview for "expert" cooks (people who take cooking seriously). We all seem to know at least one person that falls under this category. Our goal is to establish what questions to ask experts, to help inform design process.
 - Examples:
 - Identify "choking" points for novices and key differences between levels of expertise
 - How do experts think when they balance all the things they have to do?
 - What do novices find hard; what prevents people from cooking more?
- Explore potential "assistant" role of a machine
 - Identify how work load is distributed among people who cook in pairs; this should help inform the role, if any, our application should take as an assistant. For instance, when people cook in pairs, one person may keep track of what has been done thus far and what needs to be done next, while another executes.
- Establish initial code base

- Establish git repo / project folder structure.
 - Server-side
 - See if anyone has already written some code that can connect to BigOven via node. If nothing already exists, implement basic API calls to establish data-structures given by BigOven API.
 - Client-side (thinking tablet given "home use"/life balance/lightweight/nice screen)
 - Investigate media related needs that may arise, e.g. text-to-speech functionality, dynamically loaded images (via API).
 - If we are god-like in our programming abilities, we may also look into streaming video from another source (the API does not support video).

M2: Synthesize Results (November 4th)

- Synthesize Results
 - Surveys have been conducted.
 - Results will be synthesized into multiple paper-based designs (group parallel prototyping?).
- Feedback
 - Get feedback on paper-based designs, interview-style.
 - Use feedback to inform design changes.
 - Decide which design features from each paper-based design the final design should possess.
- Final Paper Design
 - Use feedback to establish final paper-based design.
- Implement API
 - Implement server-side API endpoint that carries out requests to BigOven on behalf of application client.
 - Implement client-side event-based async service which can invoke API calls to our data server. The service should make the call, serialize the response data structure into native Java data structures, and dispatch an event containing the serialized response.

M3: Implementation (November 11th)

- Implement the front-end design
 - This should be done in terms of view-based code. View logic should be decoupled from model, so that future design iterations are straight forward.
- Implement any necessary back-end code
 - If we establish that additional functionality in addition to the BigOven API is needed. This will be done using a basic nosql db (Mongo, we know it and it's good for rapid prototyping), and node/expressjs (so long as everyone is comfortable with node as a backend).

- Whoever implements the back-end is responsible for implementing client service calls to our API.
- This may turn out to be necessary for basic logging purposes.

M4: Conduct Lab-based Study (November 18th)

- Conduct user study
 - Within subjects (perhaps not as effective as between, but will consider learning effects)
- Possible design conditions
 - Person a: cookies with app, cake without app
 - Person b: cake with app, cookies without app

M5: Finalize (November 25th)

- Finalized Design based on results from user study.
- Implement final design changes.
- Finalize paper.

REFERENCES

- [1] <http://www.cookingclassy.com/>
- [2] <http://allrecipes.com/>
- [3] <https://play.google.com/store/apps/details?id=com.quasarcomputing.cookassistantfree.activities>
- [4] <http://api.bigoven.com/>