

Flamework Framework: レシピデータに基づき 火力を制御するフレームワークの試作

高田一真^{†1} 金子翔麻^{†1} 鈴木涼太^{†2} 神山洋一^{†2} 相澤裕貴^{†1} 大内泰良^{†1}
西條瞳^{†1} 高瀬理奈^{†1} 本田健人^{†1} 渡邊恵太^{†1†2}

概要 : IoT 技術の普及に伴い, キッチンにもスマート調理器具と呼ばれる IoT デバイスが導入され始めた. しかし, それぞれ違うレシピデータの形式に基づいて調理するため, 複数の機器を使って1つの料理を実現するのは難しい. そこで本研究では, デジタル化されたレシピと直接連携するスマート調理器具を提案する. そして本論文では調理工程の要素の1つである火力に着目し, レシピデータに基づいて火力を制御するフレームワーク Flamework Framework を提案する. 本論文では Flamework Framework の試作を通じ, 料理体験の変化について考察する.

1. はじめに

IoT (Internet of Things) 技術の普及に伴い, 家庭のキッチンにスマート調理器具と呼ばれるIoTデバイスが導入され始めた. 例えば, 肉や野菜など様々な食品を最適な温度に調理するcinder[1]がある. また, レシピデータに基づいて体積が自動で変化する計量スプーンsmoon[2]がある. しかし, これらのIoT調理機器は, それぞれ違うレシピデータの形式に基づいて調理するため, 複数の機器を使って1つの料理を実現するのは難しい.

そこで我々は, IoT調理機器が連携し1つの料理を実現する共通のレシピデータ形式やデバイスについて議論するFabFoodProjectを立ち上げ, 研究を進めている[3]. 本論文では, 調理工程の要素の1つである火力に着目し, レシピのデータに基づいて火力を制御するフレームワーク Flamework Frameworkを提案する.

Flamework Framework は, レシピデータから取得した消費電力の数値に基づいて火力を制御するフレームワークである. 本システムにより, 火の強さが弱火や中火といった曖昧な表現ではなく, 数値に置き換わるので, 違うコンロを使う家庭でも正確に同じ火力を再現できる. また, 調理中の細かい火の調整も本システムが行うので, ユーザに火の管理を意識させないことが可能になる. 本論文では, Flamework Framework の試作を通じ, 可能性や料理体験の変化, マシンリーダブルなレシピについて考察する.

2. Flamework Framework

Flamework Framework は火力をプログラマブルにするフレームワークである(図1). 調理するレシピの各手順と対応した火力データに基づいて制御するので, ユーザは Flamework Framework に指示するだけで火を使った調理ができる.



図1 Flamework Framework のUI(左)と本体(右)

2.1 システム実装

Flamework Framework は, ネットワークを通して火力を制御できるコンロと, それと連動して調理手順を提示するアプリケーション, 専用のレシピデータの3つで構成する.

2.1.1 ハードウェア

コンロは Panasonic 社の IH 調理器 KZ-PS1P[4]をベースに試作した. ネットワークを通じてデバイスに命令を送ることで, 火力制御ができる. コマンドは火力のオンオフと強さの指定がある. 本デバイスでは, 火力の強さを6段階で指定する.

2.1.2 アプリケーションと使用方法

ユーザはアプリケーションをタブレット端末で操作する. 火を使った調理を終える時, レシピデータに実行時間が設定されていれば自動で止まるが, 食材の状態が変化したときに止める場合は, ユーザがアプリケーションを操作する必要がある. 例えば, ジャガイモが柔らかくなった状態や, 肉の色が変わった状態などはユーザが判断して止めなければならない.

2.2 レシピデータ

Flamework Framework がデータに基づいて火力を制御で

^{†1} 明治大学総合数理学部先端メディアサイエンス学科

^{†2} (株) シードルインタラクティブデザイン



図 2 Framework Framework の UI: ユーザは点火ボタンを押して、その手順に書かれていることを行う

きるようにするため、デバイスが読み込めるデータが必要になる。本論文では既存のレシピを JSON 形式に整形し、ユーザの作業とデバイスが読み込む火力のデータを定義した (図 3)。

このレシピデータを作成する時、いくつかの制約がある。例えば、従来のレシピの『食材がある状態になるまで焼く』といった見目で判断しなければいけない手順の場合、デバイスが食材の状態を直接判断することはできないので、時間と火力をキーにしてレシピを整形する必要がある。

3. 考察と議論

3.1 試用と考察

レシピデータ内で火力の強さと実行時間を定義し、Framework Framework がそれを直接読み込んで実行することで、人が火加減を意識して操作する必要はなくなった。たとえば、炊飯のような火力調整や加熱時間が肝心のレシピなら、レシピデータがあれば、火を使う工程で人が介在することなく調理を終えることができる。また、レシピを作る時に点火してから消火するまでの時間と、その間の火力の遷移を記録しておくことで、ある食材の状態を変化させるプロセスを数値化出来る。レシピ作成者が判断した状態が数値としてレシピデータに埋め込まれるので、次にそのレシピを使う人は UI の実行ボタンを押すだけで同じ火力の状態を再現できる。

関連研究として、ロボットが調理を支援するシステム[5]がある。この研究では専用の GUI アプリケーションでレシピを定義し、ロボットがそのレシピとタイムラインに基づいて調理を支援する。しかし、レシピを作る際に、ロボットが行う範囲とユーザが行う範囲を上手く分担しないと支援にはならない。また、ロボットが動くための環境を用意しなければならない問題がある。この点において、本システムは従来のコンロにコンピュータを組み込む、またはコンロを置き換えるだけであるため、導入コストが低いと言える。

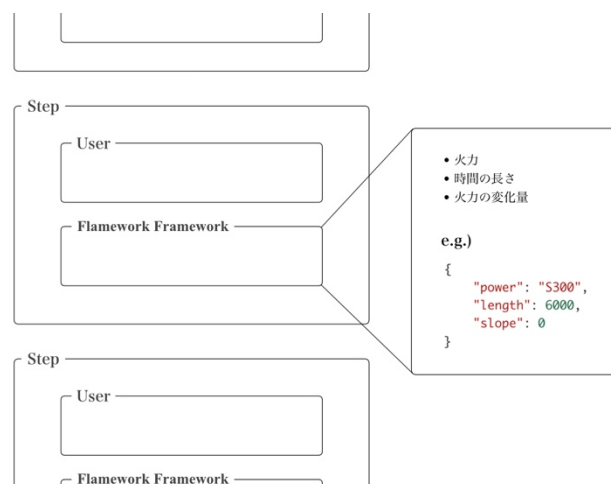


図 3 レシピデータの概要: Framework Framework Object 内に火力の設定や実行時間を設定している

3.2 他のデバイスとの連携

さらに、他のデバイスと連携することで、人の入力が必要としていた部分がデバイスに置き換わる。例えば、食材の見た目の状態で火を止める場合、今までは人が見て判断して止めていた。しかし、食材の見た目を判断するのがデバイスに置き換わることで、正確なタイミングで火を止めることが可能になる。

3.3 目標とする料理の在り方

家庭で料理をする際、様々な工程が含まれる。そのため、まずは置き換えやすい作業を機器に任せ、徐々にロボットのようなデバイスによって置き換わるだろう。こうした流れを踏まえながら、レシピデータのフォーマットを設計する必要がある。このデータフォーマットができることで、作り手は専用の GUI エディタでデータを編集、火力を制御して調理プロセスを記録することが次の展開になる。

4. おわりに

本論文では、火力を例にレシピのデータ構造を考案し、それに基づいて火力を制御するフレームワーク Framework Framework とコンロを提案、試作した。

参考文献

- [1] cinder, <https://www.cindergrill.com/> (参照 2017-12-20).
- [2] 渡邊恵太, 佐藤彩夏, 松田聖大, 稲見昌彦, 五十嵐健夫. smoon: Web の実体化に依る行動支援とその試作. 第 19 回インタラクティブシステムとソフトウェアに関するワークショップ予稿集(WISS2011), pp. 84-89. 日本ソフトウェア科学会, 2011.
- [3] Fab Food Project, <https://keitalab.com/fabfoodproject/> (参照 2017-12-25)
- [4] Panasonic IH 調理器 KZ-PS1P, <http://panasonic.jp/ih/p-db/KZ-PS1P.html> (参照 2017-12-25)
- [5] Y. Sugiura, D. Sakamoto, A. Withana, M. Inami, and T. Igarashi. Cooking with robots: designing a household system working in open environments. CHI'10, pp. 2427-2430, New York, NY, USA, 2010. ACM.