

グラフィカルデータフローによる 調理レシピプログラミング言語の提案

吉川 祐輔^{†1} 宮下 芳明^{†1}

本稿では、調理レシピを記述するためのデータフロープログラミング言語を提案する。提案手法でレシピを記述することで、調理と食材の流れを視覚的に把握しつつ、構造化されたレシピを得ることができるようになる。本稿ではさらに、調理レシピをデータフロープログラミング言語として表すために必要な要件と、もたらされるであろう未来について述べる。

A Proposal of Cooking Recipe Programming Language using Graphical Dataflow

YUUSUKE KIKKAWA^{†1} and HOMEI MIYASHITA^{†1}

We propose a data flow programming language to describe cooking recipes. It brings visual showing of the flow of cooking and foodstuffs and the structured recipes to describe recipes by the technique that we suggest. In this paper, we observe requirements to express cooking recipes as data flow programming language and our vision.

1. はじめに

1.1 背景

通常、料理のレシピは人に読まれるようにテキストで記述する。よって、それを機械で読んで利用する場合、レシピを言語解析を用いて構造化しなければならない。そのような研究

には、浜田らの手法¹⁾ や高野らの手法²⁾ があるが、自然言語が曖昧性を内包しているが故に、常に完全な構造のレシピが得られるとは限らないことが言及されている。

一方、構造化されたレシピを得るために、RecipeML³⁾ のような XML を用いたレシピ記述法がある。しかし、レシピを記述する側からすると、XML を用いるのは難解であり、また記述された XML を人間が理解するのは難しい。

現在、プログや COOKPAD⁴⁾ などのレシピ共有サイトを用いて自分のレシピを公開している人は多いが、そのような人たちにも簡単に利用できる、構造化したレシピの記述法があれば、世界中の様々な料理のレシピの構造を得られ、それを利用できるようになるはずである。しかしながら、曖昧性を排除した構造を持ちつつ、人間にも書きやすい記法は未だ確立されていないと言える。

1.2 レシピプログラミング

安村は、料理のレシピの記述は計算機のプログラミングに似ていると指摘している⁵⁾。また『Structure and Interpretation of Computer Programs⁶⁾』では、プログラムについて次のように述べられている：

“Thus, programs must be written for people to read, and only incidentally for machines to execute.” (プログラムは人々によって読まれるように記述されるべきで、たまたま機械がそれを実行できるに過ぎない (著者訳))

つまり、人間を、レシピ (プログラムのソースコード) に従って料理する (スクリプトを実行する) 計算機であるかのように捉えれば、料理のレシピは人間の調理工程を記述するためのプログラミング言語であると言える。そしてレシピをプログラミング言語として捉えることで、プログラミング言語の特性である階層化された記述や再利用性を用いることができる。

1.3 データフロープログラミング言語

データフロープログラミング言語は、データの流れを線でつなぎデータフローグラフとしてプログラムを作成する言語であり、データや処理の流れを視覚的に把握できるという特徴がある。

一方、浜田らの開発した調理支援システム⁷⁾ や宮脇らの開発した調理支援システム⁸⁾ では、調理中に提示するレシピをフローグラフを用いて表現しており、人間がレシピの構造を把握するうえでフローグラフで表現することは有効であると考えられる。

これらより、データフロープログラミング言語としてレシピを記述することで、料理の材料や調理工程の流れを視覚的に把握しながら構造化されたレシピを“プログラミング”でき

^{†1} 明治大学大学院理工学研究科新領域創造専攻デジタルコンテンツ系

Program in Digital Contents Studies, Program in Frontier Science and Innovation, Graduate School of Science and Technology, Meiji University

るようになると思われる．

1.4 目的

我々は、構造化された調理レシピを記述するための、グラフィカルデータフローによる調理レシピプログラミング言語を提案する．

本提案手法を用いてレシピを記述することによって、XML を書くときのように複雑な記法を覚えることなく、料理の材料や調理工程の流れを視覚的に把握しつつ、機械にとっても人間にとってもその構造を理解しやすい正しく構造化されたレシピをプログラミングできるようになると考えられる．

2. 仕様への要求

この章では、調理レシピをフローグラフプログラミング言語として表すための仕様への要求を述べる．

2.1 オブジェクト

本提案手法では、全てのデータや関数は「オブジェクト」として表現し、オブジェクト同士を線でつなぐことで有向グラフとしてプログラムを作成する．

各々のオブジェクトが持つ、オブジェクト同士を繋ぐ点をインレット・アウトレットと呼ぶ．インレットはオブジェクトの上部にあり、関数の引数や変数への代入を受け取る．アウトレットはオブジェクトの下部にあり、関数の戻り値や変数の値を出力する．オブジェクト同士を繋ぐ時、その線は必ずアウトレットからインレットへと向かう．

オブジェクトはその役割によって様々な種別に分けられる．本手法で用いるオブジェクトの種別、並びにその役割を表 1 に示す．これらのオブジェクトのうち、関数オブジェクトについては 2.3 節で、モジュール・インレット・アウトレットオブジェクトについては 2.5 節で詳しく後述する．

2.2 材料部と手順部

調理レシピは、一般的なレシピと同様に、材料を表す部分と手順を表す部分の 2 つに分けて表す．

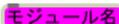
2.3 調理工程の関数化

調理工程は、プログラミング言語の関数として表現する．関数は、一つの工程或いは複数の工程の組み合わせである．

2.3.1 関数の動作

一つの関数は、二つのインレットと一つのアウトレットを持つ．第一インレットは、調理

表 1 オブジェクトの種別
Table 1 kinds of Objects

オブジェクト形状	オブジェクト名	オブジェクトが示すもの
	調理器具オブジェクト	調理器具・食器
	食材オブジェクト	食材・調味料
	数量オブジェクト	食材の量・ 出来上がる料理が何人前であるか
	形容オブジェクト	調理時間・火加減（温度）・焼き加減（色） などの情報
	テキストオブジェクト	オブジェクトを展開したとき 詳しい情報を表すテキスト
	繰り返しオブジェクト	繰り返し範囲・回数の指定
	関数オブジェクト	切る・焼く・混ぜる などの調理工程
	モジュールオブジェクト	いくつかの調理工程を経て 出来上がるもの
	インレットオブジェクト	料理の数量の入力
	アウトレットオブジェクト	出来上がるものの出力

器具並びにそれに付随する食材・形容表現、すなわち「～に」「～で」の「～」に入るものを表す．第二インレットは、それに加える食材・形容表現、すなわち「～を」の「～」に入るものを表す．アウトレットは、その工程によってできたものを表す．

以下の例では、調理器具・食材は「」で括り、調理工程は『』で括り、工程によってでき上がった中間食材は{ }で括って記述する．中間食材の名前は図には表現されないが、便宜上用いる．

例えば図 1 の (1) では、「ボウル」に「塩」「牛乳」「卵」を入れて『混ぜる』ことを表し、その結果できたもの { ボウルに入った卵液 } がアウトレットからの出力である．

何かのアウトレットを第一インレットに持つ関数は、そのアウトレットの調理器具を引き継ぐ．従って (2) では (1) の調理器具「ボウル」を引き継ぎ、「ボウル」で { ボウルに入った卵液 } に「食パン」を『浸す』事を示す．これによってできたものは { ボウルに入った、卵液に浸した食パン } である．第一インレットとアウトレットは縦に並ぶため、同じ調理器具で調理を引き続き行う場合その工程は全て縦一列に並べられる．

何かのアウトレットを第二インレットに持つ関数は、そのアウトレットの調理器具から関数の第一インレットへ食材を移すことを示す．よって (3) では、「バター」を引き「弱火」に熱した「フライパン」に、「ボウル」から { 卵液に浸した食パン } を入れ「きつね色にな

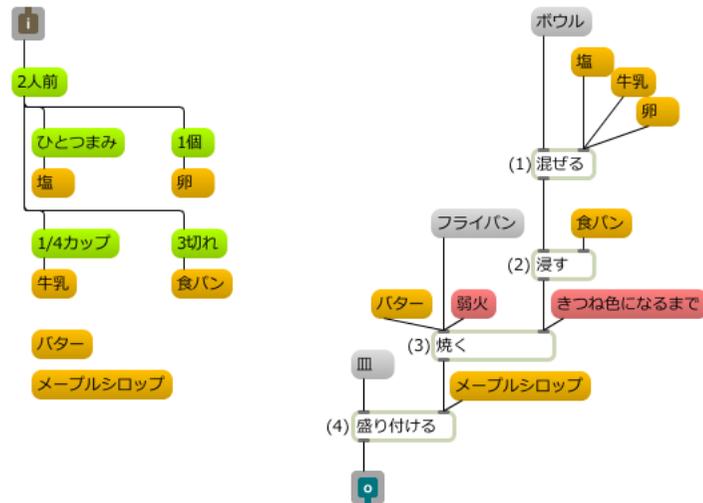


図1 フレンチトーストのレシピ
Fig.1 a recipe of french toast

るまで』『焼く』事を示す。

そして(4)では「皿」に「卵液を浸して焼いた食パン」と「メープルシロップ」を『盛り付ける』ことを示す。

2.3.2 関数の展開

一つの調理工程が複数の工程の組み合わせを表すとき、例えば、人参を「みじん切り」すると言ったとき、これは実際には、人参を「薄切り」し、「千切り」し、さらに「端から細かく切る」という工程を踏む。このような表現は、料理の得意な人にとっては複雑な工程を一言で表す簡潔な表現だが、それを知らない人にとっては難しい表現である。

このような、一つの言葉で複数の動作を表す言葉は、プログラミング言語で言えば複数の関数を組み合わせて一つの関数を定義することに似ている。

これに習い、関数を展開することでより詳しい情報を得られるようにする。複数の工程の組み合わせを表す関数はその実装を図2のように示し、一つの工程を表す関数に関しては図3のようにその工程を表す詳しい情報をテキストオブジェクトで示す。

これにより、関数名を見てその関数の動作が分かる人にとっては簡潔な表現となり、その関数の動作が分からない人は関数の実装を参照することで関数の動作を知ることができる



図2 みじん切りの関数
Fig.2 a Function of Chopping

図3 千切りの関数
Fig.3 a module of thin strips

ようになる。

2.3.3 関数化の問題点

本手法は調理工程を関数として表現しているが、多くの調理工程はその食材によって大きく意味を変える。

図2では人参の場合の「みじん切り」の関数を示したが、これが玉葱のみじん切りではそのやり方が大きく変わってしまうし、また魚を「さばく」とイカを「さばく」のではその工程は全く違うものになってしまう。

本来、調理工程は食材に付随した表現である。これは、オブジェクト指向プログラミングの考え方で言えば、例えば食材クラスの抽象メソッドとして「みじん切り」などがあり、それを継承したサブクラスの食材（人参や玉葱）が「みじん切り」の実装を持つようにすべきである。しかし、データフロー言語ではそのような実装はできないため、インレットに入力する食材でその関数を展開した時の中身を変化させるような機構を持たせなければならない。

2.4 数量・形容オブジェクトの展開

数量や形容表現においても調理独特の表現が多く、例えば「すり切り1杯」や「180度に熱した油」など、特殊な操作を要求されるものもある。これらに関しても関数などと同様、普段は隠して見えないようにしておきながら詳しい情報が欲しい時にそれを展開して提供できるようにする。

2.5 モジュール

いくつかのオブジェクトのを結合させたものを「モジュール」と呼ぶ。一つのモジュールは一つの「食材オブジェクト」のようにふるまう。

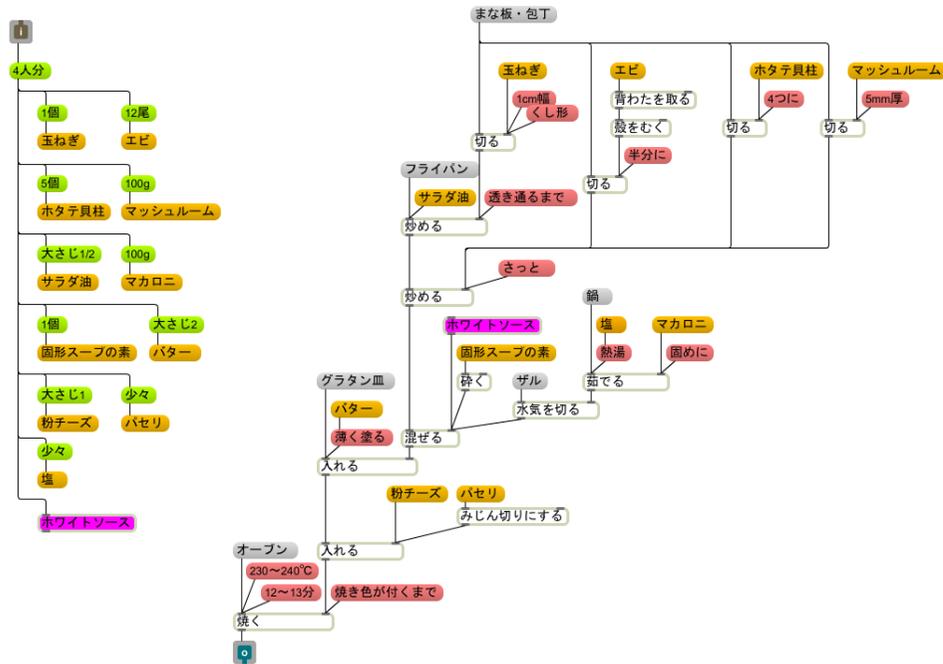


図 4 グラタンのレシピモジュール

Fig. 4 a Recipe Module of Macaroni au Gratin

一つの料理のレシピは一つのモジュールである。例えば「チキンライス」は一つの料理であるが、そのモジュールをあたかも「オムライス」の食材であるかのように用いることができる。

モジュールはインレットオブジェクトとアウトレットオブジェクトをそれぞれ一つずつ持つ。インレットはそのモジュールの数量を受け取り、アウトレットはそのモジュールの生成物を出力する。

たとえば、一度「ホワイトソース」モジュールを作ってしまうと、そのモジュールを「グラタン」「ドリア」「ラザニア」「ホワイトシチュー」などのレシピに再利用できるようになる。図 4 と図 5 は、共通のホワイトソースのモジュール (図 6) を用いたグラタンとドリアのレシピの例である。

このように、料理の特定の結合をモジュールとして持つことで、モジュール同士の組み合

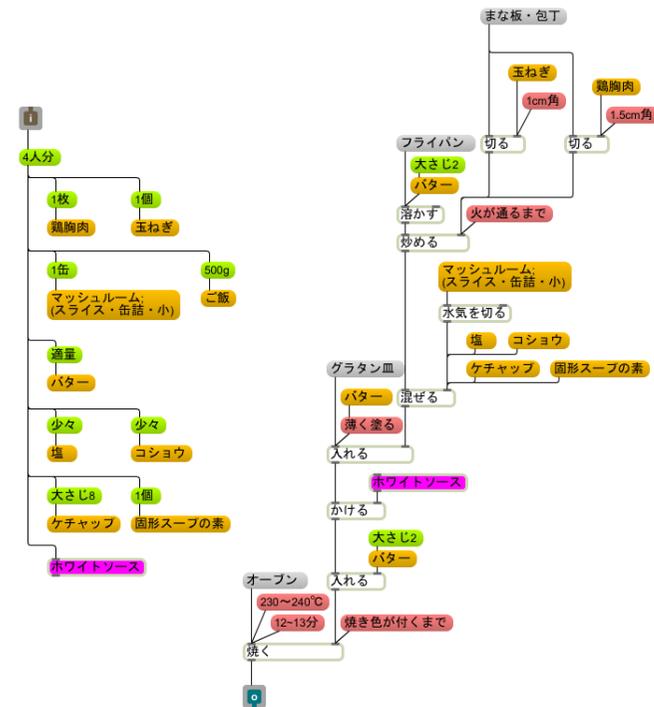


図 5 ドリアのレシピモジュール

Fig. 5 a Recipe Module of Doria

わせでレシピを簡潔に記述できるようになる。

2.6 繰り返し

レシピの調理工程が何回かに分けて繰り返される場合、図 7 のように繰り返しオブジェクトによって表現する。その調理工程が「人数分繰り返す」ものだった場合、繰り返しオブジェクトのインレットには、その料理のレシピが「何人分」のものであるかを入力する。

3. 未来ビジョン

本手法で作られたレシピは構造化されているので、これを利用した様々な利用方法が考えられる。この章では、本手法を用いてレシピを記述することで生まれる様々な利用法の可能

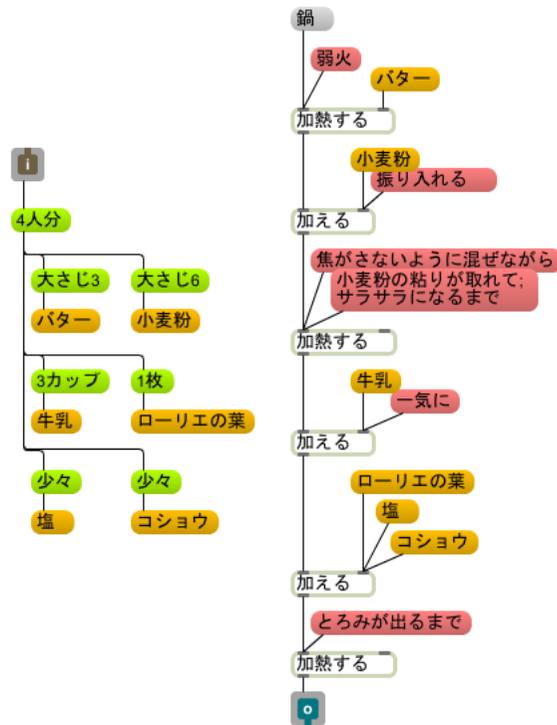


図6 ホワイトソースのレシピモジュール
Fig.6 a Recipe Module of White Sauce

性について述べる。

3.1 レシピのの対象人数変換

通常のレシピでは、それが何人分の料理を作るものであるかは固定されている。本稿で提案する記法を用いれば、その「何人分の料理を作るのか」を可変にできる。その料理の対象人数を変換するときには、料理のモジュールのインレットオブジェクトから何人前の料理を作るかを受け取り、その値に応じてその料理に必要な食材の量や繰り返し回数を変化させることができるようになる。

3.2 レシピの表現手法

この節では、本手法で作成したレシピの様々な表現手法について述べる。

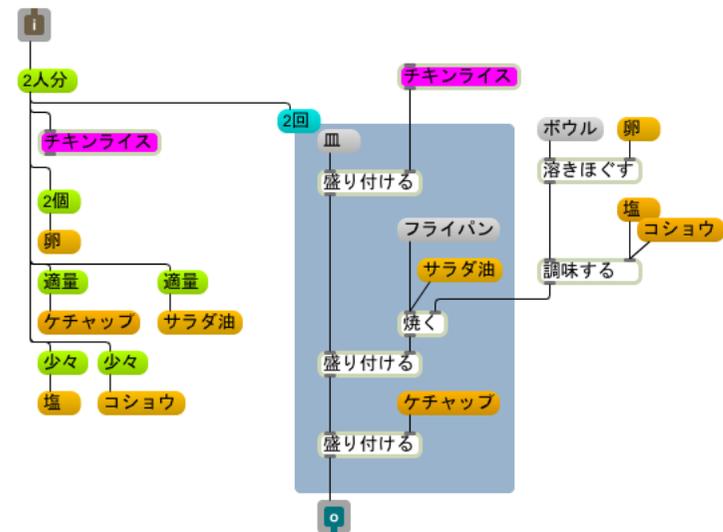


図7 オムライスのモジュール
Fig.7 a Module of Omelet Containing Fried Rice

3.2.1 タイムライン化

調理手順を、その調理時間を加味して調理手順に並べることができれば、フローグラフレシビをタイムラインとして表すことができるようになる。

高野らは、調理スケジューリングを行うことで、調理時間を考慮したタイムチャートを生成する手法を開発した⁹⁾。浜田らは料理手順のフローグラフ構造を作業空間の制約などを満たすように最適化する手法をとった¹⁰⁾。

これらの先行研究のように、調理するときタイムラインで表記することは調理を行う上で大きな助けになると考えられる。

タイムライン表示をするためには、その工程にかかる時間を知らなければならない。しかし、例えば野菜の下ごしらえなど、初級者と上級者では大きくかかる時間が変わる工程もあれば、パスタの茹で時間など誰でも同じ時間がかかる工程もある。したがって、レシピをタイムライン化するにはそのレシピにかかる時間をモジュールごとに可変にするか不変にするかを決定しなければならない。また、3.1節でも述べたとおり、何人分の料理を作っているかによっても繰り返し回数が変わる場合にも、それに従って調理時間も変動する。さら

に、その調理工程で並列処理ができるかどうか大きな問題である。例えば、「パスタを茹でながらも食材を切ったりすることは可能」など積極的に調理時間を短縮するために並列処理を行うこともあれば、「パスタの茹で上がりと同時にパスタソースができて上がるように調理する」など同時に調理することが前提の工程もある。

レシピのタイムラインの作成者に対して、これらの問題を考慮し自由にタイムラインを作成できるようにすべきである。また、調理時間は個人差が大きいので、作成されたタイムラインをそのまま利用するだけでなく、ユーザが個人ごとに自由にカスタマイズできるようにすべきである。

3.2.2 テキスト化

テキストのレシピの方が分かりやすい人に対しては、テキストでレシピを提示することができるようにする。テキストのレシピから構造化されたレシピを得る場合、完全な構造のレシピを作ることはできないが、構造化されたレシピからテキスト化されたレシピを得ることは可能である。テキストの並べ方は、前述のタイムライン化手法で並べ替えられたレシピを用いればよい。

3.2.3 ビジュアル化

関数や数量・形容オブジェクトなど展開できるオブジェクトは、それを展開したときにビジュアルな情報（画像・映像）を提示できるようにすることで、調理方法が分かりづらい工程について、理解を助ける情報を与えられるようになると考えられる。

浜田らのシステム⁷⁾では、レシピのテキストやフローグラフのノードを選択することで映像を提示し調理を支援する。宮脇らのシステム⁸⁾では、現在の調理状況に応じて映像やテキスト、フローグラフでレシピを提示し調理を支援する。

これらでは、既にある料理番組などの調理映像をレシピに対応付けているが、もっと汎用的な事柄、例えば材料の計り方や火加減の調整の仕方などについての情報もオブジェクトに埋め込み提示できるようにすることで、オブジェクトを展開したときにより詳しい情報を提示できるようになる。

3.3 設計支援

レシピの記述方法は人によって様々だが、本手法を用いることでそれを一つのフォーマットで表すことができ、複数のレシピを改変したり組み合わせたりすることが容易になる。これにより、例えばカレーとうどんを組み合わせるとカレーうどんのレシピを作ったり、ピザソースと中華まんを組み合わせるとピザまんのレシピを作ったりするように調理レシピのマッシュアップをすることができる。

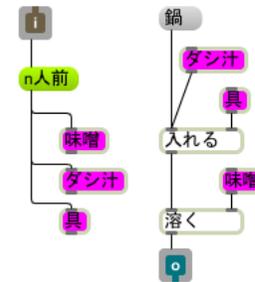


図 8 抽象化された味噌汁モジュール
Fig. 8 an Abstraction Recipe Module of Miso Soup

表 2 派生モジュール
Table 2 Abstract of Miso Soup

サブモジュール	派生モジュール		
味噌	赤味噌	白味噌	八丁味噌
ダシ	鰹節	昆布	煮干
具	大根	豆腐	つみれ
	鶏肉	わかめ	ごぼう
	こんにゃく	油揚げ	刻みしょうが

他にも、レシピは通常、本や Web 上のテキストのものをそのまま読むことになるが、例えばアレルギーがあって使用できない食材や手に入りにくい食材がレシピに用いられていた場合など、それを代替する食材を用いるレシピへとユーザが簡単にカスタマイズできるようにすべきである。

3.3.1 抽象モジュール

一つの料理の抽象モジュールを作ることで、その料理をベースとした料理に様々なバリエーションを持たせることができる。例えば図 8 のように「味噌汁」の抽象モジュールを一つ作り、そして表 2 のように、そのサブモジュールとして様々な「味噌」「ダシ」「具材」の派生モジュールを受け取れるようにすることで、様々な味噌汁を表現することができるようになる。

3.4 データベース化

本手法を用いて作られたレシピを収集しデータベースとすれば、レシピの共有や料理のカテゴリや、食材、調理法などから検索できるようになる。これを例えば塩澤らのレシピ検索手法¹¹⁾などと組み合わせることができれば、レシピ同士を組み合わせるとバランスの良い献立を自動生成できるようになる。

また、レシピの構造に着目することで、その構造の類似度を計算したり、或いはそこからその料理の難易度を自動的に判定する式が定式化できれば、難易度による検索やソートができるようになる。

3.5 料理ロボット・料理器具

例えば、本手法のプログラムが、杉浦らの Cooky¹²⁾ のような調理ロボットや、調理家電

のプログラムに利用できるような形で用いることができれば、自動調理のためのプログラミングとして、本手法を用いることができるようになる。

4. おわりに

本稿では、調理レシピをフローグラフプログラミング言語として記述することを提案し、それを記述するための仕様、並びに本手法で記述することで利用できるだろう様々な可能性について示した。

今後は、本稿で説明した手法によりレシピを記述するためのエディタを開発し、本手法を利用できる環境の構築を目指す。

参 考 文 献

- 1) 浜田玲子, 井手一郎, 坂井修一, 田中英彦: 料理テキスト教材における調理手順の構造化, 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, Vol.85, No.1, pp.79-89 (2002).
- 2) 高野哲郎, 上島紳一: Cooking Scenario: レシピの Scenario 化とその応用, 電子情報通信学会技術研究報告. DE, データ工学, Vol.103, No.190, pp.19-24 (2003).
- 3) FormatData: RecipeML(The Recipe Markup Language).
<http://www.formatdata.com/recipe/ml/>.
- 4) クックパッド株式会社: COOKPAD.
<http://cookpad.com/>.
- 5) 安村通晃: Programming2.0: ユーザ指向のプログラミング, 情報処理学会夏のプログラミング・シンポジウム報告集, No.2006, pp.115-122 (2007).
- 6) Abelson, H., Sussman, G.J. and Sussman, J.: *Structure and Interpretation of Computer Programs - 2nd ed.*, The MIT Press (1996).
- 7) 浜田玲子, 宮澤 寛, 鈴木幸敏, 岡部 淳, 佐藤真一, 坂井修一, 椎尾一郎: コンピュータ強化キッチンによるインタラクティブ調理支援, 第 13 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 (WISS 2005) 論文集, No.38, pp.49-52 (2005).
- 8) 宮脇健三郎, 佐野睦夫: ユーザ適応型タスクモデルによる調理ナビゲーションシステム (料理メディア研究会特別セッション), 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, Vol.107, No.454, pp.63-68 (2008).
- 9) 高野哲郎, 上島紳一: 段取りの導出を行う調理支援システムの提案 (<特集> 矢島脩三教授定年退職記念), 情報研究: 関西大学総合情報学部紀要, Vol.22, pp.117-142 (2005).
- 10) 浜田玲子, 井手一郎, 佐藤真一, 坂井修一: マルチメディア調理支援ソフトウェア「HappyCooking」, 第 2 回デジタルコンテンツシンポジウム, Vol.CD-ROM (2006).
- 11) 塩澤秀和, 三田村祐介: 食材の優先度を考慮した料理レシピの検索 (セッション 3:イ

- ンタラクシオンデザイン:理論と実践 (3)), 情報処理学会研究報告. HCI, ヒューマンコンピュータインタラクシオン研究会報告, Vol.2007, No.41, pp.51-57 (2007).
- 12) 杉浦裕太, 坂本大介, Withana, A.I., 稲見昌彦, 五十嵐健夫: Cooky: 調理順序指示インタフェースと料理ロボットの開発, 第 17 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 (WISS 2009) 論文集, No.63, pp.75-80 (2009).