料理プログラミングの為の枠組みについて

吉川祐輔[†]　宮下芳明[†]

本論文では，レシピを読んでその通りに作ることが出来る程度の初心者を対象とし，データフロープログラミングの枠組みを利用して料理レシピの改変やマッシュアップが行える環境を構築した．提案システムでは，ある料理を食材のように用いることができるモジュール化機構や，画像処理によって調理工程を表示する簡易ビジュアライザを設けた．

Frameworks for Cooking Recipe Programming

Yuusuke Kikkawa[†] and Homei Miyashita[†]

In this thesis we propose a recipe programming environment as emergent media to arrange or mash-up recipes, and finally to develop brand-new cooking, for people that follows a recipe to the letter. The system provides module function so that the user can handle a cooking as a 'cooking ingredient'.

1. はじめに

新しい料理を生み出すことは難しい．新しい料理を生み出そうという意識を持つこともまた，難しいものである．プロの料理人は常に新しい料理を模索し創作しているが，レシピを読んでその通りに作ることが出来る程度の初心者は「決められた手順があって，その通りに作らなければいけない」というイメージに囚われてしまう．本システムは，既存のレシピの一部を改変したり組み合わせたりすることで新しい料理を作り出す，創発メディアとしてレシピを記述する環境を提供する．

ユーザがコンテンツを生成していくメディアはCGM(Consumer Generated Media) と呼ばれる．CGM では，しばしば「N次創作」と呼ばれる現象が発生する．これは，オリジナルコンテンツを利用して新しいコンテンツを作成するという「二次創作」に対し，それがさらに利用され継承され，三次・四次・五次，と連なっていくさまを表す言葉である．ウェブサービス，ニコニコ動画[1)] では，投稿された動画を利用して新しい動画が作成されるケースが非常に多い．ニコニコ動画の場合，N次創作は今や特有の文化として定着している．他にもN次創作を奨励するCGMは多数存在し，たとえばゲームポータルサイト，ニコゲー[2)] は，ユーザがゲームを作りそれを公開し，コミュニケーションしながら共同制作を行える．使用音源や画像などの素材は，誰かが公開したものを利用したり自分で作って公開することができ，(エディット可能なものならば) 誰かが作ったゲームを改変して新しいゲームを作ることが可能である．オリジナル曲やイラストを投稿できるCGMコンテンツ投稿サイト，PIAPRO[3)] では，「関連する動画」「関連する作品」を登録する機能や，「創作ツリー」という派生作品の親作品を登録できるシステムを持っており，N次創作における作品同士のつながりを意識させる作りになっている．プログラミング環境 Scratch[4)] は，自分の作ったプログラムを共有したり，他人が作ったプログラムをダウンロードしたりできるプロジェクト共有サイトで活発なコミュニケーションが行われている．wonderfl[5)] は，ActionScript による Flash クリエイタの為の SNS であり，新規に ActionScript を作る機能や別のユーザが作った ActionScript をアレンジできる機能を持っている．

ここで注目すべき点は，どのCGMメディアも非常に活発な作品投稿がなされており，中にはプロ並みに優れたコンテンツも生まれている，ということである．本論文では，冒頭で述べたように「レシピを読んでその通りに作ることが出来る程度の初心者」を対象とし，既存のレシピの改変からスタートして，最終的にはオ

† 明治大学 理工学研究科 新領域創造専攻 ディ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

リジナルレシピ創作と配信につながる意識変化を促したいと考えた．

料理においてはレシピ CGM サービス，COOKPAD[6] の「つくレポ」のように，レシピに対してそれを作ったことを報告する仕組み程度しかなく，上記のようなN次創作的な運動には至っていない．しかしながら，多くの新しい料理が生まれることは人間の食文化を豊かにする．さらにそこに一般の人々のクリエイティビティが関わることは非常に意味があると筆者は考えている．また，キャラクターのデザインを施したお弁当「キャラ弁」をとってみても，アマチュアの料理とは思えない完成度を見せているものも多く，主婦が持つ料理に対する潜在的なクリエイティビティはもっと生かされるべきだと考えている．もちろん最初からプロのように創作料理を行えなくても，まず最初はレシピをアレンジしたりマッシュアップしたりすることから入ることができないだろうか？

さて，このような目的で料理のレシピを記述する場合，テキストベースのレシピでは問題がいくつかある．まず，すでにある記述を再利用したり改変したりすることが難しいことがあげられる．普通の文章で書かれた記述は前後との整合性を強く考慮しなければならないため，単純にコピー&ペーストするだけでは記述した内容を入れ替えることは難しい．また，自然言語に内包される構造の曖昧性ゆえに，テキストベースでレシピを記述するとそのレシピの記述も曖昧な構造が発生してしまう．テキストベースのレシピを構造化する試みとしては，浜田らの手法[7] や高野らの手法[8] があるが，自然言語は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これらで言及されている．

プログラムがしばしば料理のレシピに例えられるように，プログラミング言語の記述はレシピの記述と似ており，その類似性は安村が指摘している[9]．そこで本システムでは，既存のレシピを利用しやすい形態として，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．こうすることで自然言語で構造の曖昧さを排除し，再利用しやすい形で記述することが可能になる．本論文では，プログラミングの枠組みとして，データフロープログラミングを採用した．これは，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．浜田らの開発した調理支援システム[10] や宮脇らの開発した調理支援システム[11] 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いるという先行例がある．つまりデータフロープログラミングの記述を料理に採用することに新規性はないが，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の有効性は高いと考えられる．

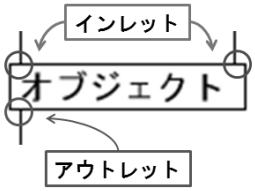

図 1　オブジェクト
Fig. 1　an object

筆者らは，このためのプログラミング言語を提案した[12]．本論では，その提案を元に，さらにビジュアライザなどの支援機構などを付加したシステムについて述べる．

本システムでは，レシピのアレンジやマッシュアップが簡単になるような仕組みを多く設け，またモジュール化機構によって，オムライスにおけるチキンライスのように，ある料理を「食材として」新しい料理を創造することも可能である．また本システムでは，レシピ上に簡易ビジュアライザを設けている．データフロープログラミング言語では，そのプログラムの「状態」を常に知ることができ，その一部を変更すればそれはただちに反映され，自分が今行った変更が及ぼす影響がすぐに分かる．本システムのビジュアライザは，調理工程を経るごとに食材の画像に処理を加えていくことで，その調理工程が行われた結果を表示する．これによりプログラムのデバッグがしやすくなり，また工程の変化によって起きる食材の変化を見せることでレシピの改変の結果をすぐに知ることができ，レシピ創作のモチベーションの向上が期待できる．

2. システム外観

2.1 オブジェクト

本手法では，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．このとき,「オブジェクト」はグラフの「ノード」であり,「線」はグラフの「エッジ」である．

各々のオブジェクトが持つ，オブジェクト同士を繋ぐ接続点を「インレット」・「アウトレット」と呼ぶ（**図 1**）．インレットはオブジェクトの上部にある接続点で，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る接続点で，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

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	オブジェクトが持つパラメータ(型)
全体量	全体量オブジェクト	全体量(数値), 単位(文字列)
数量	数量オブジェクト	数量(数値), 単位(文字列), 数量計算関数(関数)
調理器具名	調理器具オブジェクト	調理器具名(文字列), 調理器具画像(画像)
食材	食材オブジェクト	食材名(文字列), 食材画像(画像), 色(色)
調理工程	調理工程オブジェクト	調理工程名(文字列), 調理器具画像(画像), 食材画像(画像) 色(色), 画像処理関数(関数)
形容表現	形容オブジェクト	形容表現(文字列), 色(色), 倍率(数値)
モジュール	モジュールオブジェクト	モジュール名(文字列), 食材画像(画像), 色(色)
テキスト	テキストオブジェクト	テキスト(文字列)
i	インレットオブジェクト	(なし)
o	アウトレットオブジェクト	(なし)

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，並びにそれらが持つパラメータは**表 1** の通りである．

本稿では以降,「○○オブジェクト」のことを簡略化して【○○】と記述する．

2.2　材料部と手順部

一般的なレシピでは材料を表す部分と調理手順を表す部分を分けて書く．本システムでレシピを書く場合においても，それらをそれぞれ「材料部」「手順部」として分けて記述する．**図 2** のように，左を材料部，右を手順部として表示する．材料部と手順部の【食材】はリンクしており，一方で【食材】を生成（消去）すると，もう一方の【食材】もそれに応じて生成（消去）される．

3.　材　料　部

材料部では,【食材】がプログラミング言語で言う「変数定義」のような役割を持つ.【食材】が数量を持つときは【数量】をインレットに接続する.【数量】は【全体量】と接続し,【全体量】は【インレット】と接続する．

3.1　自動数量変換

【数量】のインレットへ【全体量】が接続されているならば，レシピを開いたときまたは【全体量】の値が変化したときに,【数量】の値はその数量計算関数で計算される．

【数量】の数量計算関数はユーザが自由に設定できる．多くの食材の量は全体量に対して線形に変化するが，全体量が小さい時，煮物の煮汁などはその量が線形に変化しない．ユーザが関数を設定できることで，このような場合にも柔軟に対応できる．

4.　調　理　部

調理部では,【調理工程】がプログラミング言語で言う「関数」のような役割を持つ．一つの料理のレシピは main 関数であり，main 関数上で様々な関数を呼び出すことで一つのプログラムを形成するように，レシピ上では様々な調理工程を踏むことで一つの料理を作り出す．

この章では図 2 の調理部を例に，調理工程を一つひとつ追うことで本システムの調理部の記述法を解説する．

4.1　調理工程オブジェクト

一つの【調理工程】は二つのインレットと一つのアウトレットを持つ．

第１インレットへの接続は，調理器具並びにそれに付随する食材・形容表現，すなわち「フライパンに」や「ボウルで」などの「～に」「～を」に当たるものを表す．第２インレットへの接続は，それに加える食材・形容表現，すなわち「塩を」や「食パンを」などの「～を」に当たるものを表す．アウトレットはその工程によってできたものを表す．

以降この節では,「」で括られたものは調理器具・食材を,『』で括られたものは調理工程を,｛｝で括られたものは工程によってでき上がった中間食材を示す．中間食材の名前は図には表現されないが，便宜上用いる．() で括られたものは**図 3** で示された調理工程の手順である．

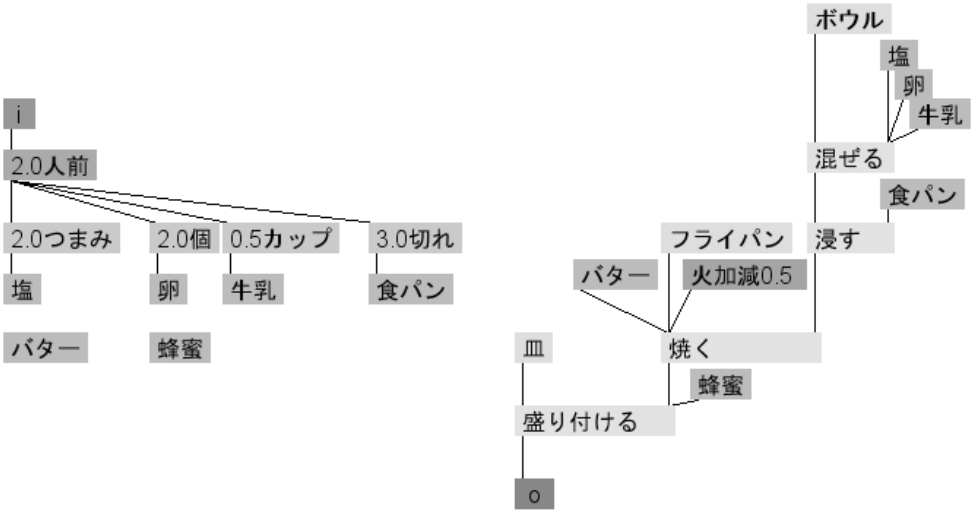

図 2　フレンチトーストのレシピ
Fig. 2　a recipe of french toast

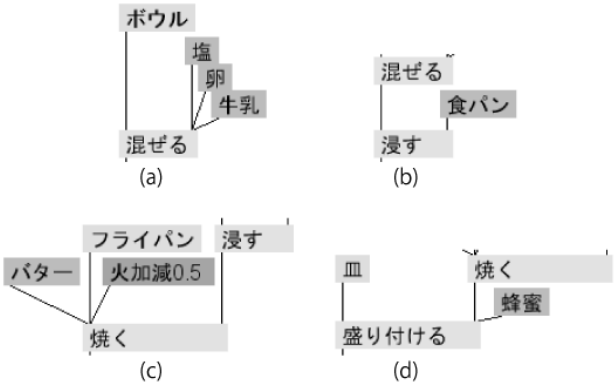

図 3　図 2 の部分レシピ
Fig. 3　a parted recipe of Fig.2

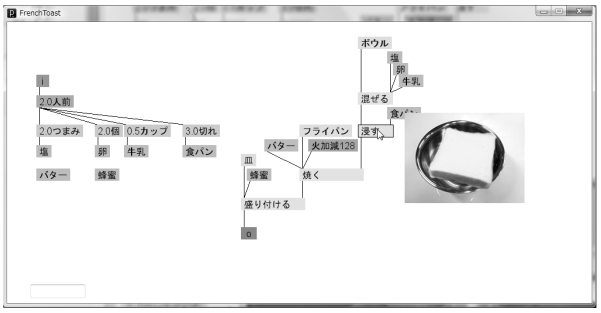

図 4　ビジュアライザ
Fig. 4　visualizer system

図 3 の（a）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出力される．

ある【調理工程 A】のアウトレットが別の【調理工程 B】の第 1 インレットに接続される場合,【調理工程 B】は【調理工程 A】の調理器具を引き継ぐ．たとえば図 3 の（b）では,『浸す』は『混ぜる』の調理器具「ボウル」を引き継ぎ，その「ボウル」で｛ボウルに入った卵液｝に「食パン」を『浸す』事を示す．

ある【調理工程 A】のアウトレットが別の【調理工程 B】の第 2 インレットに接続される場合,【調理工程 B】は【調理工程 A】の調理器具を引き継がず,【調理工程 B】の第 1 インレットへ接続されている【調理器具】に食材を移すことを示す．たとえば図 3 の（c）では,「バター」を引き「火加減」の強さ 0.5 で熱した「フライパン」に「ボウル」から｛卵液に浸したパン｝を入れ『焼く』事を示す．

そして図 3 の（d）では,「皿」に｛卵液を浸して焼いたパン｝を移し，それに「蜂蜜」を加えて『盛り付ける』ことを示す．

4.2　簡易ビジュアライザ

本システムはレシピの簡易ビジュアライザ機能を持つ．これは，レシピ上のいくつかのオブジェクトの上にマウスカーソルを乗せると，その画像を画面に描画する機能である（**図 4**）．

ビジュアライズ機能を持つオブジェクトは【調理器具】【食材】【調理工程】の 3 種である．

【調理器具】や【食材】では，その調理器具や食材の画像を表示する．このとき，それの名前と同じ名前の画像ファイルを読みにいく．そのため，ユーザは画像ファイルを用意しさえすれば，使用したい画像を簡易ビジュアライザに簡単に追加することができる．

【調理工程】では，その【調理工程】のインレットに接続されているオブジェクトが持ついくつかのパラメータを取得し，それを用いて簡単な画像処理を行なうことで中間食材の画像を生成しそれを画面に描画

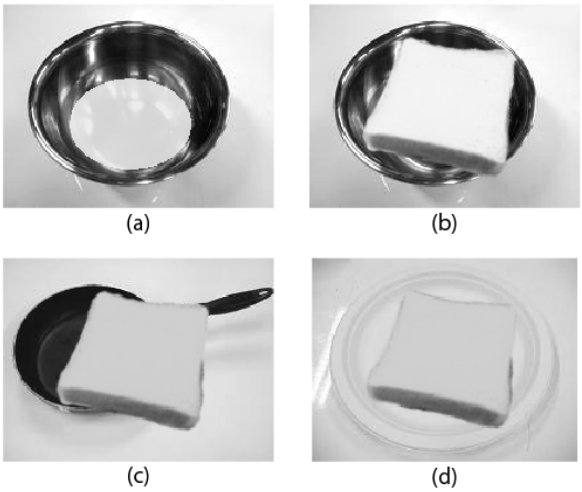

図 5 簡易ビジュアライザの効果
Fig. 5 results of simple visualizer

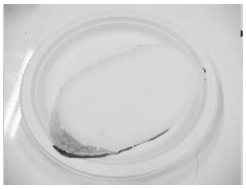

図 6 フランスパンの画像を使った描画結果
Fig. 6 a result with french bread image

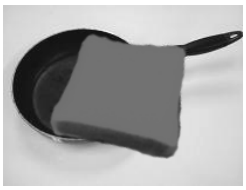

図 7 焦げたパン
Fig. 7 burnt bread

する．

この簡易ビジュアライザは，レシピ上の何らかのオブジェクトの情報が更新されるたびにすべての処理が更新される．そのため，レシピ上の一部を変更したとき，それが及ぼす影響は即座に全体に反映される．

4.3 簡易ビジュアライザの画像処理

この節では【調理工程】の簡易ビジュアライザで行っている画像処理について，図 2 で使われている【調理工程】を例にあげて述べる．この節では，オブジェクトが持つパラメータを“”で括って示す．

【調理工程】は 4.1 節で前述の通り，その工程は何の調理器具を使って調理するのかを第 1 インレットから取得する．すなわち【調理工程】は，第 1 インレットへ【調理器具】が接続されているならば【調理器具】から，【調理工程】が接続されているならば【調理工程】から“調理器具画像”を取得する．

「混ぜる」オブジェクト（図 3 の（a））では，第 2 インレットに接続されている【食材】に設定された“色”パラメータを取得する．そしてそれらの値を平均することで混ぜられた“色”を得て，その色をボウル用のデフォルト画像に乗算合成し画像を生成する．生成された画像を「ボウル」オブジェクトから得られた“調理器具画像”にオーバレイして表示する（**図 5**(a)）．

「浸す」オブジェクト（図 3 の（b））では，直前の「混ぜる」オブジェクトから“色”と“調理器具画像”を取得する．第 2 インレットへ接続されている「食パン」オブジェクトから受け取った“食材画像”と，「混ぜる」オブジェクトから受け取った“色”を乗算合成し画像を生成する．生成された画像を「混ぜる」オブジェクトから得られた“調理器具画像”にオーバレイして表示する（図 5(b)）．ここで受け取る【食材】を変えると，その食材の画像を使ってビジュアライズできる．**図 6** は，「食パン」のオブジェクトの代わりに「フランスパン」を使った場合のビジュアライズ結果である．

「焼く」オブジェクト（図 3 の（c））では，第 1 インレットへの入力である「フライパン」オブジェクトから“調理器具”の画像を，第 2 インレットへの入力である「浸す」オブジェクトから食材の画像を得て，第 2 インレットから得られた画像に“焼き色”を乗算し画像を生成する．“焼き色”は「火加減」オブジェクトから取得する．そして生成した食材画像を“調理器具画像”へオーバレイして表示する（図 5(c)）．ここで「火加減」オブジェクトが持つ“倍率”を変化させると，“焼き色”を変化させることができる．**図 7** は，「火加減」の“倍率”を 0.5 から 1.0 に変化させた場合のビジュアライズ結果である．

「盛り付ける」オブジェクト（図 3 の（d））では，第 1 インレットへの入力である「皿」オブジェクトから取得した“調理器具画像”に，第 2 インレットへの入力「焼く」から取得した“食材画像”をオーバレイして表示する（図 5(d)）．

ユーザが関数を指定すれば，本稿であげた【調理工程】以外のものでもビジュアライズすることが可能となる．

5. モジュール

例えば「チキンライス」は一つの料理であるが，そのレシピ（**図 8**）を一つの食材として扱うことで，「オムライス」のレシピを簡単に記述することができる（**図 9**）．このように，本システムでは，一つのレシピ

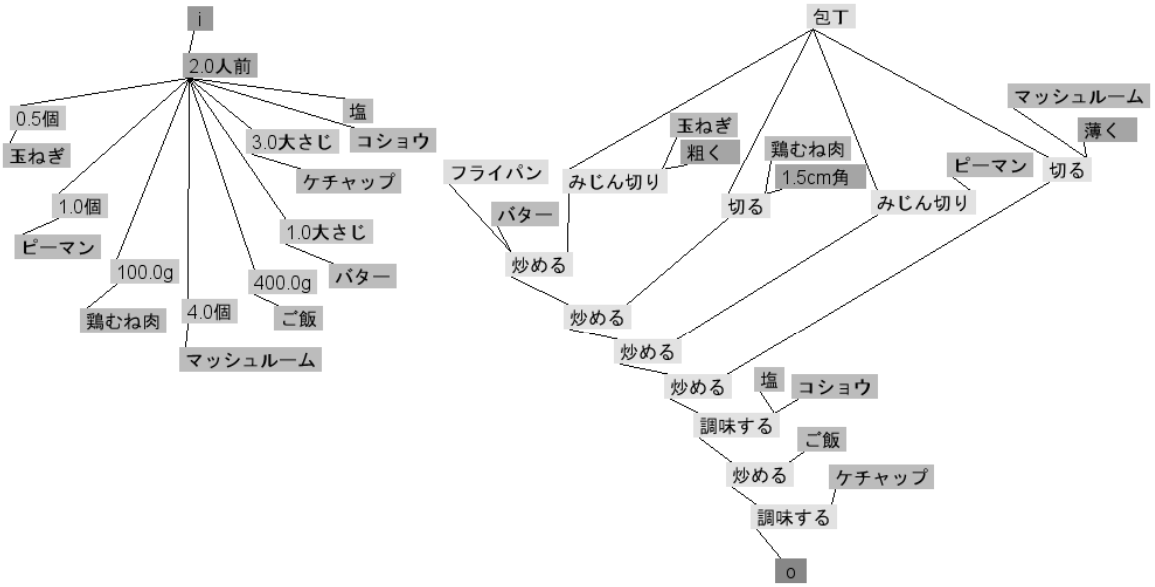

図 8　チキンライスのレシピ
Fig. 8　a recipe of chicken with rice

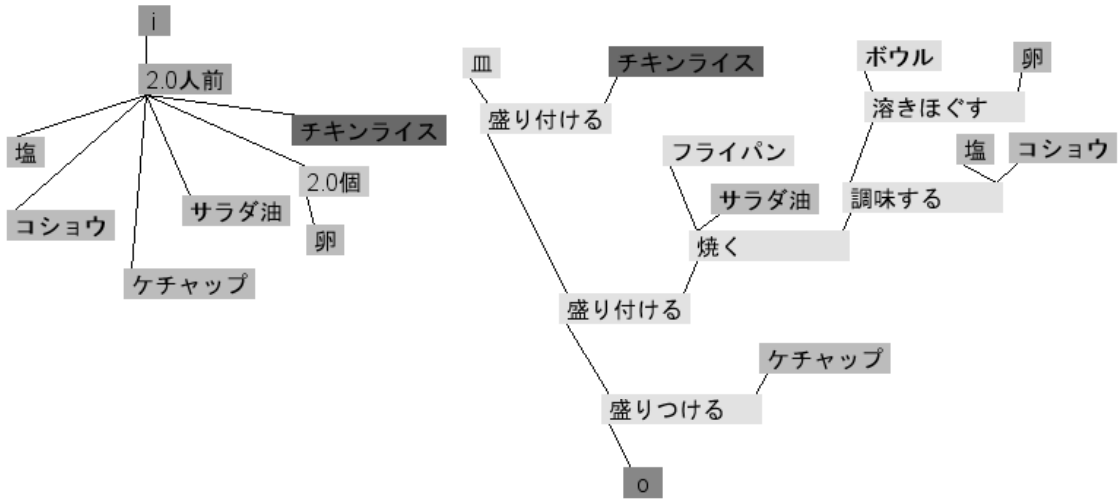

図 9　オムライスのレシピ
Fig. 9　a recipe of omelet containing fried rice

を【モジュール】として記述できる．

モジュール化については，一つの料理のレシピに限らずレシピの一部分についてもモジュール化して利用することもできる．例えば「ホワイトソース」のレシピを一度作ってしまえば，それを【モジュール】として「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．

システム上で【モジュール】を選択するとその【モジュール】のレシピを開くことができ，その【モジュール】の内部を編集することができる．また，ある【モジュール】自体を別の【モジュール】に置き換えることで，レシピの一部をそっくり別のものに取り替えることも可能である．

前述の通り，一つのレシピは【インレット】と【アウトレット】をそれぞれ一つずつ持つ．あるレシピを【モジュール】として扱うとき，そのレシピの【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．すなわち，レシピを一つの関数として捉えると，そのレシピの引数宣言が【インレット】であり戻り値の記述が【アウトレット】である．

一つの【モジュール】は一つの【食材】のようにふ

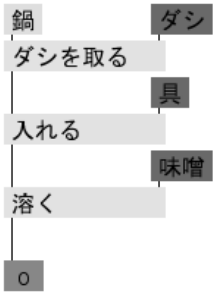

図 10 抽象的味噌汁の手順部
Fig. 10 a procedure part of abstract miso soup

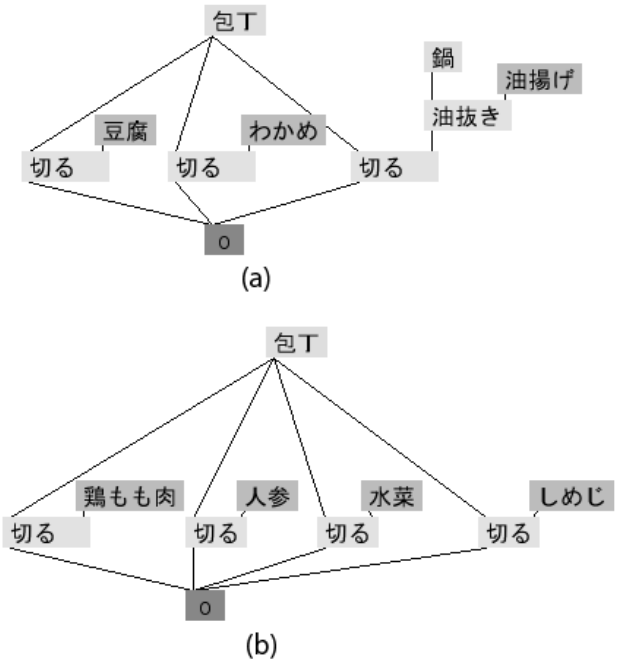

図 11 味噌汁の具モジュール
Fig. 11 modules of miso soup ingredients

るまうが,【モジュール】が【食材】と違うのは，材料部でのインレットの受け取り方である.【食材】は【数量】をインレットで受け取るが,【モジュール】は【全体量】を受け取る.

5.1 レシピの抽象化

レシピを細かくモジュール化することによって，レシピを抽象化することができる．例えば，味噌汁はおおよそ「ダシを取り」「具を入れ」「味噌を溶く」ことで完成する（**図 10**）．この「ダシ」や「具」,「味噌」の部分を様々なモジュールに置き換えることで，様々な味噌汁を定義することができる．たとえば，味噌汁の具モジュールとして**図 11** の (a) が用いられていれば，それは「豆腐とわかめと油揚げの味噌汁」となり，(b) が用いられていればそれは「鶏肉と水菜の味噌汁」となる．このように，レシピを抽象化することによって，一つのレシピから様々なバリエーションの料理を作り出すことができるようになる．これは，抽象クラスとして「味噌汁」を，その抽象メソッドとして「ダシ」や「具」,「味噌」を定義し，そのサブクラスとして様々な味噌汁を定義することに似ている.

そして，例えば図 9 の「チキンライス」モジュールを「焼きそば」にすれば「オム焼きそば」のレシピになり，或いは食材である「ケチャップ」を「ハヤシライスソース」モジュールに変えれば「オムハヤシ」のレシピになる．こういった既存の創作料理も，レシピの一部分を別のレシピ（或いはレシピの一部）に置き換えたものが多くあり，レシピを抽象化することは新しい料理のレシピを生み出す手助けになるだろうと考えられる.

6. 関 連 研 究

木村らは，作曲行為を支援するために，楽曲の構造をツリーグラフとしそれを学習アルゴリズムによって探索することでよりよい構造の楽曲を得る，というシステムを作った[13]．佐藤らは，シークバーの概念を拡張したシークロープを用い，楽曲の一部を切りだしてそれらを組み合わせることで新たな楽曲をリミックスしたりマッシュアップしたりすることができるシステムを示した[14]．これらのように，楽曲の場合，その一部を入れ替えたり組み合せたりすることで新たな楽曲を作り出すシステムは幾つか存在するが，本稿ではレシピの一部を入れ替えたり組み合せたりすることで新たなレシピを作り出すことを可能とした.

槙野らは，菓子レシピを構造化し，その要素を入れ替えることで新たな菓子を作るためのシステムを制作した[15]．このシステムでは，既存の菓子レシピの交換可能な箇所を部分的に変更することで新たな菓子を作ることに主眼をおいており，全く新しい料理を作り出すことを目指す本システムとは趣を異にしている.

7. お わ り に

本稿では，レシピを読んでその通りに作ることが出来る程度の初心者を対象とし，データフロープログラミングの枠組みを利用して料理レシピの改変やマッシュアップが行える環境を構築した．提案システムでは，ある料理を食材のように用いることができるモジュール化機構や，画像処理によって調理工程を表示する簡易ビジュアライザを設けた.

今後は，こうした編集と閲覧をオンラインで行えるように，オンライン上での掲示板に関連画像ファイルをアップロードし，他社がそれをもとに新しい料理レ

シピを試行錯誤できるようにシステムを拡張する予定である．これにより，オンラインコミュニティ上でさらなるN次創作が起こることが期待される．

参 考 文 献

1) 株式会社ニワンゴ：ニコニコ動画. http://www.nicovideo.jp/
2) 株式会社ニワンゴ：ニコゲー. http://www.nicoga.jp/
3) クリプトン・フューチャー・メディア株式会社：PIAPRO. http://piapro.jp/
4) Lifelong Kindergarten Group at the MIT Media Laboratory: Scratch. http://scratch.mit.edu/
5) 株式会社カヤック：wonderfl. http://wonderfl.net/
6) クックパッド株式会社：COOKPAD. http://cookpad.com/
7) 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).
8) 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).
9) 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).
10) 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).
11) 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，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).
12) 吉川祐輔，宮下芳明：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，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ョン研究会報告， Vol.2010, No.4 (2010).
13) 木村　誠，土井晃一郎，山本章博：構造を持つ楽曲データを対象とした質問学習にもとづく楽曲生成，情報処理学会研究報告. [音楽情報科学], Vol.2008, No.89, pp.27–32 (2008).
14) 佐藤　剛，宮下芳明：Seek Rope：曲げて切って結べるシークバー，インタラクション 2010 論文集 (2010).
15) 槙野理恵，和泉憲明，小林一郎，橋田浩一：レシピの構造を反映したメタデータに基づく部分レシピの再利用法，人工知能学会第 19 回セマンティックウェブとオントロジー研究会，pp.SIG–SWO–A802–02 (2008).

質 疑 応 答

Q 現在のグラフの問題点として，アウトプットが一つしか無いのは問題ではないか．例えば，椎茸の戻し汁を取っておく，というのは書けないのか．

A アウトプットを２つ以上書ける機能は今のところない．

Q レシピの入れ替え可能な部分をある程度判定できないか．

A 入れ替え可能な部分をコンピュータに判断させるのは少し難しい．或いは，逆にミスマッチなものを組み合わせてそれが美味しかったら，それはそれで良いのではないか．

Q 例えば「切る」というのでは，たくさん切り方があると思うが，それはどういったようにビジュアライズするのか．

A 「切る」のビジュアライズについてはこれから考えていく．

Q ツリーの構造を持っているからツリーで表現すればいい，というのは早計である．文章も本来ややこしい構造を持っているが，最終的にそれを一次元に落とすことによって我々は読めるようになっている．レシピもそうではないのか．グラフで書けば分かりやすいのだからそれで良い，というのは物事の片面しか見ていないのでは．

A 実際の調理は複雑な工程を持っており，別の良い表記方法があればそちらも取り入れていきたいと考えている．

Q 文章で書かれたレシピには多くの情報が隠されている．どのレベルの人を対象とするのかによって，レシピの書き方は変わる．これはどの層をターゲットにしているのか．

A レシピ通りに料理を作れるレベルで，これから新しい料理に挑戦しようという層を考えている．

Q 新規性はどこにあるのか．レシピの表現形式の研究は今までにもたくさんやられている．

A 例えば，フローグラフを用いたレシピサービスや料理支援システムはすでにある．これらでは，既存のレシピを持ってきて組み合わせたりするのは難しい．そういったことができる，レシピをプログラミングする環境を作っていきたいと思っている．