

調理支援システム Cooking Studio

レシピのシナリオ化と調理スケジューリング手法

高野 哲郎[†] 上島 紳一[†][†] 関西大学大学院 総合情報学研究科 〒 569-1095 高槻市霊仙寺町 2-1-1E-mail: [†]t-takano@mtj.biglobe.ne.jp, ^{††}ueshima@res.kutc.kansai-u.ac.jp

あらまし 本稿では、生活に密着した Web データの 1 つであるレシピを対象として、料理初心者に対して「調理の手本」を示す調理支援システム Cooking Studio を提案しその構成法について述べる。初めに、レシピの構造解析を行い、システムにとっての調理の基本的な指示書となる調理シナリオの生成手順について述べる。次に、ユーザ毎の調理環境に基づいた調理手順の導出のためのスケジューリング手法について述べる。最後に、調理手順の情報を付与した調理シナリオを基に調理シミュレーションを行う調理支援システム Cooking Studio について述べる。本システムでは、調理スケジューリングで得られた調理手順を時間軸上にマップして表示している。また、これをそれぞれの工程における調理の様子を同期させることで、利用者にとって直感的に理解し易いインタフェースとしている。

キーワード レシピ, シナリオ, 構造化, スケジューリング, 調理支援システム

Cooking Studio: Cooking Simulation from Web Recipes

Generation of Scenario and Cooking Scheduling

Tetsuro TAKANO[†] and Shinichi UESHIMA[†][†] Graduate School of Informatics, Kansai University

2-1-1 Ryozenji, Takatsuki-City, Osaka, 569-1095, JAPAN

E-mail: [†]t-takano@mtj.biglobe.ne.jp, ^{††}ueshima@res.kutc.kansai-u.ac.jp

Abstract This paper proposes a cooking support system Cooking Studio for cooking novices. Our system first collects recipes from the Web, analyze them morphologically and semantically to generate Cooking Scenario. Next we describe a scheduling method to derive cooking procedure based on each user's cooking environment. We also show a prototype system of Cooking Studio. Our system displays a cooking procedure expressed in a time chart as the result of our cooking scheduling in user's own kitchen, as well as instructions, figures, and movies. We believe Cooking Studio provides helpful GUI for cooking novices.

Key words recipe, scenario, structuring, scheduling, cooking support system

1. はじめに

近年、Web 上には様々な料理のレシピが存在している [1] [2]. レシピには料理に必要な材料と調理法が簡潔に記載されている。しかし、経験や知識が不足している料理初心者にとっては、調理工程を想像するには十分な情報を与えているとは言えない。

また、レシピで記述されている調理手順はレシピは個々の調理者の調理環境に基づいたものではない。つまり、レシピには調理場所や道具など制約がある場合や複数の料理を並列に調理する場合における調理の段取りに関する情報が含まれていない。

料理初心者はこれらのレシピに含まれない情報を読み取るこ

とができないために調理の失敗に陥り易いと考えられる。そこで本稿では、初心者に対して「調理の手本」を示す調理支援システム Cooking Studio を提案を行い [3], 本システムの構成法について述べる。本システムでは、生活に密着した Web データの 1 つであるレシピデータを対象として扱う。

初めに、調理工程を明示的に示す情報として調理シナリオを定義する。調理シナリオは、本システムが記述内容通りに動作することで調理をシミュレートすることを目標とした XML 文書である [4]. ここでは、料理の基本データとなるレシピを Web から収集し、レシピの構造解析を行い、システムにとって調理の基本的な指示書となる調理シナリオの生成手順を述べる。

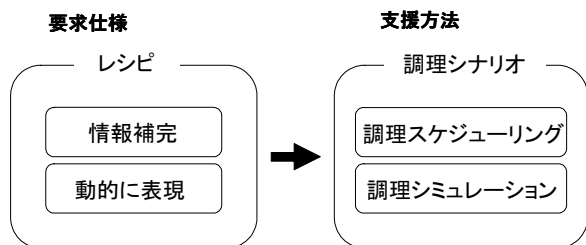


図1 要求仕様と支援方法

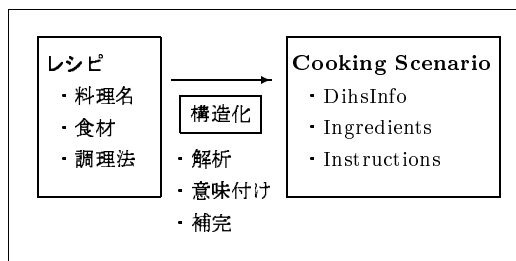


図2 レシピの構造化文書 Cooking Scenario

次に、任意の調理環境における調理手順の導出法について述べる。本手法では、調理シナリオ上の粒度を単純工程とし、調理スケジュールリングを行う。これにより、調理時間の短縮を図り、料理の特性に応じた調理手順の発見が可能となる。

最後に、現在構築中の Cooking Studio のプロトタイプシステムについて述べる。本システムは前述の調理シナリオ通りに動作をすることで、人間が調理を行うのと同様の振る舞いや結果を導く。システム利用者はシステムが行う調理の振る舞い、つまり「調理の手本」を参考にして調理を行う。ここでは、調理スケジュールリングで得られた調理手順を時間軸上にマップし表示をすることで、利用者にとって直感的に理解し易いインタフェースとしている。

2. 研究動機

2.1 料理レシピの特徴

一般に、調理を行う際に基本となるデータはレシピである。レシピは料理を作るために必要な材料とその調理法についての情報が簡潔に書かれている。また、レシピでは調理の動作を表すために料理特有の用語を用いる場合がある。例えば、食材の切り方を表す用語に「乱切り」、「針切り」、「千六本」等がある。

しかし、レシピ中にはその詳細についての記述は存在しない。よって、レシピの字面だけの情報では、例え料理一品を作るだけでも具体的な調理の様子を想像することは難しい。つまり、レシピが持つ情報は、それを読む調理者の経験や知識量によってはその内容を理解できない場合がある。

また、レシピには調理の際に使用する調理器具や道具についても記述されている。しかしこれは、調理者毎の異なる調理環境を考慮した内容ではない。つまり、レシピは個々の調理環境に応じて「いつ、どこで、どの料理についての調理を行えばよいか」という調理のスケジュールリングを与えていない。そのため、調理者自身が個々の調理環境で実現可能な調理のスケジュールリングを決定しなければならない。

2.2 調理支援システム

2.2.1 要求仕様

調理経験や知識の少ない料理初心者がレシピの情報を解釈することや調理の段取りを決めることは困難である。この事は、初心者が料理の失敗に陥り易い原因にもなっている。そこで本稿では、料理初心者に対して「調理の手本」を示す調理支援システムを提案する。本システムの要求仕様を以下に示す。

- レシピから解釈するのが困難な情報を補完する

レシピは簡潔に書かれているため、調理法についての情報が

不足している。また、レシピは個々の調理環境における調理スケジュールリングについて明示的に書かれていない。本システムは、レシピから上記の内容について不足している情報を補う。

- レシピを動的に表現する

Web上のレシピは通常、視覚的に静的なデータとして存在している。本システムは、これを調理の様子を時間経過と共に動的に表現することで利用者にわかりやすい情報の提示を行う。

2.2.2 支援方法

本システムでは、上記の要求仕様を満たすために下記の方法を用いて初心者への支援を行う。

(1) 調理シナリオの生成

本稿では、調理の各動作について明示的に示す情報として調理シナリオを定義する。調理シナリオの生成には、まず、レシピの構造分析を行い、レシピの構造を理解することで不足している情報を特定し、その情報の補完を行う。

(2) 調理手順の導出

本システムでは、実際に調理者が調理を行う環境に基づいた調理手順を導出する。さらに、ここでは調理時間の短縮を目指し、複数の動作を並列処理することで調理を段取り良く行える調理手順を計算する。本稿では、これらの調理特有の制約を考慮に入れた調理スケジュールリング法を提案する。

(3) 調理の様子を可視化

本システムは、上記の調理手順を加えた調理シナリオの通りに動作することで、調理の様子を可視化する。調理の様子を可視化とは、マルチメディアコンテンツを利用して調理のシミュレーションを行うことを指す。

前述の要求仕様と上記の支援方法との対応関係を図1に示す。

3. 調理シナリオ

3.1 調理シナリオの定義

本節では、本支援システムにとっての調理の基本的な指示書となる調理シナリオ Cooking Scenario の定義を行う。レシピの構造化文書である調理シナリオの概念を図2に示す。調理シナリオでは、調理の各動作について明示的に示すために調理スキーマを与える。ここでは、料理レシピの記述内容から推測した文書構造を調理スキーマ(図3)とし、その詳細を下記に示す。

まず、調理シナリオ(Cooking Scenario)は料理に関する情報(DishInfo)とその料理を作るのに必要な材料(Ingredients)とその調理法(Instructions)について記述されている。料理に関する情報とは、料理名(DishName)、料理の種類(Type)と料理の完成イメージ(Image)に関する情報を指す。

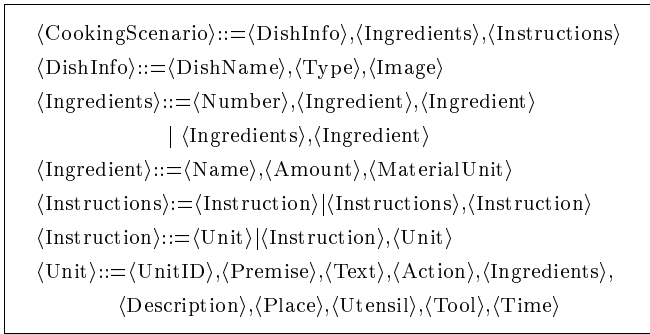


図3 調理スキーマ

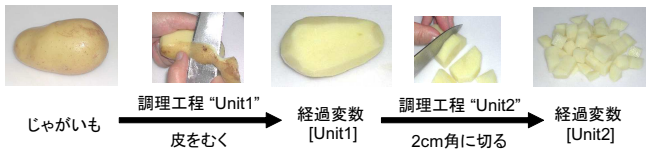


図4 経過変数

次に、材料には何人分の材料であるか (Number) とどの材料 (Ingredient) を扱うか、そして、各々の材料は材料名 (Name) と数量 (Amount) と単位 (MaterialUnit) が書かれている。

調理法はレシピでは段落に区切って書かれていることが多いので、この段落を調理工程 (Instruction) として扱う。さらに、調理工程には複数の動作が含まれているため、そこから単一の動作を粒度とした工程、つまり単純工程 (Unit) に細分化する。

各単純工程は、「どの材料を使い (Ingredients)、どの容器に入れて (Utensil)、場所はどこで (Place)、どの道具を使い (Tool)、どれだけの時間・どのタイミングで (Time)、どのような (Description) 調理をする (Action)」という情報が含まれている。そして、「単純工程自身を表す ID (UnitID) を持ち、どの単純工程の後ですべきか (Premise)」という情報と、「レシピ中でのその該当する文章 (Text)」もあわせて持つ。

3.2 情報補完

調理シナリオとは、Web データであるレシピを調理スキーマに従った機械可読な形式に変換した後、調理スキーマで定義されている要素のうち不足している情報を補完したものを言う。ここで補完すべき情報は、料理に関する知識を用いることで導くことができるため、レシピ文中では省略されている。

前節から、調理法の最小単位は単純工程であり、この単純工程において不可欠な情報は動作であると言える。また、材料以外の情報は調理動作に基づいて判断できるため、レシピ中では省略されている場合が多い。本稿では、この動作以外の情報を材料とそれ以外の情報 (場所、道具、時間等) に区別し、それぞれの情報を補完する方法について以下に述べる。

3.2.1 経過変数を用いた材料の補完

材料は調理工程を経ることにより、調理開始前に用意された状態から徐々に変化していく。調理シナリオでは、変化する材料の様子を把握する手法として経過変数を導入する。

経過変数とは各単純工程において材料を調理した結果を格納する変数を指す。経過変数の名前には単純工程の ID を使用する。経過変数を材料として扱うことで単純工程間の関係を導き、

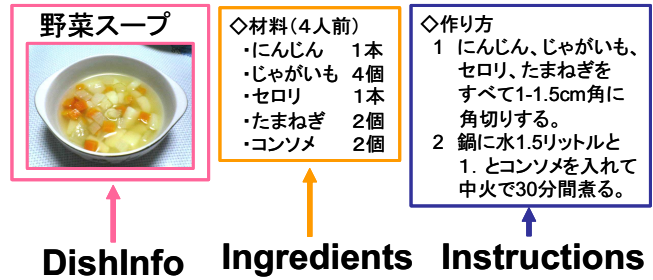


図5 レシピからの要素の抽出 (Step 1)

表1 原材料変数の割当 (Step 3)

原材料変数	Name	Amount	MaterialUnit
m1	にんじん	1	本
m2	じゃがいも	4	個
m3	セロリ	1	本
m4	たまねぎ	2	個
m5	コンソメ	2	個
m6	水	1.5	リットル

注：m6 は Step 4 より割当られる

調理全体の流れを把握することが可能となる。

【例】「じゃがいもは皮をむいて、2cm 角に切ります」という調理工程を例に説明する (図4)。この工程は、「皮をむく」という単純工程 "Unit1" と「2cm 角に切る」という単純工程 "Unit2" という2つの単純工程から構成されている。まず、「Unit1」で扱う材料は「じゃがいも」であるので、その調理の結果として「皮をむいたじゃがいも」が生成される。そこで、この「Unit1」の調理結果を経過変数 [Unit1] に代入する。

次に、「Unit2」で扱う材料は、工程文中では明記されていないが、文脈より「皮をむいたじゃがいも [Unit1]」であると判断できる。よって、「Unit2」の材料は [Unit1] とする。「Unit2」によって「2cm 角に切った [Unit1]」が生成される。これは「皮をむいた後に2cm 角に切ったじゃがいも」であることを表す。

また、「Unit2」の材料に [Unit1] を扱うことから、「Unit2」を行う前に「Unit1」を行う必要があることが分かる。このように、経過変数を用いることで単純工程間の関係が明らかとなる。■

3.2.2 動作と材料以外の情報の補完

単純工程において調理器具や道具、場所や時間等といった (動作と材料以外の) 情報は、その調理動作から判断できる。調理シナリオでは上記の情報が記述されていない場合は、動作から導き出される値を補完すべき情報とする。

3.3 調理シナリオの生成

本節では、Web データであるレシピから調理シナリオを生成する方法について述べる。

Step 1. レシピからの要素の抽出

まず、レシピから調理スキーマの要素である (DishInfo)、(Ingredients)、(Instructions) を抽出する。要素の抽出には、レシピが書かれている HTML 文書中の出現位置を利用する。図5に野菜スープのレシピから要素を抽出する様子を示す。

Step 2. 料理情報の解析

Step 1 で抽出した要素のうち、料理自身についての情報を分

表 2 Cooking Table(Step 4,5)

UnitID	Action	Ingredients	Description	Place	Utensil	Tool	Time
Unit1	切る	m1,m2,m3,m4	1-1.5 cm 角に	(調理台)	(まな板)	(包丁)	(*3)
Unit2	入れる	Unit 1,m5,m6	(*1)	(調理台)	(鍋)	(*2)	(*3)
Unit3	煮る	(Unit2)	(*1)	(コンロ)	(鍋)	(*2)	30 分

(): 補完によって得られた情報 *1: 修飾語が空値 *2: 道具がいない場合は空値 *3: 動作に相応な時間を表す

析する。調理スキーマより (DishName), (Type), (Image) を抽出する。ここで、(Type) は料理名や調理法などから判断し、「煮物」「炒め物」「パスタ料理」等の値を入れる。図 5 の場合では、(DishName) は「野菜スープ」、(Type) は「煮物」、そして (Image) は「画像ファイルの URL」を抽出する。

Step 3. 原材料変数の割当

Step 1 で抽出した要素のうち、材料について分析を行う。まず、各材料に対して原材料変数を与える。次に、調理スキーマに従って、(Name), (Amount), (MaterialUnit) を抽出する。図 5 の Ingredients より要素を抽出した結果を表 1 に示す。

原材料変数は調理により加工されていく材料の状態を把握するため、「その材料は調理開始前に準備された状態の物である」ことを示す。また、前述のように加工された材料は経過変数を用いて表す。調理シナリオでは材料をこの 2 種類の変数を用いて表現し、調理法の最小単位である単純工程間の関係を導く。原材料変数の名前は m_i (i は自然数の添え字) と記述する。

Step 4. 単純工程の導出アルゴリズム

次に、調理法から単純工程を導出する。調理法の最小単位である単純工程毎に、調理法に関する情報を抽出する。また、単純工程において不可欠な情報は調理動作であることから、調理動作を基準として単純工程を導出する。

4-1. 単純工程における調理動作の抽出

調理スキーマより、図 5 にある (Instructions) の各段落 (Instruction) に対して形態素解析を行い、調理の動作を表す「動詞」の基本形を単純工程 (Unit) の (Action) とする。

調理スキーマの要素である (Action) は動詞、(Place), (Utensil), (Tool), (Time) は名詞である。よって、これらの要素を導くために形態素解析を行う。ここでは形態素解析ツールとして茶筌 [5] を利用し、(Instruction) を入力して基本形と品詞を出力させる。

4-2. Cooking Table の生成

次に、調理スキーマに記述された単純工程の要素となる情報を抽出し、それを Cooking Table の値とする。Cooking Table は表 2 で示すような単純工程毎の情報を表した表である。

Cooking Table の値にはレシピに記載されている情報のみを利用する。これは、Step 4-1 で行った形態素解析の結果から表の該当部に言葉を埋める。Cooking Table のそれぞれの項目の値に関しては以下のように抽出する。

・「材料」(Ingredients)

単純工程で扱われた材料を原材料変数または経過変数を用いて表す。ここで、調理法の文中には材料で書かれていなかった食材が登場する場合がある。その食材についても Step 3 と同様に原材料変数を割り当てる。

図 5 の (Instructions) 中の 2 つめの (Instruction) を例に挙げる。文中の「1.」は前の (Instruction) で加工した材料全体を表している。よって、扱う材料には経過変数 [Unit1] を与える。

単純工程の動作の内容によっては材料が 2 つ以上必要な場合もある。例えば、「混ぜる」という動作では目的語を 2 つ以上持つ。そのため、「m1 を混ぜ」と混ぜる材料が 1 つしか書かれていない場合には、原則としてその直前の単純工程による調理結果、つまり経過変数を加えることとする。

・「修飾語」(Description)

「動作」に関する修飾語をこの部分に記入する。

・「場所」(Place), 「器具」(Utensil), 「道具」(Tool)

各単純工程中の名詞を以下の項目に振り分ける。

「場所」: 調理場の中で動作を行う場所、もしくは機械

「器具」: 加工する材料を入れたり置いたり包んだりする物

「道具」: 調理するために使う道具 (手の場合もある)

・「時間」(Time)

単純工程中の時間 (例: 20 秒)、もしくはその動作を行う時間的条件 (例: きつね色になるまで) を書く。「まで」と入っている文節、助動詞「たら」の入った文節などはこの項目に記入する。

最後に、残った単純工程句中の不必要な言葉 (読点・句点や助詞・助動詞等) を削除する。上記の要素に該当する言葉は数百数千語も存在しないと予想される。そのため、事前に辞書を用意しておけば自動判別も可能であると考えられる。

Step 5. 調理法に関する情報の補完

Cooking Table の値は調理工程に関する最小限の情報と定義している。そのため、調理を行うにはこの表の全項目に関する情報を完全に把握する必要がある。

ここで補完すべき情報とは Cookint Table 中で空値となっている項目である。ここでは、3.2 の手法に基づきそれぞれの項目が空値の場合に情報を補完する流れについて述べる。

・「材料」の補完

何らかの調理を行うということは必ず 1 つ以上の材料を扱うので、どの単純工程においても「材料」は空値であってはならない。しかし、単純工程で扱う材料はその文中に明示的に書かれていない場合が多くある。この場合、3.2.1 より経過変数を単純工程で扱う材料とする。

材料として扱う経過変数は以下のように特定をする。例えば、材料について記述がない単純工程 "UnitN" の動作を表す動詞が連用形であれば、直前の単純工程 "Unit(N-1)" が何らかの関係を持つ、つまり連続して調理が行われていると判断する。よって、"UnitN" では経過変数 [Unit(N-1)] を扱う材料とする。

・「修飾語」の場合

修飾語は空値でも構わない項目である。

表 3 調理シナリオの例 (Step 1)

料理名	工程 ID	材料	動作	場所	時間
スパゲティ	Unit1	m1	茹でる	コンロ	10
	Unit2	m2,m3	切る	調理台	2
	Unit3	Unit2,m4	混ぜる	調理台	2
	Unit4	Unit1,Unit3	混ぜる	調理台	3
	Unit5	Unit4,m5	和える	調理台	2
スープ	Unit1	m1	切る	調理台	2
	Unit2	Unit1	煮る	コンロ	30
サラダ	Unit1	m1,m2	切る	調理台	2
	Unit2	Unit1	混ぜる	調理台	2
	Unit3	Unit2,m3	和える	調理台	3

・「場所」「器具」「道具」の補完

それぞれの値は「動作」と「材料」によって決定する。場所に関しては、主として焼いたり煮たりする場所は「コンロ」、食材を水で洗う場所は「流し」となる。また、器具や道具は「切る」という動作の場合、器具には「まな板」を、切る道具には「包丁」を用いる。

・「時間」の補完

「動作に相応な時間がかかる」という意味の値を補う。

表 2 の Cooking Table に情報補完により与えられた値を () を用いて表す。

Step 6. 調理シナリオの生成

Step 1-4 で得られた値や Step 5 で補完を行った Cooking Table の値を基に調理シナリオを生成する。調理シナリオは書式として調理スキーマに従い、XML を用いて記述を行う。

4. 調理スケジューリング

4.1 調理の制約

調理者は並列的に調理を行い効率化を図る場合や、複数の料理を調理する場合、以下の制約を考慮に入れる必要がある。

● 調理場所の制約

調理人数、コンロやオープンの数が該当するまた、コンロやオープンは並列処理が可能な場合がある。

● 時間の制約

各料理ごとに調理をする工程の順番は決められている。また、料理によっては前後する工程を連続的に行う場合や、複数の工程について終了する時間の同期をとる場合も存在する。

● 料理自身の持つ制約

料理の種類によって、「煮物は最初に」、「パスタ料理は最後に」等の調理する順番を決める必要がある。

4.2 調理スケジューリング

4.2.1 スケジューリングポリシー

本稿では、上記の制約を考慮に入れた調理スケジューリング手法を提案する。本手法では、調理手順の決定に必要な情報を抽出するために調理シナリオを対象データとする。

本手法は以下に示す 2 つの原則に従いスケジューリングを行う。1 つは、調理時間の短縮を図るため、できるだけ短い時間で全ての料理の調理を行う。もう 1 つは、全ての工程を直列的に調理せずに可能な限り並列的に調理を行う。

表 4 工程表の例 (Step 1)

料理番号	全工程 ID	先行工程	動作の特徴	場所	時間
Dish1	A	-	parallel	Stove	10
	B	-	series	Dresser	2
	C	B	series	Dresser	2
	D	A, C	synchronize	Dresser	3
	E	D	series	Dresser	2
Dish2	F	-	series	Dresser	2
	G	F	parallel	Stove	30
Dish3	H	-	series	Dresser	2
	I	H	series	Dresser	2
	J	I	series	Dresser	3

4.2.2 スケジューリングアルゴリズム

本節では、上記の制約を満たしながら各単純工程を時間軸にマップするアルゴリズムについて述べる。これにより、任意の調理環境における調理スケジューリングを導く。

Step 1. 調理グラフの生成

調理シナリオから料理の各単純工程の順序関係を表す調理グラフを生成する。調理グラフの定義は以下の通りである。

定義 調理グラフはノードとエッジからなる非巡回的な有向グラフである。ノードは変化する材料の状態を表す。また、複数の料理を調理する様子をグラフで表現するため、Start ノードと Goal ノードを用いる。調理グラフ全体から見て、先端である Start ノードは調理全体の開始時点、末端の Goal ノードは全ての料理が完成した状態を表す。Start ノードからは各料理の調理開始ノードへエッジを引き、各料理の完成状態を表すノードから Goal ノードへエッジを引く。

エッジは単純工程を表し、エッジのラベルには単純工程の内容を示す。有向エッジは 2 つのノードの前後関係を表し、方向は前方に位置づけられたノード (材料) から、調理を行い後方のノード (加工された食材) へ向かうものとする。■

次に、調理シナリオから調理グラフを生成するアルゴリズムについて述べる。調理グラフを描くには、ノードとエッジの接続関係を示す情報が必要である。また、エッジのラベルとして「工程 ID」「動作の特徴」「場所」「時間」に関する情報を持つ。これらの情報を調理シナリオから抽出する。

例として、複数品の料理を調理する場合を想定し、表 3 に示す調理シナリオからグラフを生成する。まず、ノードとエッジの関係を把握するために工程表を用いる。工程表は調理グラフのエッジのラベルが持つ情報を値をまとめた表である。

また、工程表では調理スケジューリングを行うために、各料理の工程の順序関係を導く。表 3 より生成した工程表を表 4 に示す。工程表の各項目とその値については以下の通りである。

[料理番号] … 調理する料理の ID を表す。ここでは、表 3 で書かれている料理の順に Dish1, Dish2 … と ID を付与する。

[工程 ID] … 全ての料理において一意である工程 ID を表す。Dish1 の最初の単純工程から A, B … と ID を付与する。

[先行工程] … その工程を調理する直前に行うべき工程を表す。調理シナリオで材料として扱う経過変数から以下のように導く。

【例】表 3 の料理名「スパゲティ」、工程 ID「3」の単純工程を例

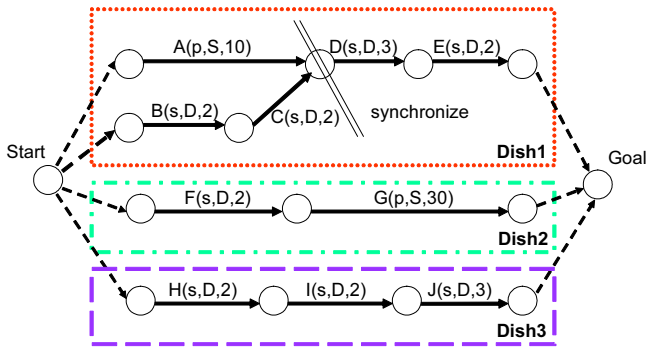


図6 調理グラフ (Step 1)

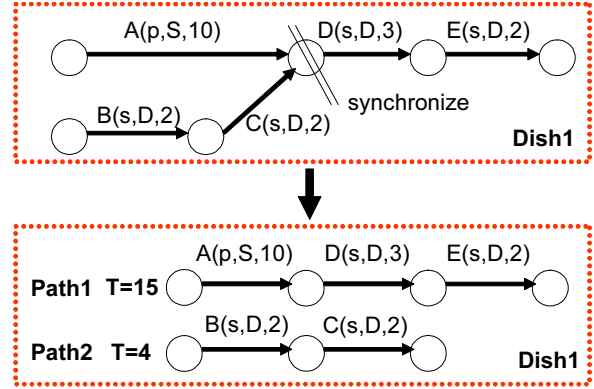


図7 調理グラフのパス分割 (Step 2)

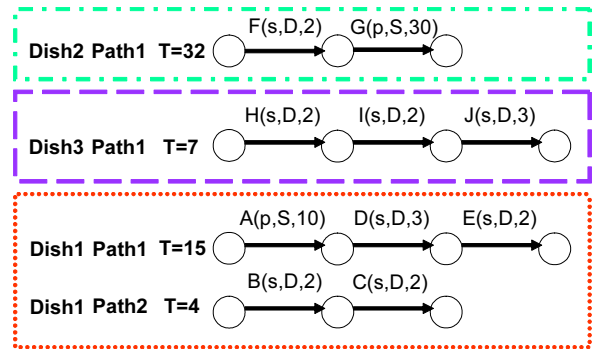


図8 順序割付後のパスリスト (Step 3)

に挙げて説明する。この単純工程で扱う材料は”Unit2,m4”と書かれている。これは、Unit2 で調理を行った材料を Unit3 で扱うことを意味する。つまり、Unit3 で調理を行うためにはその直前に Unit2 を行う必要がある。この関係より、表4の工程ID”C”では、先行工程”B”を値として持つ。■

[動作の特徴] … 調理を行う場所が「コンロ」「オープン」、動作が「煮る」「茹でる」等の並列的な調理が可能な場合、この項目に”parallel”の値を代入する。また、複数の先行工程がある場合は、それぞれの工程の終了時間の同期をとることを表す”synchronize”を値として代入する。上記に該当しない調理工程（表4中の値は”-”）では、直列的に調理を行う。

[場所・時間] … 単純工程を行う場所や時間を表す。ここでは、調理シナリオに書かれている値を代入する。

次に、作成した工程表から調理グラフを作成する。工程表の先行工程よりノードとエッジの接続関係を導く。調理グラフの各エッジのラベルには”工程ID（時間、場所、調理の動作）”を記述する。また、動作の特徴である”synchronize”を表すために、そのエッジの始点ノードに二重線を引く。表4より作成した調理グラフを図6に示す。

Step 2. パスの分割

本手法では、Step 3 で示すように料理の種類によって調理を行う順序を割り付ける。順序を割り付ける単位としてパスを用いる。パスとは、調理グラフの連続するエッジにおいて最大の調理時間を含む分岐のない部分グラフを指す。また、部分グラフ化を行うのは、同一料理において並列化して調理を行う場合にどのパスから調理を行えば良いかを計算するためである。

ここでは、Step 1 で作成した調理グラフを対象として、同じ料理において分岐がなくなるまでパスの分割を繰り返す。パスに分割する際には、調理グラフから該当するエッジを除去する。図6内のDish1においてパスの分割の結果を図7に示す。また、こうして得られたパスをリスト化しておく。

Step 3. 順序割付

Step 2 で作成したパスに対して料理の種類に応じた調理を行う順序を割り付ける。順序割付のルールを以下に示す。

・順序割付ルール

1. 料理の種類により調理の順番を決める

ここでは、料理の種類を「煮物・冷し物」「anytime」「焼き物・炒め物」「麺類（パスタ）」の4種類とし、調理の順番をこ

の通りとする。これは、それぞれの料理を一番おいしい状態で食べられるようにそれぞれの順番を設定した。「煮物・冷し物」は、高・低温状態で長時間調理を続けるために一番最初に調理を行う。一方、麺がのびてしまうため「麺類（パスタ）」を扱う料理は全調理の最後に調理する。また、「焼き物」「炒め物」は料理が温かいままで食べるために「麺類」の前の順番とした。最後に、これら以外の種類の料理は調理時間を選ばないため、その種類を「anytime」とした。

2. 調理時間が長い料理から先に調理を行う

同じ種類の料理が複数ある場合は、これは、先に完成する料理がもう一方が完成するまでに待つ時間を短くするために、調理時間が長い料理（あるいはパス）から先に調理を行う。

上記のルールに従い、Step 2 で得られたパスリストに対して順序割付を行ったパスのリストを図8に示す。

Step 4. 資源割付

ここでは、資源割付アルゴリズムにより Step 3. で割り付けたパスの順序に基づき各エッジに対して資源を割り付ける。資源とは、調理における資源として「調理時間」と「調理場所」を指す。また、各エッジに資源を割り付けることで、各単純工程は「調理開始時間」「調理終了時間」「調理場所」という調理手順に関する情報が取得できる。本手法で用いる資源割付アルゴリズムを以下に示す。

・資源割付アルゴリズム

1. 調理環境における調理場所を特定する

個々の調理環境において存在する調理場所（「調理台」「コンロ」「オープン」）を特定する。それぞれの調理場所は「現在使

表 5 Time Table(Step 4,5)

工程 ID	調理開始時間	調理終了時間	調理場所
F	0	2	Dresser
G	2	32	Stove1-Parallel
H	3	5	Dresser
I	5	7	Dresser
J	7	10	Dresser
A	10 → 17	20 → 27	Stove2-Parallel
B	11 → 23	13 → 25	Dresser
C	13 → 25	15 → 27	Dresser
D	20 → 27	23 → 30	Dresser
E	23 → 30	25 → 32	Dresser

注：矢印左が修正前，右が修正後の調理時間を表す

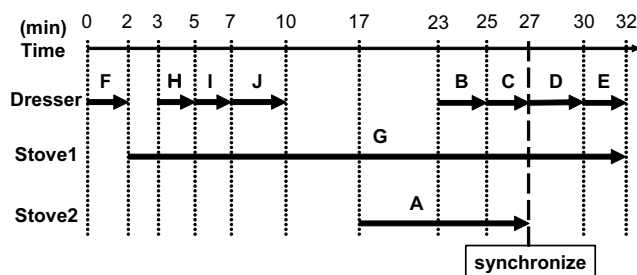


図 9 タイムチャート (Step 6)

具体例として表 5 の Time Table にて該当する調理時間について修正を行い，その様子を矢印を用いて示す。

Step 6. 調理タイムチャートを生成

Step 4. でエッジに与えられた「調理開始時間」「調理終了時間」「調理場所」を視覚的に表すために調理タイムチャートを生成する。調理タイムチャートは，調理場所と調理時間をそれぞれ縦軸，横軸にとり，各エッジを該当する位置に配置する。同期を取るべき単純工程は，縦線を引くことでそれを表す。図 8 より生成した調理タイムチャートを図 9 に示す。

5. 調理支援システム Cooking Studio

5.1 システム概要

本章では，料理初心者が調理の際に直面する問題の解消を目的とした調理支援システム Cooking Studio を提案する。またここでは，Cooking Studio のプロトタイプシステムについて述べる。本研究では，初心者に対して「調理のお手本」を示し，初心者が調理をする際に利用できるシステムの実現を目指す。

本システムは前述の調理シナリオ通りに動作をすることで，人間が調理を行うのと同様の振る舞いや結果を導く。システム利用者はシステムが行う調理の振る舞い，つまり「調理の手本」を参考にして調理を行う。ここでは，調理スケジューリングで得られた調理手順を時間軸上にマップし表示をすることで，利用者にとって直感的に理解し易いインタフェースとしている。

本研究で作成したプロトタイプシステムの構成を図 10 に示す。本システムの主な流れは以下の 2 通りである。

• Generating Scenario

Web のレシピデータを対象として調理シナリオを生成する。この際，3 章で示したように，レシピの解析・構造化・情報補完を行い，調理シナリオを生成する。生成した調理シナリオは Scenario DB に格納する。

• Cooking Simulation

初めに，システム利用者が調理する料理を選び，その料理を調理する様子をシミュレートする。まず，DB から調理する料理に該当する調理シナリオを取得する。次に，調理シナリオを用いて調理を行う環境に応じた調理スケジューリングを行う。そして，本システムは調理スケジューリングにより得られた手順通りに調理シミュレーションを行う。この時に，シミュレーションの様子を GUI に表示する。

5.2 システムの利用

本節では，利用者がシステムを利用する流れについて述べる。

用中」か「使用可能」のどちらの状態であることを示す値を持つ。

2. 時間 T において調理可能なエッジを探す

時間 T (調理開始時点から経過した時間，初期値 = 0) において，(1) 先行工程がない，もしくは全て終了している，(2) 既に調理開始時間が与えられていない，(3) パスの優先度が一番高い，これら 3 つの制約を満たすエッジがあれば，3. へ進む。全てのエッジがこのルールに該当しない場合，全てのエッジに調理開始時間が与えられていれば終了する。調理開始時間が与えられていないエッジがあれば，一番早く未使用となる調理場所が使用可能となる時間まで T を進める。

3. 調理場所の使用状況を確認する

2. で求めたエッジのラベルに書かれている調理場所が時間 T において使用可能であれば 4 へ進む。調理場所が使用中であれば 2. に戻り，別の調理可能なエッジを探す。

4. 動作が並列化が可能か判定する

3. で求めたエッジに調理開始時間 T を，調理終了時間として時間 T+t (t=エッジに書かれた調理時間) を与える。また，このエッジが並列的に調理を行えるならば，調理場所を使用中として時間 T を 1 分進めて 2. へ戻る。直列的に行う調理工程であれば，時間 T を T+t まで進めて 2. に戻る。

本手法により得られた単純工程毎の調理開始・終了時間をまとめた表を Time Table として表 5 に示す。

Step 5. 調理時間の修正

Step 4. では，「全ての単純工程は資源に空きがあればすぐに調理を行う」というスケジューリングポリシーのもとで調理手順を決定した。しかし，上記の方法では複数の工程について終了する時間の同期をとる場合，同期を取るべき単純工程について終了時間が揃わないため，対応ができない。

また，「麺類 (パスタ)」の料理を調理する際にはその料理が完成する時点で全部の調理が終了することが望ましいが，前述の調理手順の決定方法ではこの条件を満たすことができない。

そこで，Step 4. でエッジに与えられた「調理開始時間」ならびに「調理終了時間」について，全料理の終了時間から開始時間へと遡って上記の条件を満たすように修正を行う。

ここでは，同期を取るべきエッジは調理終了時間が遅い方に揃える。また，「麺類」の料理のエッジについては，可能な限り調理終了時間に料理が完成するように遅い時間へずらす。

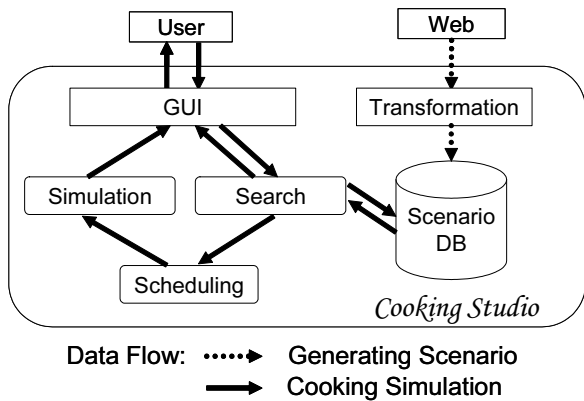


図 10 システム構成図

本システムが表示する画面を図 11 に示す。まず、利用者は調理を行いたい料理を GUI にて選択する。本システムは、選択された料理について、調理手順を計算し、その結果を用いて調理シミュレーションを行い GUI に表示する。

図 11 の調理シミュレーション画面では、右側に調理の段取りをタイムチャート化したものを表示している。タイムチャートでは、横軸に調理場所、縦軸に時間を取り、各調理工程がいつどこで行うべきかを表す。また左側には、現在行うべき調理工程の様子を図やアニメーション、文字を用いて表示する。調理シナリオは XML で記述し、形態素解析ツールとして茶筌[5]を利用し、プロトタイプシステム本体は macromedia Flash MX [6] を用いて構築を行った。

6. 関連研究

関連研究としては、以下の研究が挙げられる。まず、濱田ら[7] は料理教材をもとにした調理手順の構造解析手法を提案している。ここでは、料理の調理法について辞書を用いて解析し、構造解析の結果をグラフで表現している。また、これらは料理番組に付随した教材テキストを対象としており、料理映像とレシピの対応付けることでマルチメディア料理教材として統合することを目的としている[8]。これに対して、本システムでは、Web 上のレシピデータを対象としており、調理工程についてのスケジューリングを行いながら仮想空間においてレシピの可視化を行うことを目的としたシステムとなっている。

一方、Web データの活用法として、灘本ら[9] は Web 情報の受動的な視聴を目的とした Web 情報の番組化を行っている。ここでは、静的な Web 情報の文書構造を利用して動的な番組という形式に変換するアプローチをとっている。本研究では、Web データの文書構造を利用するだけでなく、文書が表す意味を解釈した上で必要な情報の可視化を行っている。

また、エージェントと利用者のインタラクション設計言語として石田ら[10] が Q の提案を行っている。ここでは、設計言語 Q を 3D 仮想空間でのシミュレーションに応用している。本研究においても料理世界の言語として調理シナリオを定義し、Web 上にあるレシピデータをもとに調理シナリオを生成した。また、調理シナリオの解釈実行系として調理支援システム Cooking Studio の構築を実現することを目指している。

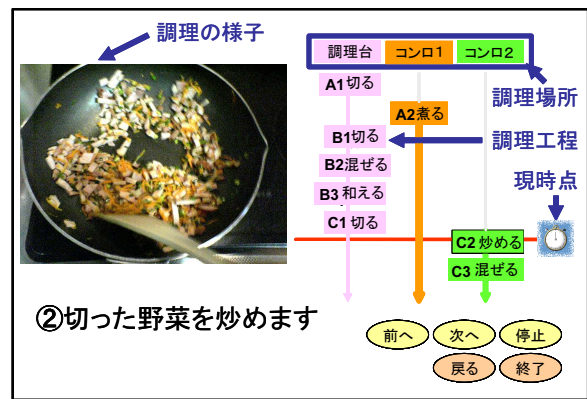


図 11 システム画面 調理シミュレーション画面

7. おわりに

本稿では、レシピの調理シナリオ化とその実例として調理支援システム Cooking Studio の提案を行った。料理初心者が調理の際に直面する問題に対し、本システムの利用によって情報の面での解消を目指している。本研究における今後の課題として、

- (1) HTML で記述されたレシピデータから調理シナリオの効率的な生成法
 - (2) 調理スケジューリングの検証
 - (3) 料理初心者が調理しやすいインタフェースの検証
- などが挙げられる。

今後、このような Web 適応技術の発展に伴い、人と機械による協調作業を行う機会が増えると思われる。

文 献

- [1] 3分クッキング, <http://www.ntv.co.jp/3min/>.
- [2] おかずのクッキング, <http://www.tv-asahi.co.jp/okazu/>.
- [3] 高野哲郎, 上島紳一, "調理支援システム Cooking Studio の提案", The Second International Conference on Creating, Connecting and Collaborating through Computing, 7B-3, 2004.
- [4] 高野哲郎, 上島紳一, "Cooking Scenario -レシピの Scenario 化とその応用-", DBWS2003, 1A-4, 2003.
- [5] 形態素解析システム 茶筌, <http://chasen.aist-nara.ac.jp/>.
- [6] macromedia FLASH MX, <http://www.macromedia.com/jp/>.
- [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 Vol.J85-D-II, No.1 pp79-89 2002 年 1 月.
- [8] 三浦宏一, 高野 求, 浜田玲子, 井手一郎, 坂井修一, 田中英彦, "料理映像の構造解析による調理手順との対応付け", 電子情報通信学会論文誌 Vol.J86-D-II, No.11 pp1647-1656 2003 年 11 月.
- [9] 灘本明代, 服部多栄子, 近藤宏行, 沢中郁夫, 田中克己, "Web コンテンツの受動的視聴のための自動変換とスクリプト作成マークアップ言語" 情報処理学会論文誌: データベース, vol.42, No.SIG1(TOD8), pp.103-116 2001.
- [10] 石田 亨, 福本理人, "インタラクション設計言語 Q の提案" 人工知能学会論文誌 vol. 17, No. 2, pp.166-169, 2002.