

RFID を用いたインテリジェント冷蔵庫システムのプロトタイピング

金野 紋子[†] 増永 良文[‡]

[†]お茶の水女子大学大学院 人間文化研究科博士前期課程 数理・情報科学専攻

[‡]お茶の水女子大学 理学部情報科学科

〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†] ayako@db.is.ocha.ac.jp, [‡] masunaga@is.ocha.ac.jp

あらまし 近年、ユビキタス社会を実現するに当たり、情報家電への注目が高まっている。我々は、冷蔵庫に注目し、冷蔵庫に入れる食品のデータをデータベースに格納することにより、従来の冷蔵庫が単に食品を貯蔵、冷却するという役割に加え、冷蔵庫の中身に関して、たとえば、外出先から携帯電話を使って食品の残量、賞味期限が近いものはどれかといった情報を確認できたり、残り物でできるレシピを教えてくれるなどの機能をもったインテリジェント冷蔵庫システムの開発に取り組んできた。本システムは、ユーザビリティにも重点を置き、食品の特定に RFID、食品の重さ取得に電子天秤を用いている。本稿では、考案したプロトタイプシステムの概要と実験を行った結果について述べ、データベースと実世界との一貫性が保持されない場合の改善策など、システム評価について報告する。

キーワード ユビキタスコンピューティング, ユーザインタフェース, RFID, 情報家電, 冷蔵庫

Prototyping of an Intelligent Refrigerator System using RFID

Ayako KONNO[†] Yoshifumi MASUNAGA[‡]

[†] Graduate Division of Mathematics and Computer Science (Master's Program), Ochanomizu University

[‡] Department of Information Science, Faculty of Science, Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: [†] ayako@db.is.ocha.ac.jp, [‡] masunaga@is.ocha.ac.jp

Abstract Recently, in order to realize a ubiquitous society, the research and development of information appliances have been getting more attention. We have focused on the development of an intelligent refrigerator system based on database technology. By storing food data into the database when a user is putting RFID-tagged foods into the refrigerator, the intelligent refrigerator system becomes to have a lot of functions. For example, it functions not only as a cooler or a storage but also for users to know what foods remain in the refrigerator currently, or when the food expiration dates are, or the dishes which a user can cook from the remaining foods, etc. From the aspect of usability, this system uses RFID tags for identifying foods and an electronic balance for getting weight of foods. In this paper, we describe the overview and the result of an experiment of a prototype which we have developed, and report the system evaluation and suggest the improvements when the database causes an inconsistency with the real world objects.

Keyword Ubiquitous Computing, User Interface, RFID, Information Appliance, Refrigerator

1. はじめに

近年、ネットワーク技術の進展に伴い、冷蔵庫、洗濯機、電子レンジ、エアコンなどの家電を情報化することで、それぞれの家事支援機能に加え、家事労働の

軽減、安心などのメリットを提供する情報家電への期待が高まっている。その結果、多くの団体がその取り組みを精力的に行っている[1].

しかし現在のところ、情報家電と言えばネットワー

ク家電中心で、情報家電の実現に必要な基盤技術に対して重点的に取り組まれているが、ユーザの利便性を追求した取り組みにはさらなる発展の余地がある。

そこで、ユーザが家電を使う際伴うあらゆるデータをデータベースに格納することにより、ユーザビリティを高め、ユーザの生活を豊かにする情報家電を開発したいと考えた。我々は、様々な家電の中でも、冷蔵庫は食品の貯蔵庫であり、かつ食生活の中心的役割を担っているという観点からデータベースを活用しやすと考え、特に冷蔵庫に注目してインテリジェント冷蔵庫(i-冷蔵庫)システムを提案する。i-冷蔵庫システムは従来の、単に食品を貯蔵、冷却するという役割に加え、たとえば、冷蔵庫の中身を外出先から携帯電話を使って食品の残量、賞味期限が近いものはどれかといった情報を確認できたり、残り物でできるレシピを教えてくれるなどの機能をもつ。本システムは、ユーザビリティにも重点を置き、食品の特定に RFID、食品の重さ取得に電子天秤を用いる。

我々はこれまで、情報家電の動向を調査し、ユーザの利便性を考慮したインテリジェント冷蔵庫システムのあらゆる可能性を提案した[1]。

2. RFID を用いた i-冷蔵庫システム

2.1. i-冷蔵庫システムの概要

まず、全体的なシステム概要とそれを実現する機器類、設置場所を述べる。

簡単にシステム概要を説明すると、ユーザが冷蔵庫に食品を入出庫する際、RFID リーダ、電子天秤、タッチパネルにより食品データを取得し、データベースにデータを格納・更新を行う SQL 文を発行する(図 1 のデータ格納・更新部)。また、ユーザがタッチパネル上または携帯電話で冷蔵庫の中身、過去の食品履歴、残り物で作れる料理のレシピを閲覧したい時、タッチパネルではボタンを押下、携帯電話では指定の URL にアクセスすることにより、欲しい情報を得ることができる(図 1 のインテリジェント問合せ部)。

図 2 のように、タッチパネルと RFID R/W は冷蔵庫の扉、電子天秤は冷蔵庫の中の食品を入れる籠の下、DBMS を持つ PC は冷蔵庫の背後に配置する。食品を冷蔵庫の中に入れる前に必ず、食品を一つ一つ RFID タグ付きの袋に入れ、袋に入れた食品 i 食品と定義する。i 食品を籠の中に次々と追加していくので、一個当たりの重さは入れる前後の差分で出す。本研究では、食品を使い切ったら、タグを使い捨てるのではなく、新しい食品に使いまわすところもポイントである。

2.1.1. RFID

本システムを実現するために重要な役割をもつ RFID について述べる。RFID (Radio Frequency

Identification) は、ユビキタス・コンピューティングの世界を実現するのに注目され、例えば、流通業界でバーコードに代わる、モノの識別・管理技術としての期待が高い。RFID タグは、主に、非接触の読み取りが可能、耐環境性、情報の書き換え、複数同時認識可能、保持できる情報量の大きさ、という多数の点で、バーコードや 2 次元コードより優れているといえ、現在、Suica や FeliCa など世の中に急速に普及している技術である。

2.1.2. 電子天秤

本システムで用いる電子天秤は島津製作所製で、天秤自身にデータ転送機能を持ち、RS-232C ケーブル 1 本で天秤と PC を接続すると、重さデータを PC に転送することが可能である。

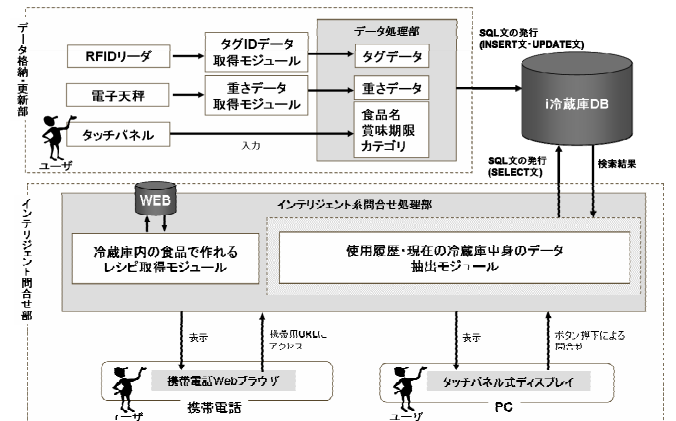


図 1: システム構成図

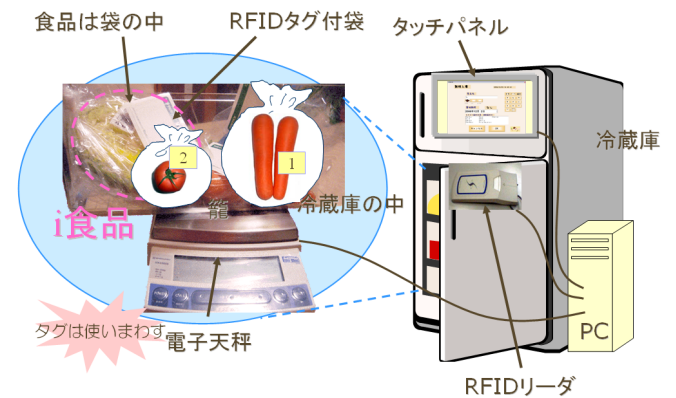


図 2: 機器構成のイメージ図

2.2. i-冷蔵庫システムの要求定義

i-冷蔵庫の機能を UML[2]のユースケース図、ユーザの操作手順をアクティビティ図で示す。

図 3 は i-冷蔵庫システムのユースケース図で、ユーザが i-冷蔵庫に対して接する場合の全てを表している。上の 3 つのユースケースは従来の普通の冷蔵庫を利用する場合と基本的に一緒であるが、i-冷蔵庫システム

の場合、入庫の際、ユーザは多少の入力負担を被る。「冷蔵庫の中身を閲覧する」、「食品の使用履歴を閲覧する」、「残り物で作れる料理のレシピを閲覧する」は i-冷蔵庫システムだからこそ実現できるユースケースである。

次にそれぞれのユースケースに対し、具体的にユーザとシステムが行う一連の手順を、アクティビティ図でもって詳述していく。しかし、このアクティビティ図は基本的に正常系のシナリオで不完全なので、今後は、実験の結果を元に異常系の流れも含め、完全にユーザの行動を網羅する必要がある。

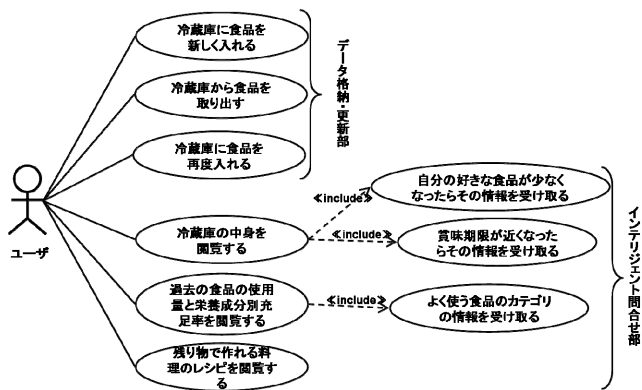


図 3: ユースケース図

データ格納・更新系のユーザのアクティビティ図の一部を図 4 に示す。

食品名の入力負担を軽減するため次のように工夫した。図 5 のように携帯電話のようなボタンを用意し、ユーザが食品名の頭文字を入力すると、いままで登録した食品の中から、頭文字が一致している食品名を候補としてボタンで表示し、該当するボタンを押すと、その食品名が入力される。

図 4 の「冷蔵庫に食品を再度入れる」は「冷蔵庫から食品を取り出す」の「出庫」を「入庫」、「リーダの前を通しながら冷蔵庫から取り出す」を「リーダの前を通しながら冷蔵庫に入れる」に変えたものである。

インテリジェント問合せ部のユーザのアクティビティ図の一部を図 6 に示す。遠隔ユーザの場合、室内ユーザのタッチパネルの「中身を確認ボタンを押す」が「携帯電話用 URL にアクセス」に変えたものである。

2.3. データベース設計

2.3.1. i-冷蔵庫データベースの概念モデル

E-R ダイアグラムを用いて i-冷蔵庫データベースを構築するための概念モデルを図 7 に示す。

2.3.2. リレーショナルテーブル群

E-R ダイアグラムに従うと、「i 食品」、「中身」、「貼

り付け」、「袋詰め食品」、「タグ」、「食品分類」、「カテゴリ」、「入出庫」テーブルが作成される。補足として、袋詰め食品とカテゴリが M 対 N なのは、一食品につき複数のカテゴリ、かつ、一カテゴリにつき、複数の袋詰め食品が対応しているからである。例えば、ユーザは、キャベツを入庫するとき、カテゴリを「野菜」、「ロールキャベツ」として登録できるといったようにユーザの好きなように分類することができる。

E-R 図を元にリレーショナルテーブルを作成すると、本システムのデータベースを構成するリレーショナルテーブル群は図 8 のように「i 食品」、「袋詰め食品」、「タグ」、「食品分類」、「カテゴリ」、「入出庫」、「参照」、「食品の栄養成分」の 8 つのテーブルになる。なお、i 食品テーブルの「状態フラグ」は 1 なら使用中を表し、0 なら使用してない状態を表す。また、入出庫テーブルの「処理」は 1 は入庫、2 は出庫、3 は使い切りを表す。

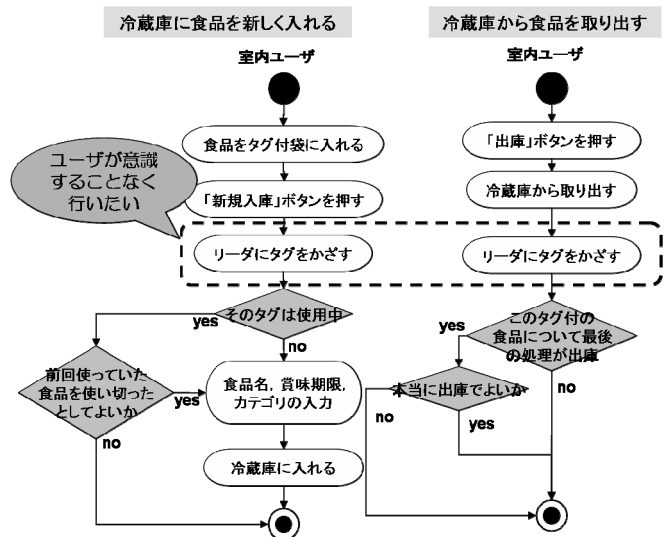


図 4: データ格納・更新系のユーザのアクティビティ図



図 5: 食品名の入力

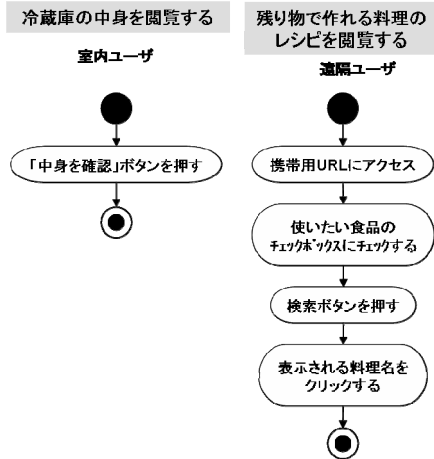


図 6: インテリジェント問合せ部のユーザアクティビティ図

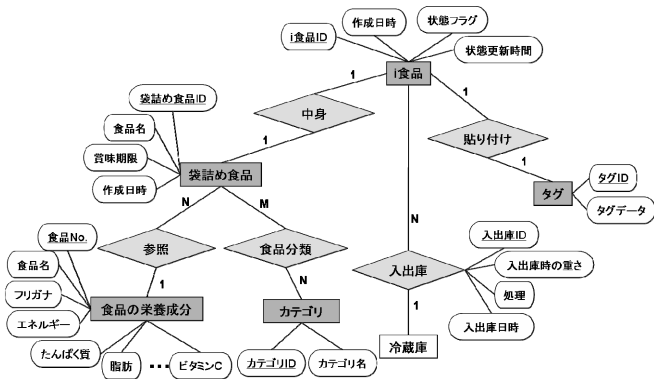


図 7: E-R ダイアグラム

図 9 と図 10 に実際のテーブル定義を示す。

「i 食品」を「ifood」, 「袋詰め食品」を「food_bag」, 「タグ」を「tag」, 「食品分類」を「food_class」, 「カテゴリー」を「category」, 「入出庫」を「store」, 「食品分類」を「nutrition_ref」, 「食品の栄養成分」を「nutrition」として表している。

```
mysql> describe ifood;
```

Field	Type	Null	Key	Default	Extra
ifood_id	int(11)	NO	PRI	NULL	auto_increment
food_b_id	text	YES		NULL	
tag_id	int(11)	YES		NULL	
new_store_datetime	datetime	YES		NULL	
status_flg	smallint(1)	YES	MUL	NULL	
upd_datetime	datetime	YES		NULL	

```
mysql> describe food_bag;
```

Field	Type	Null	Key	Default	Extra
food_b_id	int(11)	NO	PRI	NULL	auto_increment
food_b_nm	varchar(50)	YES	MUL	NULL	
expire_date	datetime	YES	MUL	NULL	
upd_date	datetime	YES		NULL	

```
mysql> describe tag;
```

Field	Type	Null	Key	Default	Extra
tag_id	int(11)	NO	PRI	NULL	auto_increment
tag_data	char(32)	NO	MUL		

```
mysql> describe food_class;
```

Field	Type	Null	Key	Default	Extra
food_b_id	int(11)	NO	PRI		
cat_id	int(11)	NO	PRI		
upd_date	datetime	YES		NULL	

```
mysql> describe category;
```

Field	Type	Null	Key	Default	Extra
cat_id	int(11)	NO	PRI	NULL	auto_increment
cat_nm	varchar(50)	NO			
upd_date	datetime	YES		NULL	

```
mysql> describe store;
```

Field	Type	Null	Key	Default	Extra
store_id	int(11)	NO	PRI	NULL	auto_increment
ifood_id	int(11)	NO	MUL		
store_datetime	datetime	YES		NULL	
weight	float	YES		NULL	
operation	smallint(1)	NO	MUL	1	

図 9: 実際のテーブル定義 1

i食品				中身		貼り付け	
i食品ID	作成日時	状態フラグ	状態更新時間	i食品ID	袋詰め食品ID	i食品ID	タグID
1	2005/01/08 18:00:00	1	NULL	1	1	1	1
2	2005/01/08 18:10:00	1	NULL	2	2	2	2

JOIN

i食品				袋詰め食品		タグ	
i食品ID	袋詰め食品ID	タグID	作成日時	状態フラグ	状態更新時間	タグID	タグデータ
1	1	1	2005/01/08 18:00:00	1	NULL	1	0304590711150020B4009282222F
2	2	2	2005/01/08 18:10:00	1	NULL	2	03015907111500222A009282222F

食品分類				入出庫			
袋詰め食品ID	食品No.	フリガナ	エネルギー	たんぱく質	脂肪	...	ビタミン
1	6212	ニンジン	37	0.6	0.1	...	4
2	6182	トマト	19	0.7	0.1	...	15

食品分類				食品の栄養成分					
袋詰め食品ID	食品No.	食品名	フリガナ	エネルギー	たんぱく質	脂肪	...	ビタミン	...
1	6212	にんじん	ニンジン	37	0.6	0.1	...	4	...
2	6182	トマト	トマト	19	0.7	0.1	...	15	...

図 8: リレーショナルテーブル群

```
mysql> describe nutrition;
```

Field	Type	Null	Key	Default	Extra
food_no	int(5)	NO	PRI		
g_code	int(11)	NO			
furigana	varchar(40)	NO	MUL		
food_name	varchar(40)	YES		NULL	
Refuse	int(3)	YES		NULL	
Energykcal	int(4)	YES		NULL	
EnergykJ	int(5)	YES		NULL	
Water	double(10,1)	YES		NULL	
Protein	double(10,1)	YES		NULL	
Lipid	double(10,1)	YES		NULL	
Carbohydrate	double(10,1)	YES		NULL	
Ash	double(10,1)	YES		NULL	
Sodium	int(5)	YES		NULL	
Ptassium	int(4)	YES		NULL	
Calcium	int(4)	YES		NULL	
Magnesium	int(4)	YES		NULL	
Phosphorus	int(4)	YES		NULL	
Iron	double(10,1)	YES		NULL	
Zinc	double(10,1)	YES		NULL	
Copper	double(10,2)	YES		NULL	
Manganese	double(10,2)	YES		NULL	
Retinol	int(4)	YES		NULL	
CarotenesA	int(5)	YES		NULL	
CarotenesB	int(5)	YES		NULL	
Cryptoxanthin	int(5)	YES		NULL	
B-Carotene_equivalents	int(5)	YES		NULL	
Retinol_activity_equivalents	int(5)	YES		NULL	
VitaminD	int(5)	YES		NULL	
TocopherolsA	double(10,1)	YES		NULL	
TocopherolsB	double(10,1)	YES		NULL	
TocopherolsC	double(10,1)	YES		NULL	
TocopherolsD	double(10,1)	YES		NULL	
VitaminK	int(5)	YES		NULL	
Thiamin	double(10,2)	YES		NULL	
Riboflavin	double(10,2)	YES		NULL	
Niacin	double(10,1)	YES		NULL	
VitaminE6	double(10,2)	YES		NULL	
VitaminB12	double(10,1)	YES		NULL	
Folate	int(5)	YES		NULL	
Pantothenic_acid	double(10,2)	YES		NULL	
Ascorbic_acid	int(5)	YES		NULL	
Saturated	double(10,2)	YES		NULL	
Monounsaturated	double(10,2)	YES		NULL	
Polynsaturated	double(10,2)	YES		NULL	
Cholesterol	int(5)	YES		NULL	
Fibers_Soluble	double(10,1)	YES		NULL	
Fibers_Insoluble	double(10,1)	YES		NULL	
Fibers_Total	double(10,1)	YES		NULL	
Salt_equivalents	double(10,1)	YES		NULL	

```
mysql> describe nutrition_ref;
```

Field	Type	Null	Key	Default	Extra
food_b_id	int(11)	NO	PRI		
food_no	int(11)	NO			

図 10: 実際のテーブル定義 2

2.4. 現在の冷蔵庫の中身のデータ抽出方法

現在の冷蔵庫の中身を表示させると、どの食品が現在どれくらい冷蔵庫に残っているのかを知ることができる。現在の冷蔵庫の中身の画面をタッチパネルで表示したものを図 11 に、携帯電話に表示したものを図 12 に示す。タッチパネルの方では、右から、食品名、重さ、賞味期限、状態(冷蔵庫内にあれば、空欄、冷蔵庫の外にあれば使用中と表示)の項目がある。

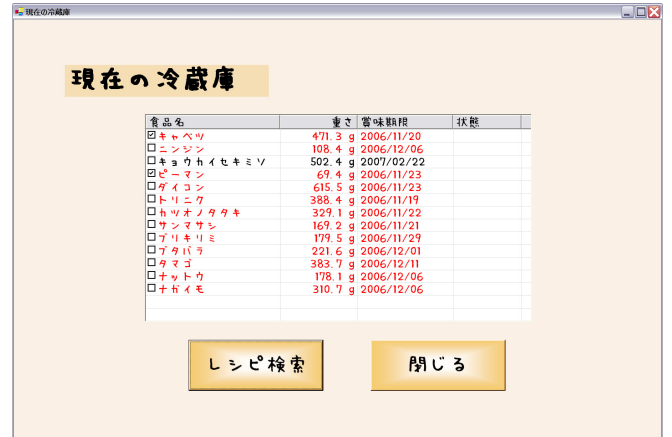


図 11: 現在の冷蔵庫の中身の画面(タッチパネル)

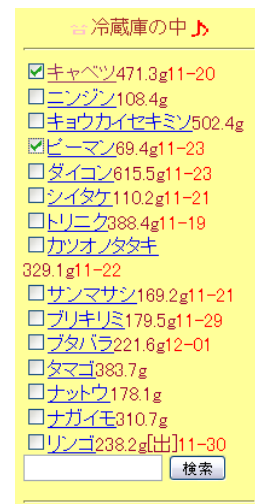


図 12: 現在の冷蔵庫の中身の画面(携帯電話)

現在の冷蔵庫の中身のデータを抽出する大まかな方法は、図 13 のような流れである。図 13 の①、②、③、⑤、⑥で発行される SQL 文の例を以下に示す。

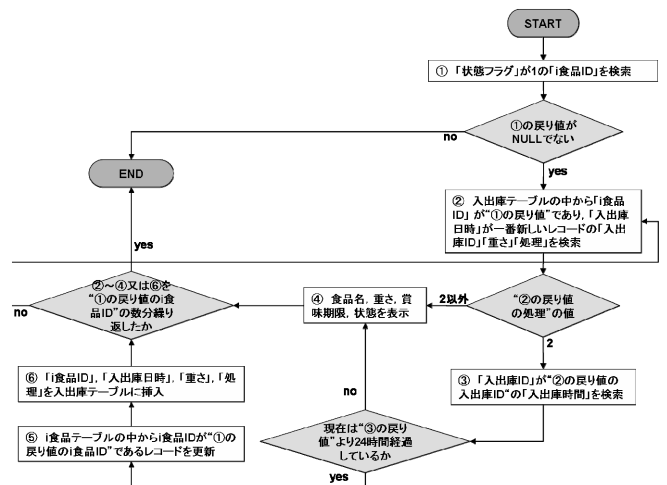


図 13: 現在の冷蔵庫の中身のデータ抽出方法

- ① SELECT a.ifood_id,b.food_b_nm,b.expire_date
FROM ifood a
LEFT JOIN food_bag b
ON a.food_b_id=b.food_b_id
WHERE a.status_flg='1'
- ② SELECT store_id,weight,operation
FROM store WHERE ifood_id='40'
ORDER BY store_datetime DESC LIMIT 1
- ③ SELECT store_datetime
FROM store WHERE store_id = '26'
- ⑤ UPDATE ifood SET status_flg = 0,
upd_datetime = now()
WHERE ifood_id = '54'
- ⑥ INSERT INTO store (ifood_id,store_datetime,
weight,operation) VALUES ('54',now(),'0','3')

2.5. 使用履歴のデータ抽出方法

過去の食品使用量と栄養成分別充足率は、ユーザがどんな食品をどれくらい使用したのかや、日々の食生活の栄養バランスを閲覧することができる。

図 14 に過去の食品使用量と栄養成分別充足率の画面を示す。図 14 上部に表示しているカレンダーの日付を押すと、左下の表にその日使用した食品の食品名、出庫日時、入庫日時、使用量、カテゴリが表示され、右下に、その日当たりの充足率が表示される。これにより、例えば、目標量の 100%未満だと、その日消費した食品が不十分だったとわかる。

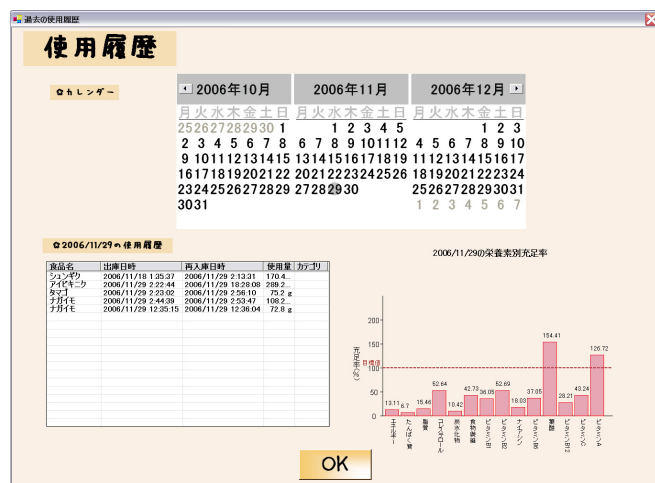


図 14: 使用履歴の画面(タッチパネル)

使用履歴のデータを抽出する大まかな方法は、図 15 のような流れである。

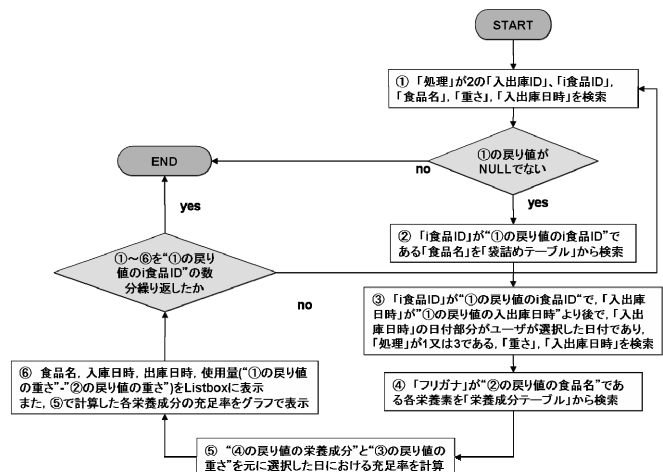


図 15: 現在の冷蔵庫の中身のデータ抽出方法

図 15 の①, ②で発行される SQL 文の例を以下に示す。

- ① SELECT store_id,ifood_id,store_datetime,weight
FROM store WHERE operation=2
- ② SELECT b.food_b_id, b.food_b_nm
FROM ifood a LEFT JOIN food_bag b
ON a.food_b_id = b.food_b_id
WHERE a.ifood_id = '48'
- ③ SELECT store_id,store_datetime,weight
FROM store WHERE ifood_id='48'
AND (operation=1 or operation=3)
AND store_datetime > '2006/11/14 23:00:40'
AND store_datetime = '2006/11/29 00:00:00'
ORDER BY store_datetime ASC LIMIT 1
- ④ SELECT Refuse, Energykcal, Protein, Lipid,
Cholesterol, Carbohydrate, Fibers_Total,
Thiamin, Riboflavin, Niacin, VitaminB6, Folate,
VitaminB12, Ascorbic_acid,
Retinol_activity_equivalents

From nutrition WHERE furigana LIKE '%ナットウ%'

2.6. 冷蔵庫内の食品で作れるレシピ情報取得方法

冷蔵庫の残り物で作れる料理のレシピを表示する方法を述べる。ただし、食品名のみによる検索なので、重さは考慮されない。

これは本システムと Yahoo!が持つ「Yahoo!グルメレシピ情報」[3]のページを連結することにより、レシピ情報を得ている。方法は次の通りである。

- ① 本システムの冷蔵庫の中身の情報を表示して、使いたい食品名の横のチェックボックスにチェックし、検索ボタンを押す。
- ② すると、食品名を検索キーワードとし、それを GET 引数として Yahoo! グルメのサイトに渡す。Yahoo! グルメサイトの URL は PC と携帯電話で異なる。

- PC の場合
URL: <http://recipe.gourmet.yahoo.co.jp/list.html>
引数: kw=検索キーワード
- 携帯電話用の場合
URL:
<http://recipes.mobile.yahoo.co.jp/mobileRecipe>
引数: ky=検索キーワード

3. 実験の目的と環境

3.1. 実験の目的と ISO 9241-11

本実験の目的は、現システムのユーザビリティを評価し、バグや改良の余地がある箇所の発見した場合は、修正してシステムパフォーマンスの向上を図るものである。

我々は、ユーザビリティの評価の尺度として ISO 9241-11 を用いることとした。

ISO 9241-11 は人間工学やユーザ・インタフェースの分野で用いられ、ユーザビリティの定義(以下参照)と、ユーザビリティをユーザの行動と満足度を尺度に規定または評価する場合に考慮しなければならない情報の認識方法を説明した国際規格である。

[ISO 9241-11 の定義]

- **ユーザビリティ (Usability):** 特定の利用状況において、特定のユーザによって、ある製品が、指定された目標を達成するために用いられる際の、有効さ、効率、ユーザの満足度の度合い。
- **有効さ (Effectiveness):** ユーザが指定された目標を達成する上での正確さ、完全性。
- **効率 (Efficiency):** ユーザが目標を達成する際に、正確さと完全性に費やした資源。
- **満足度 (Satisfaction):** 製品を使用する際の、不快感のなさ、及び肯定的な態度。
- **利用状況 (Context of use):** ユーザ、仕事、装置(ハードウェア、ソフトウェア及び資材)、並びに製品が使用される物理的及び社会的環境。

3.2. 実験環境

- アプリケーション用 PC:
 - OS: WindowsXP Home Edition
 - CPU: Pentium M 1.6GHz
 - RAM: 512MB
- Web サーバ:
 - Apache 1.3
 - OS: Red Hat Linux 3.2.2-5
 - CPU: Pentium III 695.424MHz
 - RAM: 375MB
- DBMS:
 - MySQL Ver. 5.0.15
 - OS: WindowsXP Professional

- CPU: Pentium® 4 CPU 1.70GHz
- RAM: 512MB
- RFID R/W: 1 台
 - 凸版印刷提供 TesStar TSW-2G45S-U2F
- RFID タグ: 20 枚
 - 凸版印刷提供 ラベル型 TesStar
- 電子天秤: 1 台
 - Type UX4200S
 - 4kg まで測定可能. 測定した重さを RS232C により PC に伝送可能.
- 携帯電話: 1 台
 - P901i
- タッチパネル: 1 台
 - SHARP 製 LL-151TR
- 携帯用 Web アプリケーション:
 - PHP Ver. 4.4.0 で開発
- タッチパネル用 GUI:
 - VisualBasic.NET 2003 で開発

プロトタイプングにあたっては、電子天秤を冷蔵庫に入れるのは、接続されているコード類がドアに挟まれてしまい困難だったので、冷蔵庫付近に天秤を設置した。入庫する時は、最初にボタンを押下し、リーダーに読ませた後、天秤の上に置いてから冷蔵庫に入れ、出庫する時は、食品を冷蔵庫から出して天秤の上に置いてボタンを押下した後、タグを読ませながら食品を取り上げた。

4. 実験結果

3.1.節で導入した ISO 9241-11 に則り行ったプロトタイプシステムの評価結果を示す。まず、利用状況を述べ、続いて、有効さ、効率、満足度の評価を述べ、最後に、プロトタイプシステムのユーザビリティの評価を論じる。

4.1. 利用状況

実装したプロトタイプシステムを以下の状況で活用した。

- 期間: 約 3 週間(2006 年 11 月 13 日～12 月 1 日)
- 場所: 自宅
- 被験者: 一人(開発者自身)

取得した食品に関するデータをまとめると、期間内に出し入れた食品の総数は 40 で、出し入れしたトータル回数は 127 であった。よって、1i 食品当たりの平均出し入れ回数は約 3 回という結果が得られた。実際の現在の冷蔵庫の中身をタッチパネルに表示したものを図 11、図 12 に、過去の食品使用量と栄養成分別充足率を表示したものを図 14 に示す。

なお、今回は開発者自身の単一ユーザで、ユーザビリ

ティの検証を行ったが、一般に製品においては、コンピュータリテラシのない人も活用することになるため、開発者以外の様々な人に被験者になってもらい、評価を行っていく必要があると考えている。

4.2. 有効さの評価

- 出し入れする際、正常な手順を踏まなければ、正確さは保証されない
- 特に再入庫や出庫時に天秤の上に乗せるのとリーダにかざすのを忘れる

■ 改善策

- 重さの増減をシステムが認識して、押したボタンと一貫していなければメッセージや音を鳴らして注意を促す
- RFID リーダと電子天秤の設置場所の工夫

4.3. 効率の評価

- タッチパネル操作によりユーザの入力負担を軽減
- 天秤を冷蔵庫の中に入れることができなかつたため、手間を感じた

■ 改善策

- 天秤を冷蔵庫の中に入れられるよう工夫
- 天秤と RFID リーダを合体させて、リーダにかざすことをユーザに意識させないようにする

4.4. 満足度の評価

- 携帯電話を使って現在の冷蔵庫の中身を表示し、レシピを検索できるのは非常に手軽と感じた
- 実際、移動中に冷蔵庫の中身を確認し、食事の献立を考えてから買い物するパターンが定着し、以下のような利点が得られた。
 - 今まで作ったことのない料理への挑戦欲が増した。
 - 買い物中に冷蔵庫の中身を確認できるので、買い物時間が短縮された。
 - 食品購入時の無駄がなくなり食費の節約に効果的。
 - 料理を作るモチベーションが自然に上がった。
 - 栄養バランスを前より考えるようになった。

■ 改善策

- いつも常備している食品がなくなったときのお知らせ機能の追加

4.5. ユーザビリティの評価

これまで挙げた改善策の施行後をパーフェクトの 5 として、自分自身で現システムのユーザビリティを評価すると、図 15 のようになる。

評価理由は、有効さは、天秤の上に乗せるのとリーダにかざす作業を忘れる場合があるので 3、効率は、天秤を冷蔵庫に入れていないので 3、満足度は、入力の手間をまだ感じることから 4 と評価した。

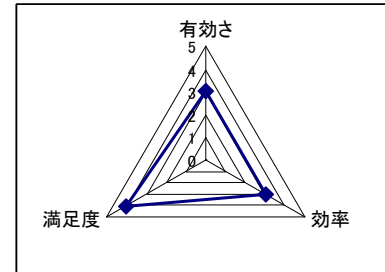


図 15: ユーザビリティ評価結果

5. まとめと今後の課題

考案したプロトタイプシステムの概要と開発者自身で行った実験結果について述べ、データベースと実世界との一貫性が保持されない場合の改善策など、システム評価について報告した。

携帯電話で現在の冷蔵庫の中身を表示するというのは非常に手軽で利用しやすかったので、実際に使ってみて本システムの有用性を感じた。しかし、入庫や出庫の際、天秤の上に乗せるのとリーダにかざす作業を怠ってしまう場合など、データベースと実世界との一貫性が保持されない場合もあった。

従って今後の課題として、ユーザの行動を網羅的にモデル化し、データベースとの誤差が最小になるよう改善策の施行、実験を繰り返し、その後、操作方法の分からない多くの人に被験者となってもらい評価してもらうことが挙げられる。同時に、よりユーザの負担を軽減させるため、RFID と電子天秤の特性を活かせる最適な設置場所を模索する必要がある。

また、今後の展望としては、データベースに格納されているユーザ個人の食生活事情を利用して、問題のある食生活を改善するよう提案できるシステム開発を目指すことも意義がある。また、冷蔵庫以外の他の格納庫内のモノの管理への適用も検討する価値がある。

文 献

- [1] 金野紋子, 増永良文: "インテリジェント冷蔵庫システムの構想," 日本データベース学会 Letters, Vol.4, No.2, pp.73-76, September 2005.
- [2] Eric J. Naiburg, and Robert A. Maksimchuk: "UML for Database Design," Addison Wesley, 300p., 2002.
- [3] " レシピ情報 - Yahoo! グルメ ", <http://recipe.gourmet.yahoo.co.jp/>