

異なるライフログを集約するための標準データモデルの考察

中村 匡秀[†] 下條 彰[†] 井垣 宏[†]

[†] 神戸大学大学院工学研究科 〒657-8531 神戸市灘区六甲台町1-1
E-mail: †{masa-n,igaki}@cs.kobe-u.ac.jp, ††shimojo@ws.cs.kobe-u.ac.jp

あらまし 様々な種類のライフログを効率よく連携・集約(マッシュアップ)するためには, ある程度標準化されたデータモデルやサービス API が必要となってくる. 本稿では, ライフログが備えるべきデータを 5W1H の観点から分析し, アプリケーション依存・非依存のデータ項目を抽出する. 抽出したデータ項目に基づき, ライフログのための標準的なデータモデル, および, サービス API を考察する.

キーワード ライフログ, マッシュアップ, 標準データモデル, サービス API

Considering Common Data Model to Mash-up Life Logs

Masahide NAKAMURA[†], Akira SHIMOJO[†], and Hiroshi IGAKI[†]

[†] Graduate School of Engineering, Kobe University
657-8531 Japan

E-mail: †{masa-n,igaki}@cs.kobe-u.ac.jp, ††shimojo@ws.cs.kobe-u.ac.jp

Abstract In order to support the data mash-up of different kinds of life logs, this paper discusses a common data model and generic APIs for life log applications. In this paper, we first examine the application-neutral or -specific data for life log in general, based on objective analysis from viewpoints of what, why, when, who, where and how. We then propose a common data model and generic APIs from the derived data.

Key words life log, mash up, common data model, service API

1. はじめに

人間の日常生活における行動をデジタルデータとして記録するライフログが注目を集めている. インターネット上には多種多様なアプリケーション・サービスが登場し, 様々な種類のライフログを Web 上で手軽に記録・共有できるようになってきている. 具体例として, ユーザの毎日の日記を記録するブログに始まり, つぶやきを記録する “twitter” [1], 写真を記録する “Flickr” [2], 三度の食事を記録する “FoodLog” [3], 体や運動のデータを記録する “からだログ” [4], 携帯電話で写真やバーコード, GPS 情報等を記録できる “ケータイ de ライフログ” [5], 睡眠時間を記録する “ねむログ” [6] 等が存在する.

こうしたライフログ・アプリケーションにおいては, ライフログの「記録」と「利活用」は同一アプリケーション内に閉じた形で行われている. しかしながら, 様々なアプリケーションでばらばらに記録されたライフログを, 集約・連携(マッシュアップと呼ぶ)することで, より付加価値の高い情報・サービスへと発展させることが期待できる.

ライフログの高度かつ柔軟なマッシュアップを支援するため, 本研究では, ライフログのための標準的なデータモデルおよび

サービス API を考察する. 既存のライフログ・アプリケーションの中には, マッシュアップ用の API やログパーツを用意しているものも存在し, 外部アプリケーションからログデータを操作することができる. しかしながら, これら API の仕様はアプリケーション毎にまちまちであり, 連携するアプリケーションの組み合わせごとに異なるプログラムロジックが必要となる. マッシュアップの効率化, 生産性をより向上するには, ある程度標準的なデータモデルおよびサービス API が望まれる.

本稿では, ライフログに必要なデータ項目を 5W1H の観点から分析し, アプリケーションに独立なものと依存するものに分類, アプリケーションに中立な論理データモデルを構築する. また, このデータモデルを操作するための汎用的なサービス API を考察する. また, 提案したデータモデルおよび API を, 既存のライフログ・アプリケーションに適用し, その有効性を検証する.

2. 準備

2.1 ライフログ

ライフログとは「人間の行い (*life*) をデジタルデータとして記録 (*log*) に残すこと」[7] である. 人間の日常行動を記録とし

て残しておき、生活の振り返りや改善、自分発見や気づき、さらには意思決定、行動支援に役立てようとする試みである。ただし一言で life といっても様々なものが考えられる。日記、心のつぶやき、見た風景、行った場所、移動した経路、生体情報（脈拍、歩数など）など、パリエーションは多岐にわたる。

しかし人間の全ての行動を特別な記録環境・装置でつぶさに記録するアプローチは、ユーザの負担が大きく、現実的ではない。ライフログが広く受け入れられるためには「必要な人が必要なものだけを記録する」ことが重要とされている [8]。

このことは、現在インターネット上で人気を集めている様々なライフログ・アプリケーションに見てとれる。これらのアプリケーションでは、それぞれ記録する内容が専門化されており、データ入力方式も含めてユーザの負担がなるべく低くなるよう工夫されている。ユーザは自分の用途に応じて好きなアプリケーションを選び、好きな時にライフログを記録する。

現行のライフログ・アプリケーションは、ライフログの記録と利用が一つのアプリケーション内に閉じている。結果的に、様々な life の log がアプリケーションごとにばらばらに管理され、ネットワーク上に分散して蓄積されていくことになる。

2.2 ライフログのマッシュアップ

複数のアプリケーションで取得されたばらばらのライフログを、ネットワーク越しに集約・連携することで、単独で利用する以上の価値が生まれる可能性がある。実際に、ブログ記事や写真、メモなど複数のデータソースを同じタイムラインに並べて「アルバム」を作る“Life-X” [9] や、「自分史」を作る“AllOfMe” [10] が存在する。他のアイデアとして例えば、“FoodLog”と“からだログ”を連携することで、食生活の実態と結果を細かく分析できるだろう。また、“twitter”と“ねむログ”を連携することで、考え事と睡眠時間の相関が取れるかもしれない。

複数のライフログの集約・連携による付加価値創造を、本稿ではライフログのマッシュアップと呼ぶことにする。こうしたマッシュアップを見越して、ライフログ・アプリケーションの中には、外部アプリケーションから自身のライフログへのアクセス手段を、サービス API や Web 部品（ブログパーツ等）という形で公開しているものも存在する。

2.3 現状の課題

図 1 に現在のライフログのマッシュアップ形態を示す。図中、Mash-up App. はマッシュアップ・アプリケーションを、線は各ライフログ・アプリケーションへのアクセスをあらわす。マッシュアップ・アプリケーションから、各ライフログ・アプリケーションへのアクセスは、2.2 で述べた API やブログパーツを利用して実現される^(注1)。

現在、こうした API や API によって得られるライフログのデータ構造には統一的な標準が無く、アプリケーションごとにまちまちである。つまり、ライフログへのアクセス手段がアプリケーションに強く依存している。したがって、マッシュアップ

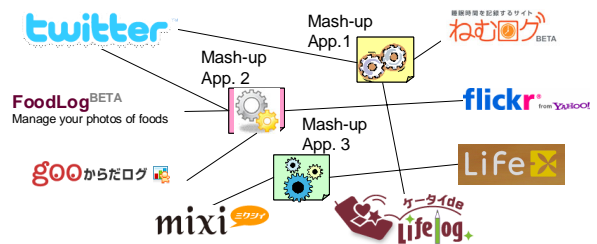


図 1 従来のライフログ・アプリケーションの集約・連携

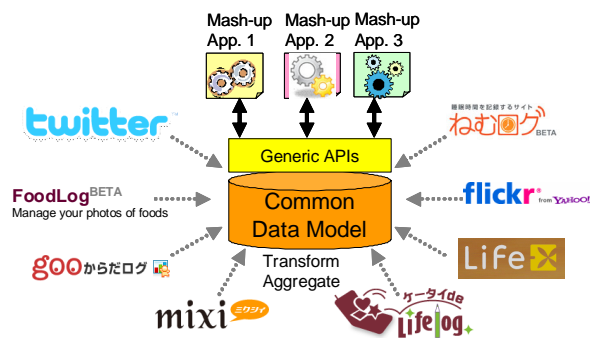


図 2 提案するライフログ・アプリケーションの集約・連携

の開発においては、連携するライフログ・アプリケーションの組み合わせごとに異なるプログラムロジックが必要となる。例えば、似たようなライフログ検索を行いたくても、アプリケーションが違えば全く異なる手段で呼び出さなければならない。このことは、マッシュアップの開発効率（生産性）を下げる。また、API 仕様の改訂に伴ってプログラムの見直しが必要となるため、再利用性、信頼性を下げる要因にもつながる。

3. ライフログのための標準データモデル

3.1 研究の目的とスコープ

本研究の目的は、ライフログのための標準データモデルおよび API を提案することである。図 2 に提案手法を用いたライフログのマッシュアップ形態を示す。各ライフログ・アプリケーションのデータは、何らかの形で、アプリケーションに強く依存しない標準的なデータモデル (Common Data Model) に変換される。この標準データモデルの上では汎用的な API (Generic APIs) が定義され、マッシュアップ・アプリケーションはこれら汎用 API を用いて開発される。ライフログへのアクセス手段がアプリケーションに強依存しなくなるため、2.3 で述べた課題が解決できる。

ここで研究のスコープを明確にするため次の目標を設定する。
目標 G1: 標準データモデルは、ライフログの論理的なデータおよびデータ構造をモデル化できること。物理的なデータ型の決定は今回対象外とする。

目標 G2: 標準データモデルは、ライフログの一次データ（生データ）をモデル化できること。二次データ（集計データ、加工データなど）は今回対象外とする。

目標 G3: API は、ライフログの汎用的な検索、取得が可能であること。アプリケーション固有のデータ操作（追加、更新）は今回対象外とする。

(注1): API が提供されないアプリケーションに関しては、出力画面の HTML を解析してデータ抽出を行うアプローチ等が考えられる。

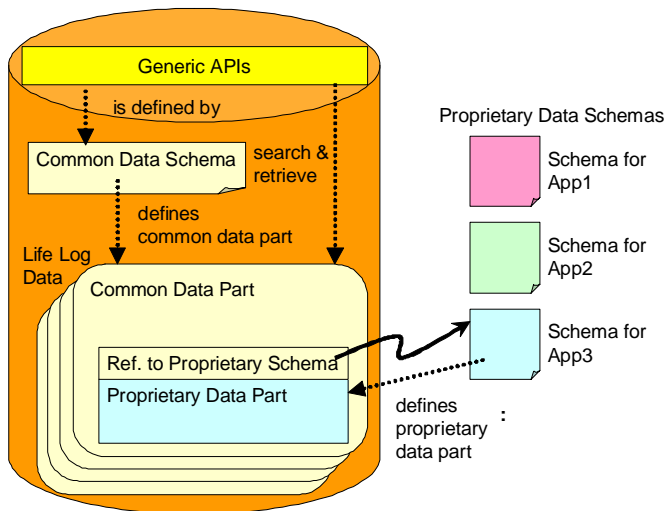


図3 標準データモデルのアーキテクチャ

3.2 アプローチ

標準的なデータモデルを構築するために、ライフログを構成するデータ項目を5W1H (What, Why, Who, When, Where, How) の観点から分析・抽出した後、アプリケーションに強く依存するものと非依存なものに分類する。

アプリケーション非依存なデータ項目は、ライフログの標準データモデルのスキーマ (標準スキーマ) に組み入れる。汎用APIは、標準スキーマを利用してアプリケーションに依存しないデータ操作を規定する。一方、アプリケーション固有のデータ項目に関しては、単なる構造付きデータとし、標準データモデル内では意味を定義・解釈しない。そのかわり、アプリケーションごとにスキーマを別定義し (固有スキーマ)、そこへの参照を保持する。

図3に提案する標準データモデルのアーキテクチャを示す。図中左側の円柱が、標準データモデルに従ったライフログのデータを保持するリポジトリを表す。円柱内、角の丸い四角が標準データモデルに従った個々のライフログデータを表している。各データは、アプリケーションに非依存なデータ (Common Data Part) と依存するデータ (Proprietary Data Part) に別れている。非依存データは標準スキーマ (Common Data Schema) を用いて定義・解釈される。また、依存データに関しては、それを定義・解釈するための固有スキーマ (Proprietary Data Schema) へのリンク (この場合、Schema for App3) が付与されている。

構造上すべてのデータ項目を固有スキーマに頼ることも可能である。しかしこの場合、各アプリケーションが提供する固有のAPIを使う従来方式と等価になり、標準データモデルの意味がなくなる。良いデータモデルおよびAPIを構築するには、アプリ非依存のデータ項目をどれだけ洗い出せるかが鍵となる。

3.3 5W1Hに基づくデータ項目の考察

ライフログを構成するデータ項目を以下の観点から考察する。

- **WHAT:** 何の記録なのか？
- **WHY:** なぜ記録されたのか？
- **WHEN:** いつの記録なのか？
- **WHO:** 誰の記録なのか？

- **WHERE:** どの記録なのか？
- **HOW:** どのように記録されたのか？

わかりやすさのため、以降では各観点の議論において登場するデータ項目をタグ (例: <date>) の形式で表現する。

3.3.1 WHATの観点からのデータ項目

2.1で述べたように、ライフログにおいて何を記録するかは非常にバリエーションあり、用途に応じた様々なアプリケーションが存在する。例えばtwitterなら「つぶやき<text>」であるし、flickrなら「写真<photo>」となる。したがって、WHATに相当するデータ項目は、それぞれのアプリケーションに特化 (依存) したデータとなる。標準データモデルにおいてはこれらを特に区別・解釈せず、<content>というタグのついた単なる構造付きデータとして扱う。また3.2で述べたとおり、<content>の内容解釈はアプリケーションごとの固有スキーマを用いる。したがって標準データモデルには固有スキーマへの参照<ref_schema>を組み入れる。

3.3.2 WHYの観点からのデータ項目

そのライフログがなぜ (WHY) 記録されたかは、明示的に書かれない限り知りえない。どのような目的<purpose>で取られたのか、どのような理由<reason>があったのか、何を契機として<trigger>記録されたのか等、WHYを説明するいくつかのデータ項目が考えられる。しかしながら、アプリケーションによってはログの目的を説明できることもあるし (例: 体重のデータを取るのには痩せるため)、理由なく取られた記録もある。したがって、WHYに関するデータ項目はアプリケーション依存、または、大規模なデータマイニングによって得られるものと考ええる。よって、個々のログデータが従うべき標準スキーマからは除外することにする。

3.3.3 WHENの観点からのデータ項目

そのライフログがいつ (WHEN) のものなのかはアプリケーションに依存しない。また、ライフログに限らず、すべてのログ (例えば計算機のログ) に付与されるべきデータ項目である。具体的には、データを記録した日付<date>や時刻<time>が代表的なデータ項目である。ただし日付や時刻を、行動 (life) が発生した時<occured_at>に紐付けるか、記録 (log) した時<recorded_at>と考えるか等、複数の解釈が可能である。アプリケーションによって複数の日付を持つものや、一つの解釈に統一するものなどまちまちである。標準データモデルではアプリケーションごとにどれか一つの解釈に決めるものとし、<date>と<time>に対応付けることとする。

3.3.4 WHOの観点からのデータ項目

ライフログが人間の行動の記録であることから、全てのライフログ (一次データ) には、誰が (WHO) その行動を起こしたか、主体となるユーザ<user>が紐付けられると考えてよい。したがって<user>を標準スキーマに組み入れる。さらに、行動が集団行動であった場合、誰と行動したか (WITH WHOM) を表すデータ<party>や、行動の対象 (FOR WHOM) を表すデータ<object>も、アプリケーションにそれほど強く依存しないデータ項目である。これらは複数ユーザの関連づける有用なデータであり、標準スキーマの候補として十分考えられる。

3.3.5 WHERE の観点からのデータ項目

そのライフログがどこ (WHERE) の記録なのかは、アプリケーションに依存しない形で存在するはずである。携帯電話の GPS 機能で記録される現在位置の座標であったり、写真であれば撮影した場所、体重であれば計った場所 (例: 家, 温泉) など、データに紐づく場所情報<location>は、データの種類を問わず、明示的あるいは暗黙的に説明できることが多い。したがって<location>を標準スキーマに組み入れる。WHEN のときと同様、場所を行動 (life) が発生した場所<occured_at>に紐付けるか、記録 (log) した時<recorded_at>とするか、二通りの解釈が可能であるが、標準データモデルでは両者を厳密に区別しない。

3.3.6 HOW の観点からのデータ項目

そのライフログをどのように (HOW) 記録したのかは、理想的にはアプリケーションに依存すべきではない。例えば、twitter を使おうが LifeSpaceTime [11] を使おうが、mixi の「今なにしている?機能」を使おうが、つぶやきやつぶやきである。また、同じ twitter でも PC からつぶやくか、携帯からつぶやくか、API でプログラムからつぶやくかは、つぶやきの内容とは独立のものである。すなわち記録手段の相違によって、ログの内容や性質が変わるものではない。ゆえに、手段に関するデータ項目は本来アプリケーションに依存するものではない。ただし、現実にはアプリケーションごとに提供される記録手段が異なるため、依存関係が生まれている。便宜上、提案する標準データモデルではどのアプリケーションで記録されたか<application>、および、どのデバイスで記録されたか<device>の 2 つのデータ項目を標準スキーマに組み入れることにする。

3.4 標準データモデル

3.3 の議論に基づき、アプリケーションに依存しないデータ項目を並べて、標準データモデルのスキーマとする (表 1 参照)。アプリケーションに依存する WHAT に関するデータ項目は、形式や内部構造を問わない内容<content>として抽象化を行っている。また、その内容の定義を表す固有スキーマへの参照リンク<ref_schema>とする。WHY に関するデータ項目はアプリケーション依存のため、必要があれば<content>の中に入れ、固有スキーマによって意味づけを行う。

既存のライフログ・アプリケーションに提案データモデルを適用するには、何らかのデータ変換を行う必要がある。

図 4 に、Flickr および twitter のデータに標準データモデルを適用する例を示す。図 4(a) は、Flickr におけるライフログの基本単位となる写真オブジェクト<photo>を、標準スキーマに即したデータ<lifelog>に変換した例である。<photo>オブジェクトは、写真のオーナーやタイトル、撮影日時、撮影場所、撮影したカメラ情報、コメント、写真画像の URL などから構成され、これらの情報は Flickr が提供する API(flickr.photos.getInfo, flickr.photos.getExif) から取得できる。この変換例では、撮影日時を<date>と<time>に、ユーザアカウント名を<user>に、撮影場所を<location>に、撮影したカメラのモデル番号を<device>にそれぞれマッピングしている。<content>には、オリジナルの<photo>データをそのままコピーし、<ref_schema>には Flickr

表 1 ライフログ標準データモデルのスキーマ

| 観点 | データ項目 | 説明 |
|-------|-------------------------------|---------------------------|
| WHEN | <date> <time> | 日付 時刻 |
| WHO | <user> <party> <object> | ユーザ 共に行動したユーザ 対象ユーザ |
| WHERE | <location> | 場所 |
| HOW | <application> <device> | アプリケーション ログ記録に用いたデバイス |
| WHAT | <content> <ref_schema> | ログの内容 外部スキーマへの参照 |
| WHY | ----- | 必要なら<content>内へ |

の API ドキュメントへの参照を指定している。オリジナルの形式をそのまま残しておくことで、Flickr データを操作するライブラリやサービスとの互換性を保っている。

図 4(b) は、twitter の API である user.timeline で取得したつぶやきデータ<status>を、標準データ<lifelog>に変換した例である。twitter では、特定の人に向けたつぶやきを@ユーザ名で指定できるため、これを対象ユーザ<object>にマッピングしている。また携帯版 twittwer の機能でつぶやきに場所を付加する「イマココ!機能」があるので、それを<location>にマッピングしている。また、つぶやきを発したインターフェース (Web, mobile, API など) を<device>に割り当てている。Flickr の例と同様、<content>には<status>の全データを含めている。

3.5 汎用 API

標準データモデルに変換されたライフログデータにアクセスするための以下の API を提案する。なお、目標 G3 で述べたとおり、データの検索、取得のみを対象としている。

インターフェース

```
getLifeLog(date, time, user, party, object,
            location, application, device);
```

引数:

- date: 日付に関するクエリ
- time: 時間に関するクエリ
- user: ユーザに関するクエリ
- party: 共に行動したユーザに関するクエリ
- object: 対象ユーザに関するクエリ
- location: 場所に関するクエリ
- application: アプリケーションに関するクエリ
- device: デバイスに関するクエリ

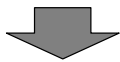
クエリ:

定数 (" "で囲んだ文字列), 論理演算子 (&&(論理積), ||(論理和), !(否定)), 範囲演算子 ([from .. to]), ワイルドカード* から構成される検索式。

戻り値: クエリに合致するライフログデータのリスト (配列)
使用例 1: "nakamura.masahide" が 2009 年 10 月に twitter と FoodLog で記録したライフログデータを全て取得する。

```
getLifeLog(["2009-10-01".."2009-10-30"],*, "nakamura.masahide", *, *, "twitter" || "FoodLog", *)
```

```
<photo id="2733" secret="123456"...>
  <owner username="masahide.nakamura"
    realname="Masahide Nakamura" ../>
  <title>Landscape of Kobe</title>
  :
  <dates posted="1100897479"
    taken="2009-10-01 12:34:56"../>
    <location latitude="-17.685895"
      longitude="-63.36914" ../>
  <exif tag="Model"../>FinePixel A123../</exif>
  <comments>...
  <urls><url>http://farm3.static.flickr.com/
    2777/2733_123456.jpg</url>...
</photo>
```



```
<lifelog id=10000001>
  <date>2009-10-01</date>
  <time>12:34:56</time>
  <user>masahide.nakamura</user>
  <party/>
  <object/>
  <location>
    latitude="-17.685895" longitude="-63.36914"
  </location>
  <application>Flickr</application>
  <device>FinePixel A123</device>
  <ref_schema> http://www.flickr.com/services¥
    /api/flickr.photos.getInfo.html</ref_schema>
  <content> <photo id="2733"../><url>http://farm
    3.static.flickr.com/27... </photo></content>
</lifelog>
```

(a) Flickr photo object

```
<status>
  <created_at>Fri Oct 16
    10:52:15 +0000 2009</created_at>
  <id>4913081442</id>
  <text>@akira.shimojo LOIS研究会原稿執筆中です。
  イマココ! 兵庫県神戸市灘区六甲台町</text>
  <source>web</source>
  :
  <user>
  <id>49944356</id>
  <screen_name>masahide.nakamura</screen_name>
  :
  </user>
</status>
```



```
<lifelog id=10000002>
  <date>2009-10-16</date>
  <time>10:52:15</time>
  <user>masahide.nakamura</user>
  <party/>
  <object>akira.shimojo</object>
  <location>
    兵庫県神戸市灘区六甲台町
  </location>
  <application>twitter</application>
  <device>web</device>
  <ref_schema>http://apiwiki.twitter.com/¥
    Return-Values</ref_schema>
  <content> <status><created_at>...<text>
    ...LOIS研究会原稿執筆中です。 ...</status></content>
</lifelog>
```

(b) twitter status data

図 4 標準データモデルへの変換例

使用例 2: 2009 年 10 月 21 日 12 時台に神戸で撮影された Flickr の写真データを全て取得する。

```
getLifeLog("2009-10-21", ["12:00:00".."12:59:59"], *,
  *, *, "*Kobe*", "Flickr", *)
```

使用例 3: "nakamura.masahide" が "shimojo.akira" と共に行った全ての行動のログを取得する。

```
getLifeLog(*, *, "nakamura.masahide", "shimojo.akira",
  *, *, *, *)
```

このように上記 API を用いると、複数のライフログ・アプリケーションにまたがるデータを、アプリケーションに依存しない方法で検索・取得できることがわかる。このことは、マッシュアップ開発の生産性を大きく向上と共に、ライフログの手軽な利活用促進に貢献できる。

なお、ライフログデータの中身 (WHAT) に依存する検索や内容更新は現状では考慮していないことに注意されたい (目標 G3 参照)。より高度な検索や、データの部分抽出・加工を可能とする複雑な API については、ライフログのコンテンツ処理への要求を考えて今後の課題としたい。

3.6 実装・運用の指針

ライフログ・アプリケーション固有のデータ形式から標準データモデルへの変換には、2 通りの実装方式が考えられる。一つめはオフライン方式である。あらかじめスケジュールされたタイミングで、バッチ処理を用いてアプリケーションからデータを一括取得・変換し、変換後のデータを蓄積しておく方式であ

る。汎用 API からのリクエストがくると、蓄積済みのデータを検索・提供する。全てのデータを変換して保持しておく必要があることと、最新データへのアクセスが制限されることが難点ではあるが、高速で軽量なデータ検索・取得が可能である。

もう一つはオンライン方式である。汎用 API からのリクエストがくるたびに、目的のライフログ・アプリケーションにアクセスし、データを取得・変換して返す。必要なデータのみを取得・変換し、最新データにリアルタイムにアクセスできる反面、データの検索・取得のオーバーヘッドがかさむ。どちらの実装方式を採用するかは、支援するマッシュアップの目的や用途に合わせて選択するべきである。

また実運用上の問題として、標準データモデルや汎用 API を誰が構築・提供するのにかつての議論がある。ひとつは、各ライフログ・アプリケーションのサービス提供者 (ライフログ事業者と呼ぶ) が、自らのデータを標準データモデルに変換し提供するやり方である。別の運用方法として、ライフログ事業者以外の第 3 の事業者 (サードパーティ事業者) が、ライフログ・マッシュアップ支援サービスとして変換プログラムを開発、標準データモデル、汎用 API を提供するやり方もある。具体的なビジネスモデルについては今後検討していきたい。

4. 評価

提案した標準データモデルがライフログの標準データ形式としてどの程度一般性を持つかを、既存のライフログ・アプリ

表 2 評価結果

| | | Twitter [1] | Flicker*[2] | FoodLog[3] | gooカタログ [4] | ケータイ de ライフログ* [5] | ネムログ [6] | Life-X[9] | LifeSpaceTime[11] | LOGPI[12] | AcTrec[13] |
|-------|---------------|-------------|-------------|------------|-------------|--------------------|----------|-----------|-------------------|-----------|------------|
| WHAT | <content> | つぶやき | 写真 | 食事(写真) | 健康データ | 多種(携帯) | 睡眠記録 | ログの集約 | 多種 | つぶやき | 行動履歴 |
| WHEN | <date> | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| | <time> | ○ | ○ | ○ | - | ○ | - | ○ | ○ | ○ | ○ |
| WHO | <user> | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| | <party> | #グループ名 | - | - | - | life:with | - | - | - | - | - |
| | <object> | @ユーザ名 | - | - | - | - | - | - | To宛ログ | 返信ログ | - |
| WHERE | <location> | イマココ!(モバイル) | ○ | - | - | ○ | - | ○ | 場所ログ | - | ○ |
| HOW | <application> | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| | <device> | <source>タグ | taken with | - | - | prf:Model | - | - | - | - | - |

ケーションへの適用可能性を通して評価する。具体的には、表 1 のデータ項目が、既存アプリケーションのオリジナルデータから抽出(導出)可能かどうかを分析する。

評価においては、前に述べた 8 種類のアプリケーションに加え、LOGPI (ログピ) [12] AcTrec [13] を追加した合計 10 種類のライフログ・アプリケーションを分析した。表 2 に結果を示す。表中 ○ はそのアプリケーションが標準で対応しているデータ項目であること、- はそのアプリケーションでは対応していないデータ項目であることを示す。また、標準的ではないが付加的な機能で追加可能なデータや、対応するデータ項目へ意味的に読み替え可能なデータは、テキストで補足説明をしている。

結果を見ると、<date>や<user>のようにどのアプリケーションでも対応しているデータ項目もあれば、<party>や<device>のようにまだ一部でしか対応していない項目もある。しかしながら、全てのデータ項目において、少なくとも 2 つ以上の既存アプリケーションが対応している。このことは、提案モデルが特定のアプリケーションに強く依存せずに中立であることを示す一つの結果であると考えている。

また、既存のライフログ・アプリケーションを分析するうち、ライフログの開示/非開示といったプライバシーやセキュリティに関するデータ項目を、個々のログデータに付与しているものが見受けられた。プライバシーやセキュリティは、ライフログのデータそのものの特徴や性質というより、ログをどう提供・利用するかというシステムの非機能要求である。そのため、本稿の 5W1H の分析からは抽出されなかった。現在のところ、こうした非機能要求に関するデータ項目は、ライフログそのものではなく、ライフログを管理するシステムやユーザ情報に付与される情報ではないかと考えている。標準データモデルに組み入れるべきかどうかは、今後の検討課題としたい。

5. おわりに

本稿では、異なるライフログ・アプリケーションの高度な集約・連携を支援するために、ライフログのための標準データモデルと、そのデータを汎用的に検索・取得するための API を提案した。ライフログが備えるべき論理的なデータ項目を 5W1H の観点から洗い出し、アプリケーション非依存な標準スキーマと、固有スキーマによって構成されるモデルを提案した。また、

提案モデルの既存ライフログ・アプリケーションへの適用可能性を評価し、モデルの一般性を確認した。

実用化に向けた課題としては、まずは物理データ設計を行うことである。ユーザ名<user>や位置<location>といった様々な形式で表現可能なデータに対し、どのような型で統一化すべきかを考える必要がある。これには、OpenID [14] や POIX_EX [15]、OpenSocial [16] 等の規格も参考にすべきである。また、実際のマッシュアップ・アプリケーションの開発を通して、提案モデルと汎用 API による効果を定量的に評価し、改善していく必要がある。また、標準データモデルを用いた実運用、ビジネスモデルについても考えていきたい。

謝 辞

この研究の一部は、科学技術研究費(若手研究 B 20700027, 21700077)の助成を受けて行われている。

文 献

- [1] Twitter, Inc., “twitter”. <http://twitter.com/>.
- [2] Yahoo, “Flickr”. <http://www.flickr.com/>.
- [3] 北村圭吾, 山崎俊彦, 相澤清晴, “食事ログの取得と処理: 画像処理による食事記録”. 映像情報メディア学会誌, vol.63, no.3, pp.376-379, March 2009.
- [4] NTT Resonant Inc., “からだログ”. <http://karada.goo.ne.jp/>.
- [5] KDDI R&D, “ケータイ de lifelog”. <https://lifelog.kddilabs.jp/lifelog/top.do>
- [6] BOstudio, Inc, “ねむログ”. <http://www.nemulog.jp/>.
- [7] IT 用語辞典バイナリ, “ライフログとは”. <http://www.sophia-it.com/content/ライフログ>
- [8] 相澤清晴, “ライフログの実践的活用: 食事ログからの展望”. 情報処理学会誌, vol.50, no.7, pp.592-597, July 2009.
- [9] Sony Marketing (Japan) Inc., “Life-x”. <http://life-x.jp/>.
- [10] AllofMe Ltd., “Allofme”. <http://www.allofme.com/>.
- [11] LifeSpaceTime, “LifeSpaceTime”. <http://www.lifespacetime.com/>.
- [12] paperboy&co., “Logpi(ログピ)”. <http://logpi.jp/>.
- [13] 山根隼人, 長尾 確, “AcTrec: 行動履歴を用いた個人行動支援”. 情報処理学会第 66 回全国大会, vol.66, no.3, pp.3.115-3.116, 2004.
- [14] OpenID Foundation, “Openid authentication 2.0 - draft 11”. <http://openid.net/specs/openid-authentication-2-0-11.html>.
- [15] ITS 情報通信システム推進会議, “位置情報表現形式ガイドライン poix_ex”. http://www.itsforum.gr.jp/Public/J7Database/p22/ITSFORUMRC-001_V20.pdf%.
- [16] Google Code, “OpenSocial”. <http://code.google.com/intl/apis/apis/opensocial/>