

クラウド・テクノロジー研究部会 Jenaを使った三つ組みデータの 格納と検索

2012/08/30

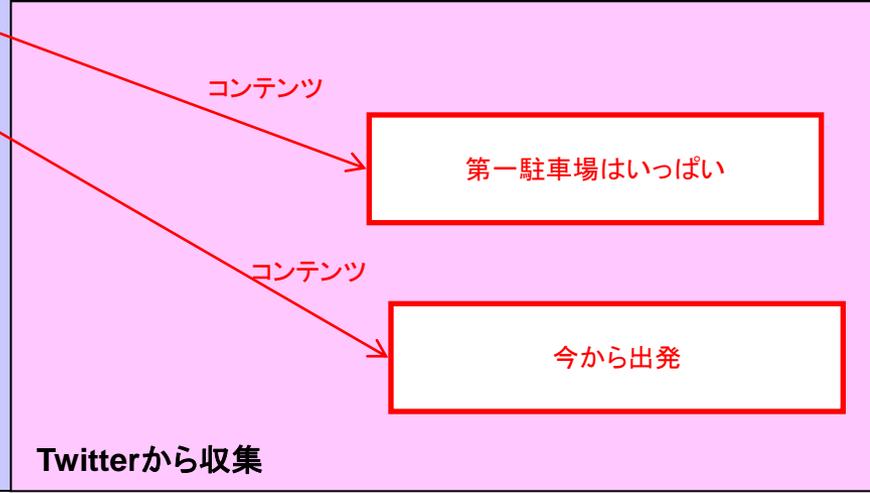
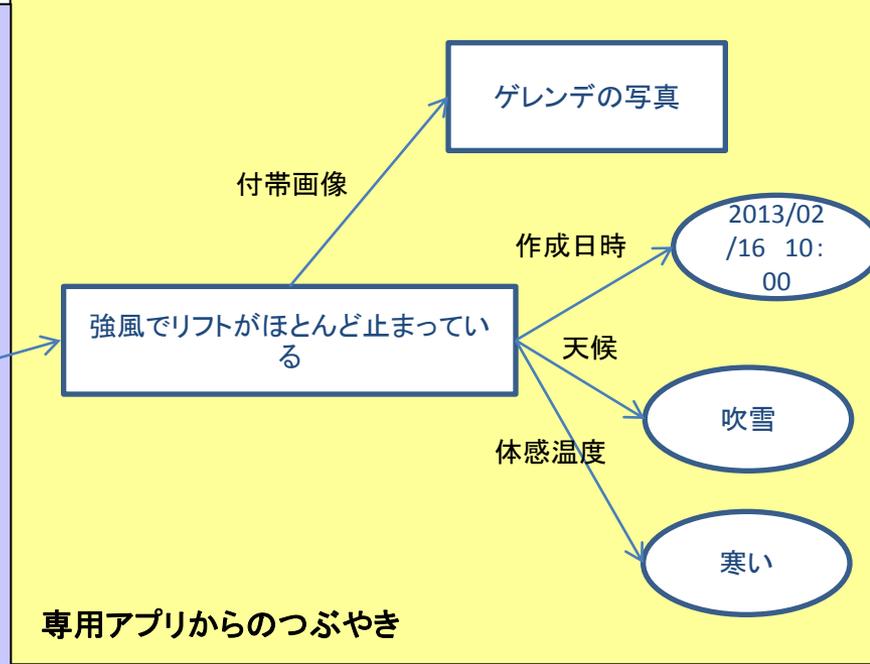
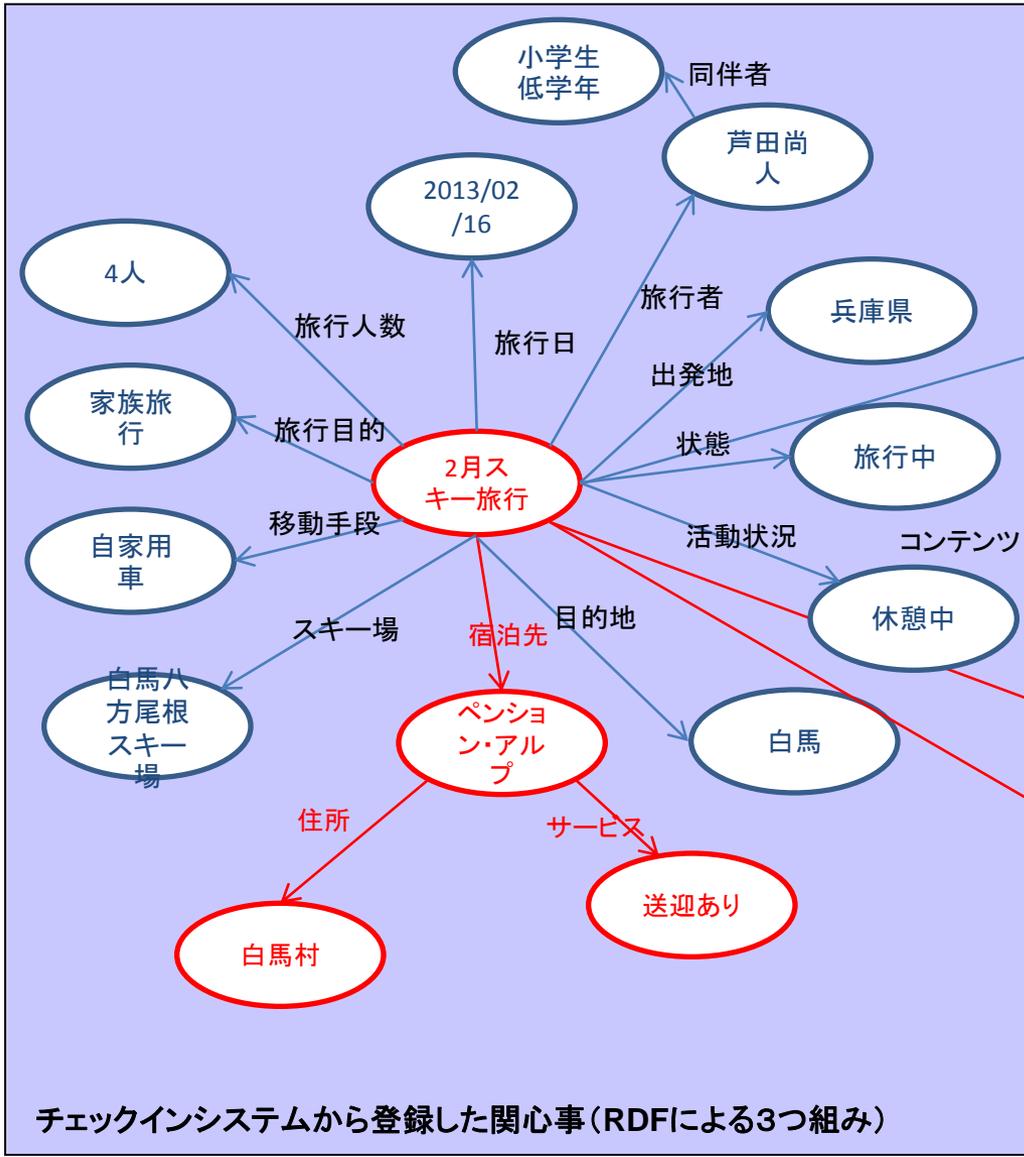
アドソル日進株式会社

荒本道隆

- 三つ組みデータ(RDF)をJenaで操作してみる
 - RDF: Resource Description Framework
 - W3Cで規格化されている
 - 触ってみれば、雰囲気は掴めるはず
- RDFデータを作ってみる
 - サンプルモデルをRDFデータに落とす
 - RDBのテーブル設計やXMLのスキーマ設計と同様、正しくなければ違和感があるはず
- SPARQL(RDFクエリ言語)を使ってみる
 - 利用例を見ても、よく分からないので

- Apacheのプロジェクトの1つ
 - <http://jena.apache.org/>
- Java製フレームワーク
 - Semantic Webアプリケーションを構築
 - RDFデータを読み、書き、処理するためのAPI
 - ファイルシステム
 - RDB (Oracle, MS-SQLServer, DB2, PostgreSQL, MySQL, Derby, H2, HSQLDB)
 - RDFとOWLを使ったルールベースの推論エンジン
 - OWL: Web Ontology Language
 - SPARQLのクエリーエンジン
 - RDFデータを公開するためのサーバ

関心事参加者とコンテンツ



- 主語、述語、目的語の3つの要素で表現
 - 主語 (Subject) : URI
 - <http://aitc.jp/LA/rdfs/1>
 - 述語 (Predicate) : URI
 - <http://aitc.jp/LA/model/ski#宿泊先>
 - 目的語 (Object) : 値
 - 「2月スキー旅行」
 - 「ペンション・アルプ」



- この部分をRDFにする場合



- 識別できるようにIDを追加



IDを振っただけだと

- これをそのままRDFにすると問題がある



- データがこうなる

主語	述語	目的語
1	タイトル	2月スキー旅行
2月スキー旅行	宿泊先	ペンション・アルプ

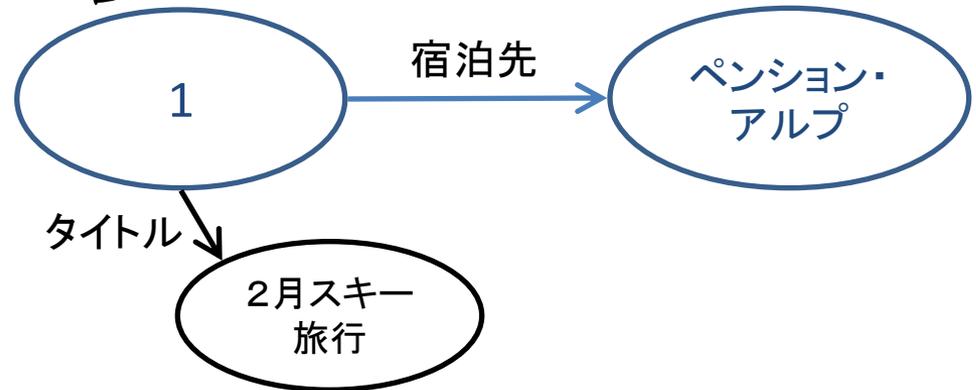
- 同じタイトルを付けられると区別がつかない

区別可能にするには

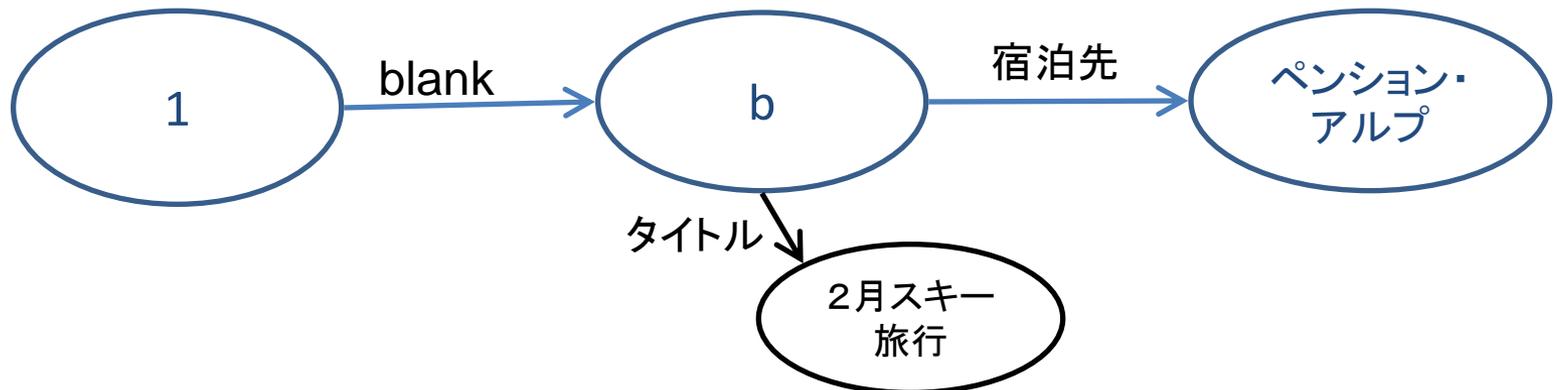
- 元



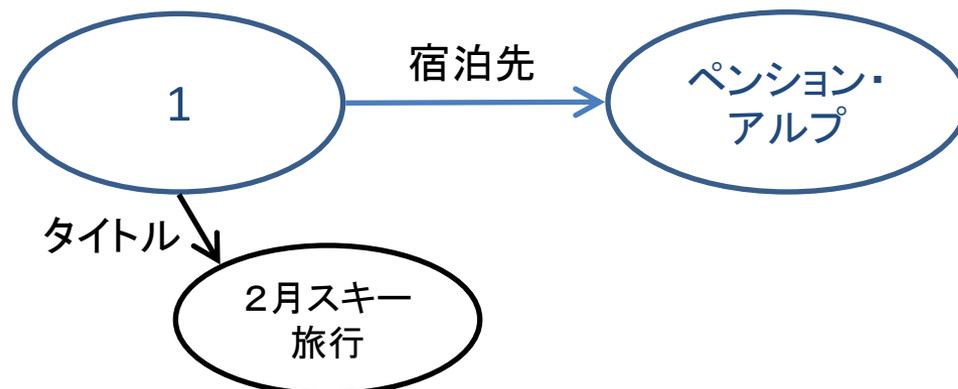
- IDを追加するパターン



- IDと空白ノードを追加するパターン



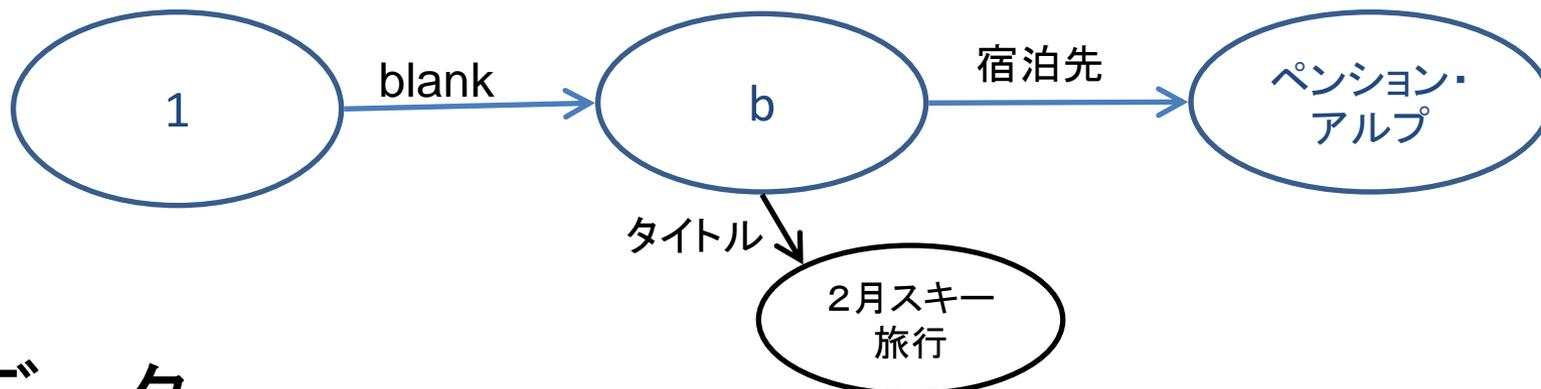
- モデル



- データ

主語	述語	目的語
1	タイトル	2月スキー旅行
1	宿泊先	ペンション・アルプ

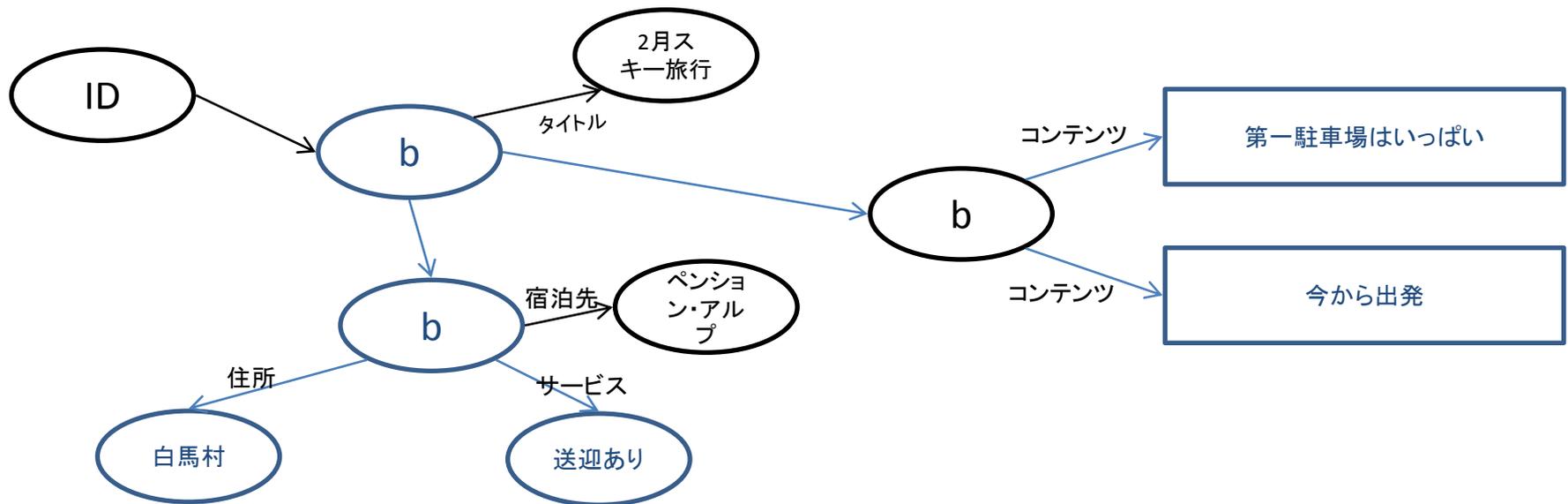
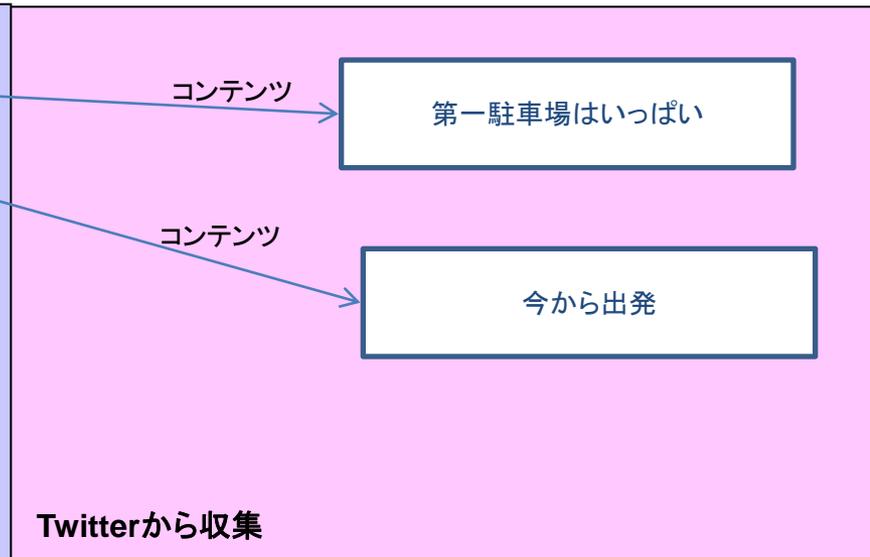
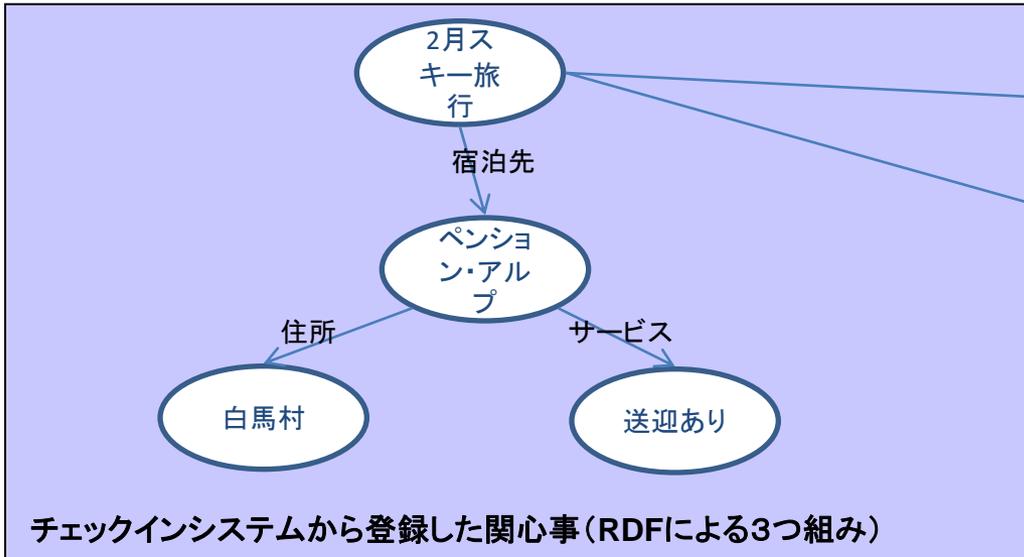
- モデル



- データ

主語	述語	目的語
1	node	b1
b1	タイトル	2月スキー旅行
b1	宿泊先	ペンション・アルプ

モデルをRDF化し易いよう変換



- IDをURIにする
 - <http://aitc.jp/LA/rdfs/1>
- 述語の書き方を決める
 - <http://aitc.jp/LA/model#blank>
 - <http://aitc.jp/LA/model#タイトル>
 - <http://aitc.jp/LA/model/ski#宿泊先>
 - <http://aitc.jp/LA/model/ski#住所>

Jenaを使ってRDFを作成する

```
// ファイルシステム上にデータセットを作成する
Dataset dataset = TDBFactory.createDataset("MyDatabases/Dataset1");
dataset.begin(ReadWrite.WRITE);

// モデルを作成する
Model model = dataset.getDefaultModel();

// IDを作成する
Resource b0 = model.createResource("http://aitc.jp/LA/rdfs/1");

// 空白ノードを作成し、ラベルを追加する
Resource b1 = model.createResource();
b0.addProperty(model.createProperty("http://aitc.jp/LA/model#", "blank" ), b1);
b1.addProperty(model.createProperty("http://aitc.jp/LA/model#", "タイトル"), "2月スキー旅行");
Resource b2 = model.createResource();
b1.addProperty(model.createProperty("http://aitc.jp/LA/model#", "blank" ), b2);
b2.addProperty(model.createProperty("http://aitc.jp/LA/model/ski#", "宿泊先"), "ペンション・アルプ");
b2.addProperty(model.createProperty("http://aitc.jp/LA/model/ski#", "住所"), "白馬村");
b2.addProperty(model.createProperty("http://aitc.jp/LA/model/ski#", "サービス"), "送迎あり");
b1.addProperty(model.createProperty("http://aitc.jp/LA/model#", "blank" ),
    model.createResource()
        .addProperty(model.createProperty("http://aitc.jp/LA/sns#", "twitter" ), "第一駐車場はいっぱい")
        .addProperty(model.createProperty("http://aitc.jp/LA/sns#", "twitter" ), "今から出発")
);
System.out.println("-----");
model.write(System.out, "RDF/XML-ABBREV");
System.out.println("-----");
model.write(System.out, "N-TRIPLE");
System.out.println("-----");

model.close();

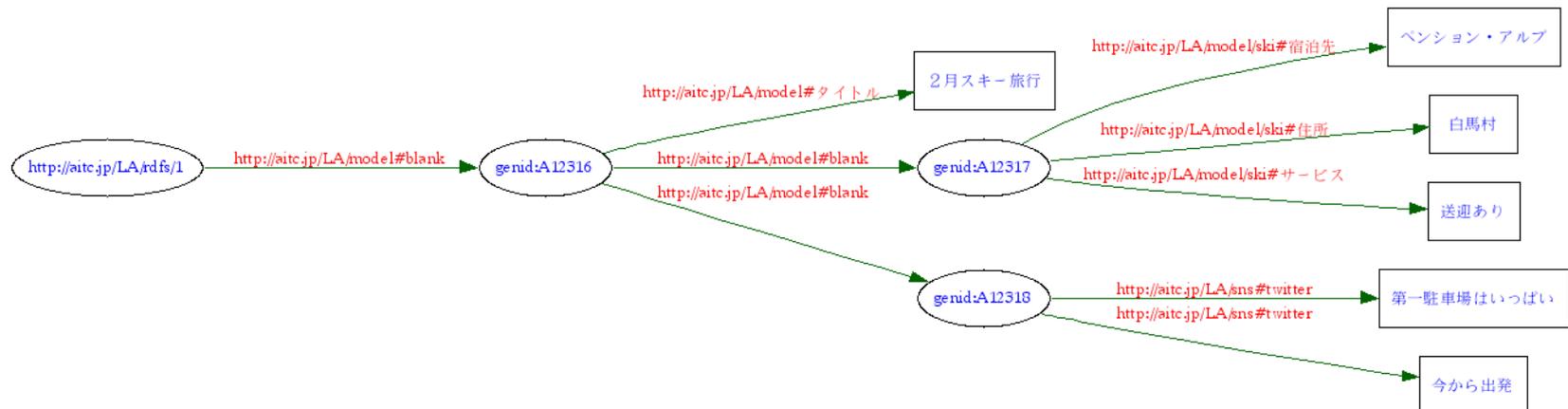
dataset.commit();
dataset.end();
```

```
<rdf:RDF
  xmlns:j.0="http://aitc.jp/LA/sns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.1="http://aitc.jp/LA/model/ski#"
  xmlns:j.2="http://aitc.jp/LA/model#">
  <rdf:Description rdf:about="http://aitc.jp/LA/rdfs/1">
    <j.2:blank rdf:parseType="Resource">
      <j.2:タイトル>2月スキー旅行</j.2:タイトル>
      <j.2:blank rdf:parseType="Resource">
        <j.1:宿泊先>ペンション・アルプ</j.1:宿泊先>
        <j.1:住所>白馬村</j.1:住所>
        <j.1:サービス>送迎あり</j.1:サービス>
      </j.2:blank>
      <j.2:blank rdf:parseType="Resource">
        <j.0:twitter>第一駐車場はいっぱい</j.0:twitter>
        <j.0:twitter>今から出発</j.0:twitter>
      </j.2:blank>
    </j.2:blank>
  </rdf:Description>
</rdf:RDF>
```

作成したRDFの検証方法

- W3Cの検証サイトに、出来たRDFを入れてみる
 - <http://www.w3.org/RDF/Validator/>

Number	Subject	Predicate	Object
1	http://aitc.jp/LA/rdfs/1	http://aitc.jp/LA/model#blank	genid:A12316
2	genid:A12316	http://aitc.jp/LA/model#タイトル	"2月スキー旅行"
3	genid:A12316	http://aitc.jp/LA/model#blank	genid:A12317
4	genid:A12317	http://aitc.jp/LA/model/ski#宿泊先	"ペンション・アルプ"
5	genid:A12317	http://aitc.jp/LA/model/ski#住所	"白馬村"
6	genid:A12317	http://aitc.jp/LA/model/ski#サービス	"送迎あり"
7	genid:A12316	http://aitc.jp/LA/model#blank	genid:A12318
8	genid:A12318	http://aitc.jp/LA/sns#twitter	"第一駐車場はいっぱい"
9	genid:A12318	http://aitc.jp/LA/sns#twitter	"今から出発"



別の表現方法

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:laProp="http://aitc.jp/projectLA/property/"
  xmlns:laSki="http://aitc.jp/projectLA/property/ski/">
<rdf:Description rdf:about="http://aitc.jp/projectLA/demo/concern/id001">
  <rdf:type rdf:resource="http://aitc.jp/projectLA/関心事"/>
  <rdfs:label xml:lang="ja">2月スキー旅行</rdfs:label>
  <laProp:コンテンツ rdf:parseType="Resource">
  <laProp:入手先 rdf:resource="http://aitc.jp/projectLA/入手先/twitter"/>
  <laProp:日時>2012-08-01T11:30:25.000-9:00</laProp:日時>
  <laProp:テキスト xml:lang="ja">第一駐車場はいっぱい</laProp:テキスト>
</laProp:コンテンツ>
  <laProp:コンテンツ rdf:parseType="Resource">
  <laProp:入手先 rdf:resource="http://aitc.jp/projectLA/入手先/twitter"/>
  <laProp:日時>2012-08-01T09:50:45.000-9:00</laProp:日時>
  <laProp:テキスト xml:lang="ja">今から出発</laProp:テキスト>
</laProp:コンテンツ>
  <laProp:宿泊先 rdf:parseType="Resource">
  <rdf:type rdf:resource="http://aitc.jp/projectLA/ホテル"/>
  <rdfs:label>ペンション・アルプ</rdfs:label>
  <laSki:住所>白馬村</laSki:住所>
  <laSki:サービス>送迎あり</laSki:サービス>
</laProp:宿泊先>
</rdf:Description>
</rdf:RDF>
```

別の表現方法

Number	Subject	Predicate	Object
1	http://aitc.jp/projectLA/demo/concern/id001	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://aitc.jp/projectLA/関心事
2	http://aitc.jp/projectLA/demo/concern/id001	http://www.w3.org/2000/01/rdf-schema#label	"2月スキー旅行"@ja
3	http://aitc.jp/projectLA/demo/concern/id001	http://aitc.jp/projectLA/property/コンテンツ	genid:A12946
4	genid:A12946	http://aitc.jp/projectLA/property/入手先	http://aitc.jp/projectLA/入手先/twitter
5	genid:A12946	http://aitc.jp/projectLA/property/日時	"2012-08-01T11:30:25.000-9:00"
6	genid:A12946	http://aitc.jp/projectLA/property/テキスト	"第一駐車場はいっぱい"@ja
7	http://aitc.jp/projectLA/demo/concern/id001	http://aitc.jp/projectLA/property/コンテンツ	genid:A12947
8	genid:A12947	http://aitc.jp/projectLA/property/入手先	http://aitc.jp/projectLA/入手先/twitter
9	genid:A12947	http://aitc.jp/projectLA/property/日時	"2012-08-01T09:50:45.000-9:00"
10	genid:A12947	http://aitc.jp/projectLA/property/テキスト	"今から出発"@ja
11	http://aitc.jp/projectLA/demo/concern/id001	http://aitc.jp/projectLA/property/宿泊先	genid:A12948
12	genid:A12948	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://aitc.jp/projectLA/ホテル
13	genid:A12948	http://www.w3.org/2000/01/rdf-schema#label	"ペンション・アルプ"
14	genid:A12948	http://aitc.jp/projectLA/property/ski/住所	"白馬村"
15	genid:A12948	http://aitc.jp/projectLA/property/ski/サービス	"送迎あり"

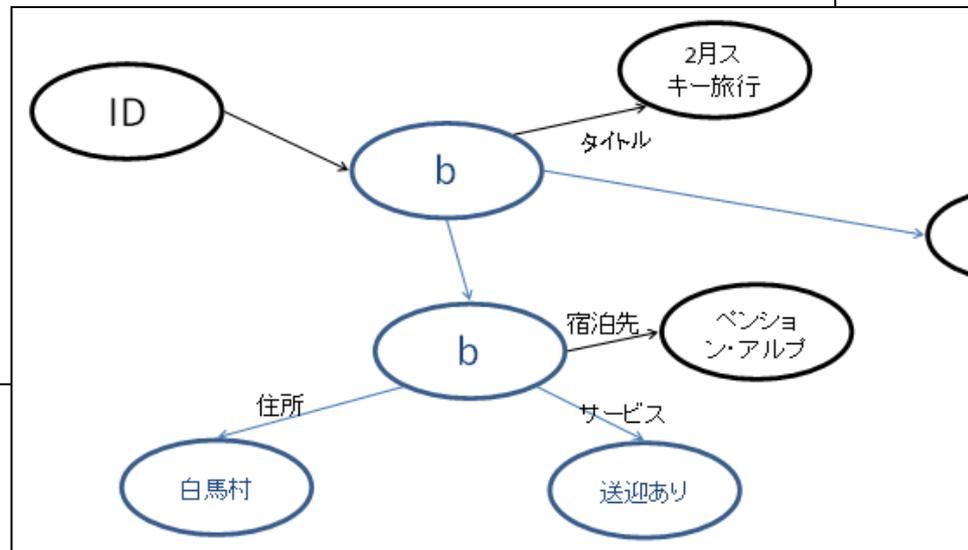


SPARQLで検索

```
// ファイルシステム上のデータセットを開く
Dataset dataset = TDBFactory.createDataset("MyDatabases/Dataset1");
dataset.begin(ReadWrite.READ);

// Query
QueryExecution qExec = QueryExecutionFactory.create(
    // "SELECT * {?s ?p ?o} LIMIT 100",
    "SELECT ?x ?y ?value WHERE "
        +" { ?x <http://aitc.jp/LA/model/ski#住所> \"白馬村\" . "
        +" ?y <http://aitc.jp/LA/model#blank> ?x ."
        +" ?y <http://aitc.jp/LA/model#タイトル> ?value ."
        +" }",
    dataset);
ResultSet rs = qExec.execSelect();
try {
    ResultSetFormatter.out(rs);
} finally {
    qExec.close();
}

dataset.end();
```



やってみて分かったこと

- 三つ組みデータ(RDF)をJenaで操作してみた
 - Javaで操作するのは簡単
 - 更新や部分更新については、未実装
- RDFデータを作ってみた
 - 考えなければならない事が、予想以上にあった
 - モデルの最適化、後々の拡張性を考慮
 - 述語の命名規約
 - 表記方法は1つじゃない
 - 記述する人によって、かなり差が出そう
- SPARQL(RDFクエリ言語)を使ってみた
 - 「ある条件に合致するのを取り出す」のは簡単
 - SQLで複数JOINするよりも直感的

- Internet上でサービスを運用
 - 他部会のアプリから利用できるように
- 述語の検討
 - 命名規約
- 更新の実装
 - 部分更新
 - SNSの発言を追加