Knowledge Graph Reasoning Challenge
ナレッジグラフ推論チャレンジ
生活行動における安心・安全を目指して
実社会版 2022

技術勉強会 2023/02/10 オンライン

提供データの解説とSPARQLによる検索

江上周作

産業技術総合研究所
人工知能研究センター
データ知識融合研究チーム

提供データセット

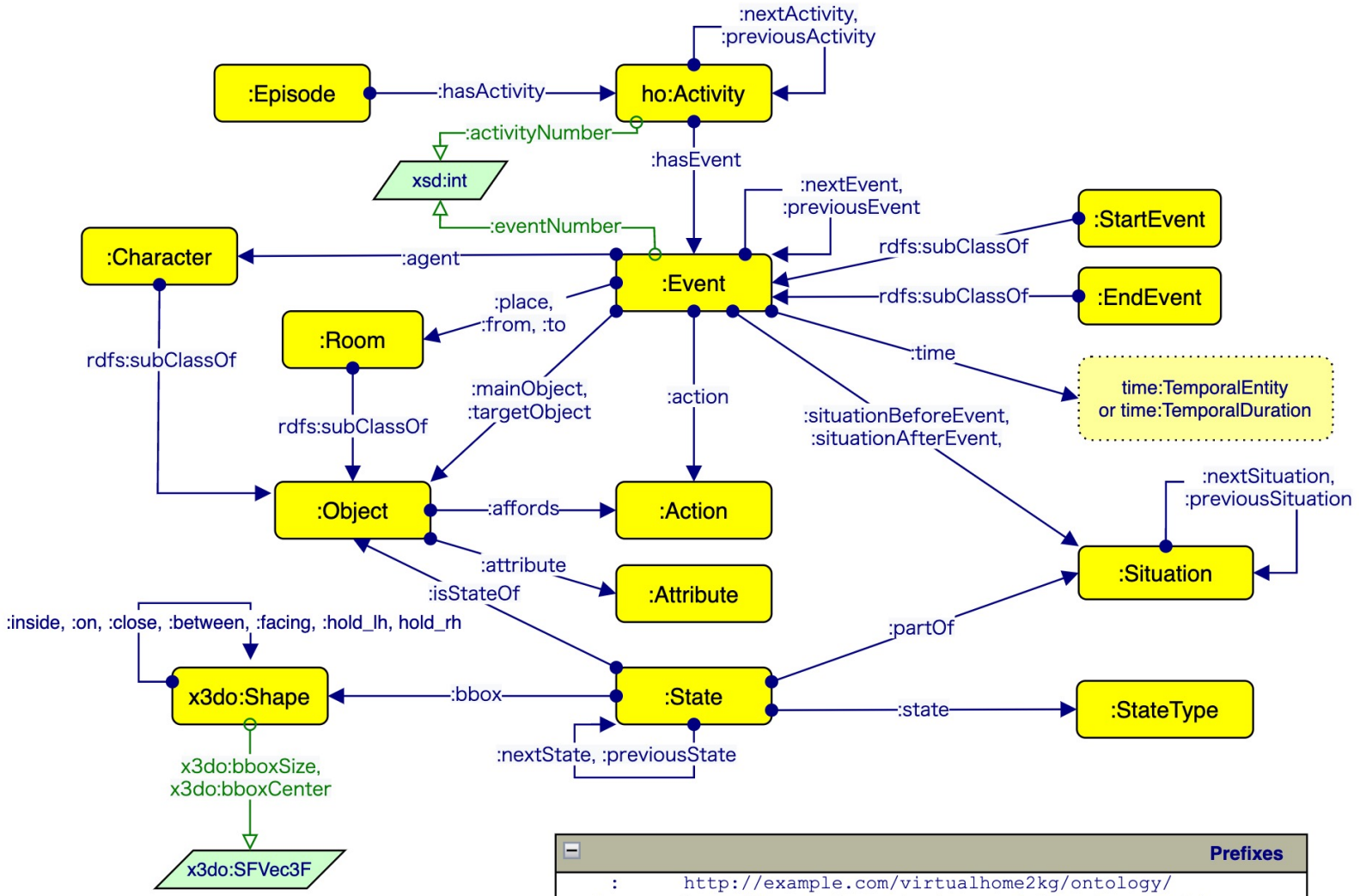

put_food_in_fridge
event10_put_food_in_fridge_scene1
event4_put_food_in_fridge_scene1
home_situation4_put_food_in_fridge_scene1
event6_put_food_in_fridge_scene1
scene1
open
event5_put_food_in_fridge_scene1
Put food in Fridge
virtualHome
hasEvent
nextEvent
situationAfterEvent
action
home_situation1_put_food_in_fridge_scene1
time_event3_put_food_in_fridge_scene1
event3_put_food_in_fridge_scene1
time
type
Event
agent
walk
character
situationAfterEvent
situationBeforeEvent
mainObject
event2_put_food_in_fridge_scene1
event1_put_food_in_fridge_scene1
kitchen
fridge
find
time_event1_put_food_in_fridge_scene1

提供データセット

- **家庭内日常生活を仮想空間でシミュレーションした動画と，その内容を表すナレッジグラフのセット**
  - https://github.com/aistairc/VirtualHome2KG/blob/main/README_ja.md
- 動画 (mp4)
  - 199種類の行動シナリオ(2023/2/10現在)
  - 1種類につきキャラクター後方視点（ファイル名末尾0），室内カメラ切替視点（ファイル名末尾1），部屋の四隅に設置した固定カメラ視点（ファイル末尾2～5）があり，合計1,206個の動画 (2023/2/10現在)
  - キャラクター動作がゆっくりな動画は高齢者の動きを再現しています
- ナレッジグラフ (RDF)
  - RDF形式
  - 動画に対応する199個のナレッジグラフ(2023/2/10現在)
  - SPARQLエンドポイントやクエリ例はこちら
- 台本データ (txt)
  - 動画とナレッジグラフを生成するためにVirtualHome2KGに与えたデータ
  - 行動のタイトルと簡単な文章説明を含む

ナレッジグラフのスキーマ

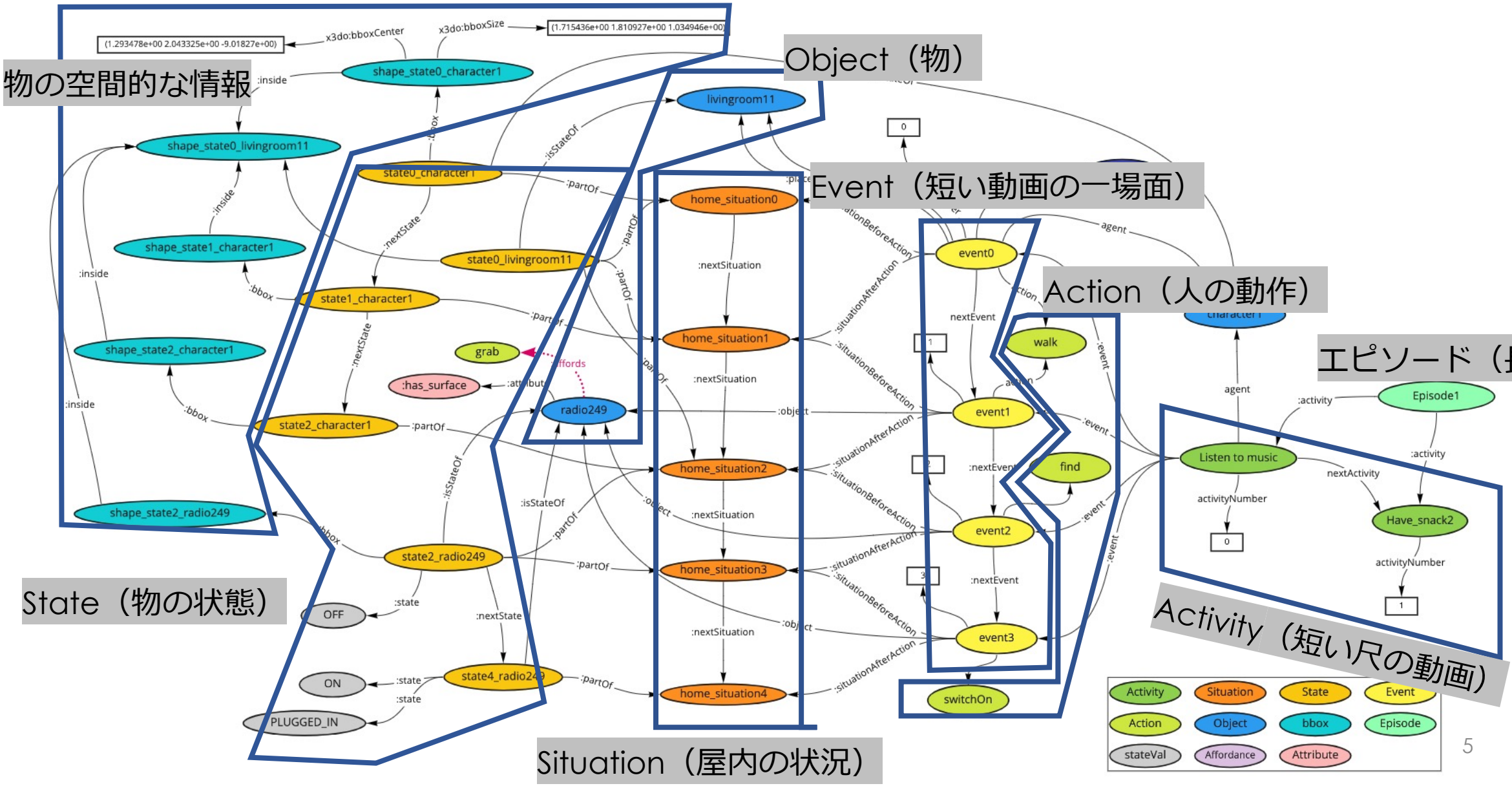
- イベントノードを中心としたKG（Event-centric KG）
- Activityは複数のイベントで構成される
- 各イベントは必ず一つのActionを持つ
- 「AをBの上に置く」などのオブジェクトが二つ必要なイベントの場合、Aをmain Object, BをtargetObjectとする
  - ho:objectのサブプロパティである
- Situationはイベントの実行前後のタイミングで作成される
- オブジェクトの状態「State」はいずれかのSituationに紐づく
  - 状態変化がなければ複数のSituationに紐づく
- オブジェクトの位置情報は3次元座標
  - 中心点
  - 3D bounding boxのサイズ



Prefixes

:	http://example.com/virtualhome2kg/ontology/
ho:	http://www.owl-ontologies.com/VirtualHome.owl#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
time:	http://www.w3.org/2006/time#
xsd:	http://www.w3.org/2001/XMLSchema#
x3do:	https://www.web3d.org/specifications/X3dOntology4.0#

ナレッジグラフの例


物の空間的な情報
Object（物）
Event（短い動画の一場面）
Action（人の動作）
エピソード（長尺）
State（物の状態）
Activity（短い尺の動画）
Situation（屋内の状況）
shape_state0_character1
shape_state0_livingroom11
shape_state1_character1
shape_state2_character1
shape_state2_radio249
state0_character1
state0_livingroom11
state1_character1
state2_character1
state2_radio249
state4_radio249
livingroom11
home_situation0
home_situation1
home_situation2
home_situation3
home_situation4
radio249
grab
:has_surface
event0
event1
event2
event3
switchOn
walk
find
Episode1
Listen to music
Have_snack2
OFF
ON
PLUGGED_IN
Activity
Situation
State
Event
Action
Object
bbox
Episode
stateVal
Affordance
Attribute

SPARQLエンドポイント

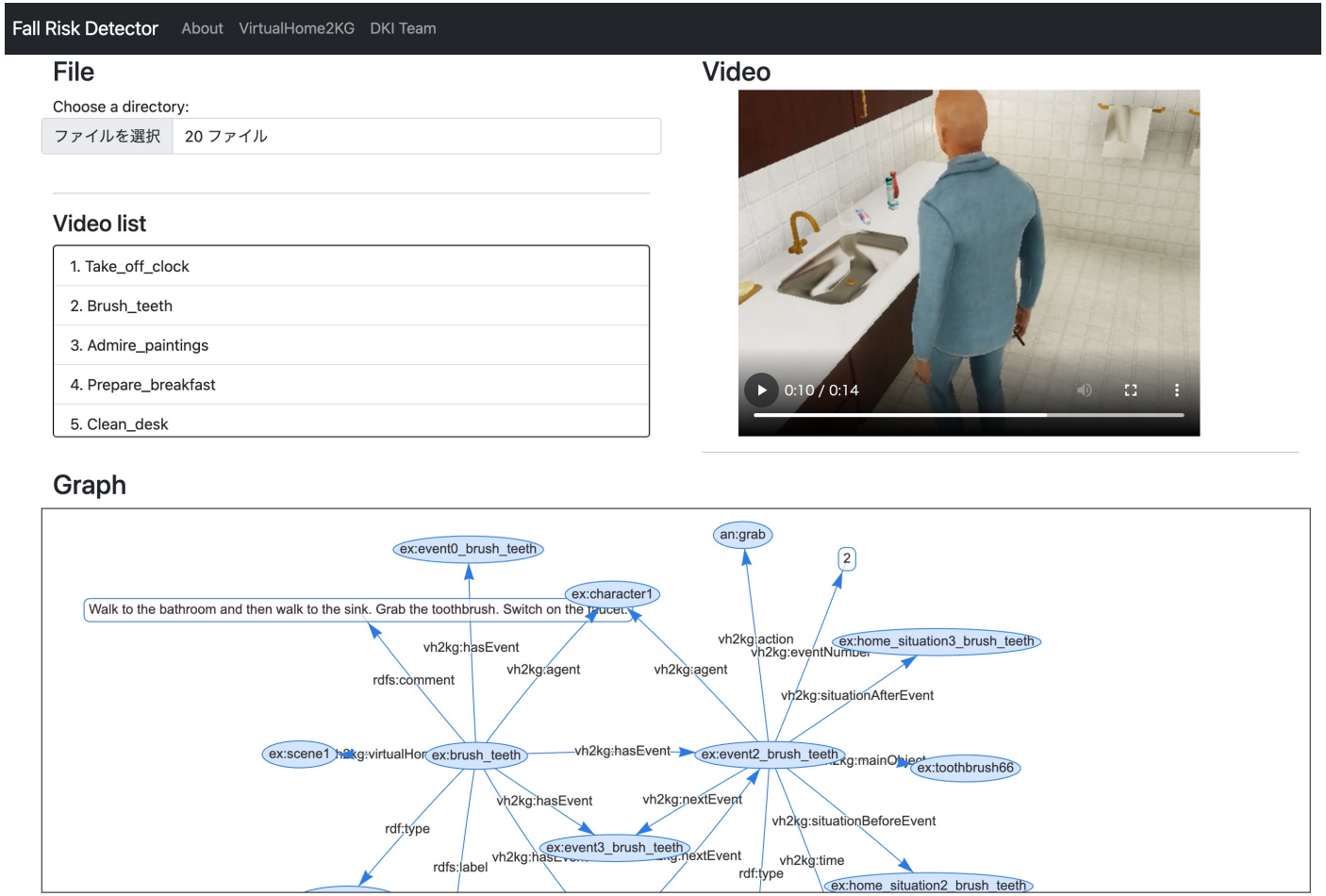
- SPARQLエンドポイントを提供しています。
  http://kgrc4si.ml:7200/sparql

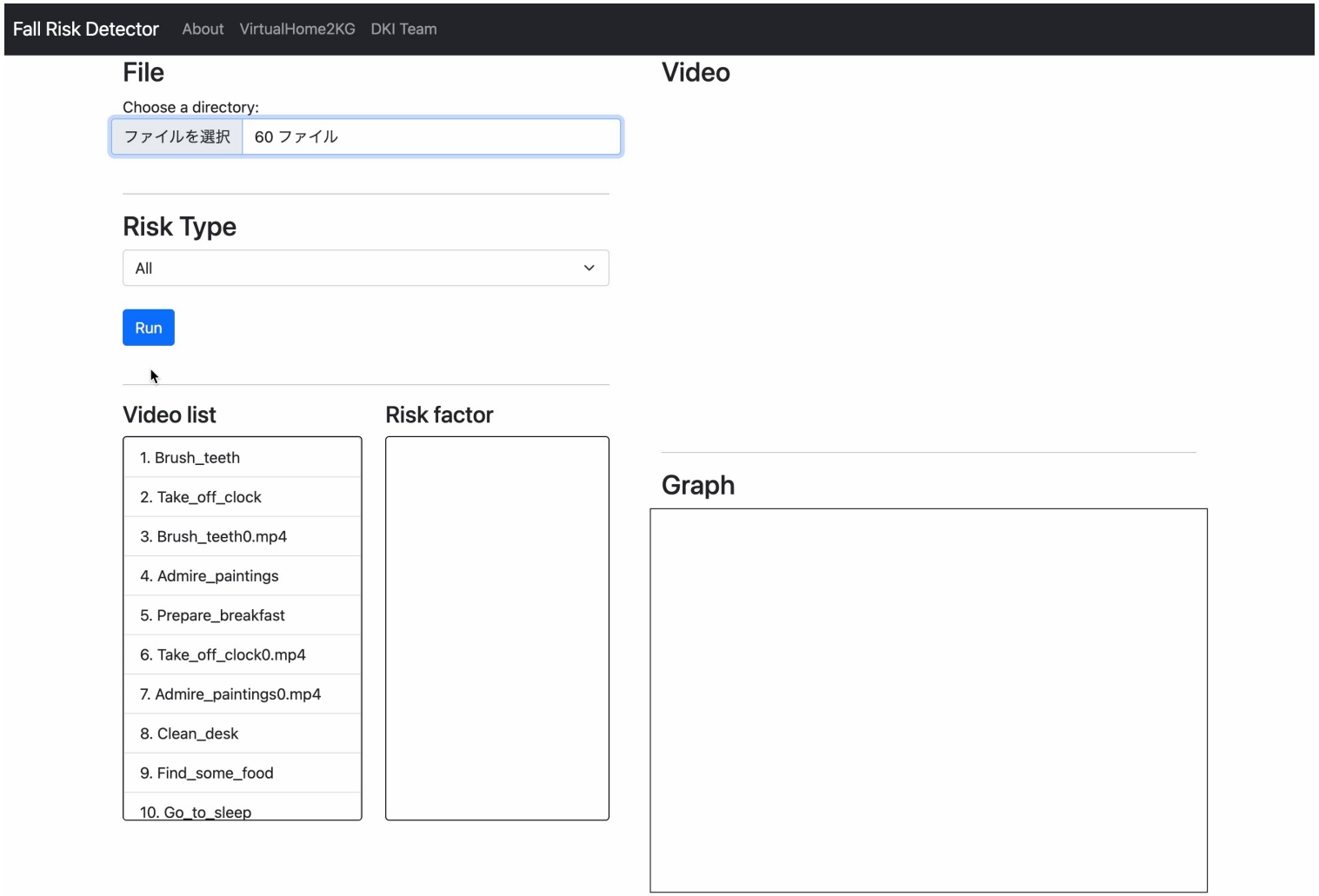
- リポジトリは「KGRC4SIv01」を選択してください(2023/02/10時点)

- トリプルストアとしてOntotext GraphDBを使用しています
- 使用方法は次のセッションで紹介

可視化ツールで確認

- 近日中に最新版を公開予定
  - 旧バージョン
  - https://github.com/aistairc/vh2kg_risk_ui/tree/swo2022-11-22
  - ローカルPCにダウンロード
  - ローカルPCのGraphDBにKGRC4SIv0というリポジトリを立てて、RDFを格納 が必要です。
  - その後、ブラウザでindex.htmlを開き、「ファイルを選択」から動画一式（ディレクトリごと）読み込んで使用します。




Fall Risk Detector
About
VirtualHome2KG
DKI Team
File
Choose a directory:
ファイルを選択
60 ファイル
Risk Type
All
Run
Video list
1. Brush_teeth
2. Take_off_clock
3. Brush_teeth0.mp4
4. Admire_paintings
5. Prepare_breakfast
6. Take_off_clock0.mp4
7. Admire_paintings0.mp4
8. Clean_desk
9. Find_some_food
10. Go_to_sleep
Risk factor
Video
Graph

SPARQL

ナレッジグラフを自由に探索できる標準Web API

SPARQLとは？

- RDFデータを検索するためのクエリ言語
  （RDBMSにおけるSQLに相当）
- Web標準化団体（W3C）により正式勧告
  - ver. 1.0（2008年1月にW3C勧告）
    - http://www.w3.org/TR/rdf-sparql-query/
    - http://www.asahi-net.or.jp/~ax2s-kmtn/internet/rdf/rdf-sparql-query.html
      （日本語訳）
  - ver. 1.1（2013年3月にW3C勧告）
    - http://www.w3.org/TR/sparql11-overview
    - http://www.asahi-net.or.jp/~ax2s-kmtn/internet/rdf/REC-sparql11-query-20130321.html
      （日本語訳）

グラフデータベース（トリプルストア）

- **RDF等のグラフデータを格納し，SPARQLによる操作を可能にするデータベース**
  - トリプルストア，RDFストアとも呼ばれる
- オープンソース，商用ともに様々なトリプルストアが存在

Ontotext GraphDB	Blazegraph
Virtuoso	AllegroGraph
Amazon Neptune	Stardog
RDFox	Oracle database
KATANA GRAPH	IBM Graph
JanusGraph	TigerGraph
MarkLogic	Neo4j

詳しくはこちら https://en.wikipedia.org/wiki/Comparison_of_triplestores

SPARQL基本パターン

```
SELECT  ?s ?p ?o
WHERE {
        ?s ?p ?o .
}
```

SELECT句

以降のWHERE句内に記述するトリプルパターンから結果として返す変数を直後で指定

```
SELECT  ?s ?p ?o

WHERE {

        ?s ?p ?o .

}
```

変数

変数は先頭に「?」をつける。
SELECT句の直後に指定した変数を結果として返す

```
SELECT  ?s ?p ?o

WHERE {

        ?s ?p ?o .

}
```

WHERE句

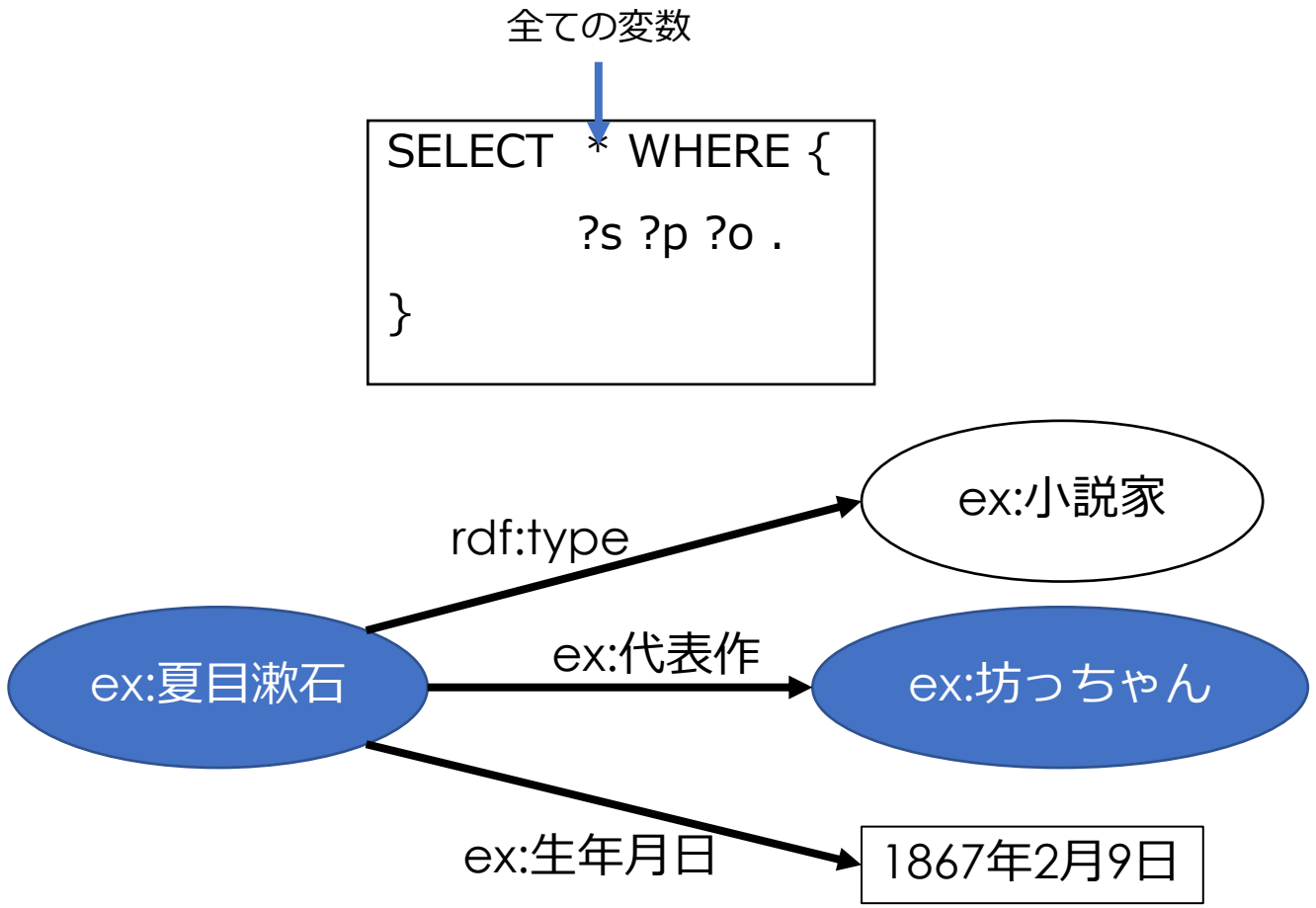
必ず {} で括る。
{} 内に検索するトリプルパターンを記述する.
{} 内のトリプルはTurtleと記法が同じ.

```
SELECT  ?s ?p ?o

WHERE {

        ?s ?p ?o .

}
```

この場合，主語・述語・目的語が全て変数なので，全てのトリプルパターンを検索している

検索イメージ

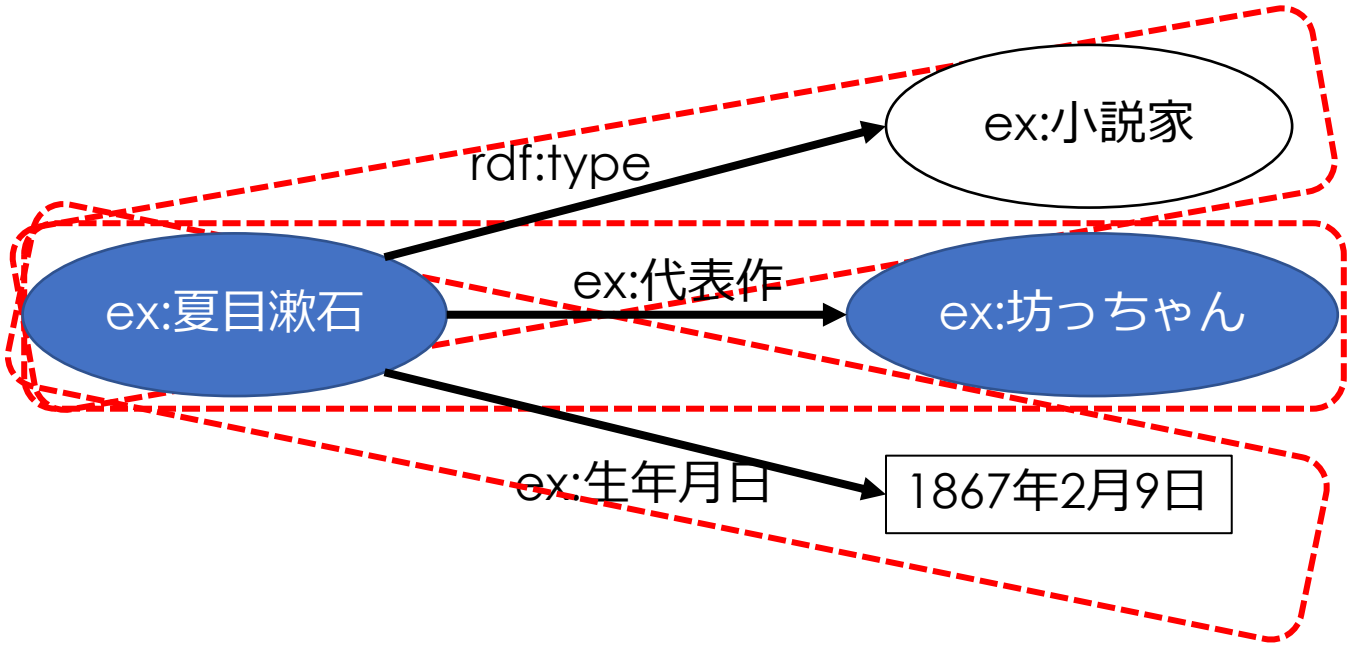
- 下図のグラフに対して次のSPARQLを実行する場合



検索イメージ

- WHERE句内のトリプルパターンに赤枠部分が該当

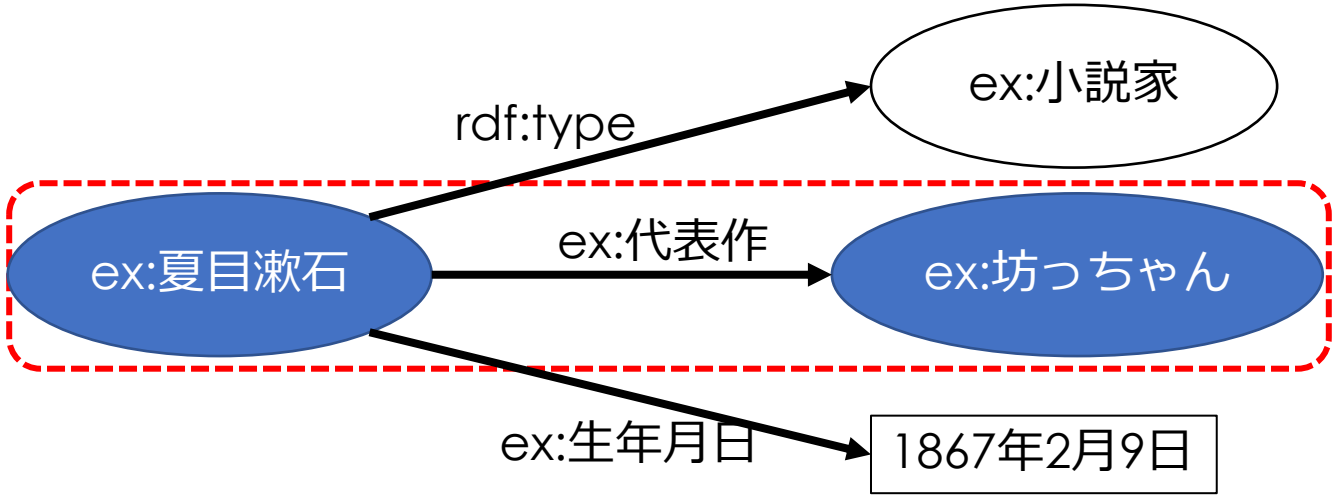
```
SELECT  * WHERE {
        ?s ?p ?o .
}
```



検索イメージ

- WHERE句内のトリプルパターンに赤枠部分が該当

```
SELECT  * WHERE {
        ?s ex:代表作 ?o .
}
```



SPARQL検索

- アクティビティの一覧を取得する

```
PREFIX ex: <http://example.org/virtualhome2kg/instance/>
PREFIX : <http://example.org/virtualhome2kg/ontology/>
select DISTINCT * where {
    ?activity :virtualHome ex:scene1 .
}
```

実行結果

- 「インターネットをブラウズする」というアクティビティ中のイベントとアクションを取得する

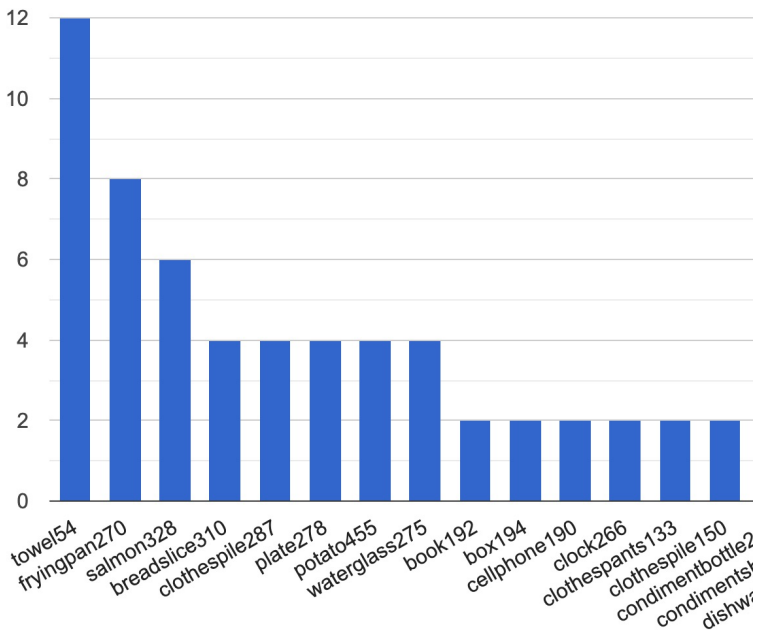
```
PREFIX ex: <http://example.org/virtualhome2kg/instance/>
PREFIX : <http://example.org/virtualhome2kg/ontology/>
select DISTINCT * where {
    ex:browse_internet_scene1 :hasEvent ?event .
    ?event :action ?action .
}
```

実行結果

SPARQL検索

- よく掴まれているオブジェクト

```
PREFIX ho: <http://www.owl-ontologies.com/VirtualHome.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX : <http://example.org/virtualhome2kg/ontology/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX ac: <http://example.org/virtualhome2kg/ontology/action/>
select ?name (count(?object) AS ?count) where {
        ?objectClass rdfs:subClassOf :Object .
    ?object a ?objectClass ;
            rdfs:label ?label ;
            dcterms:identifier ?id .
    ?event ho:object ?object .
    ?event :action ac:grab .
    BIND(concat(?label, ?id) AS ?name)
} group by ?object ?name order by desc(count(?object))
```



実行結果

SPARQL検索

- オブジェクトの高さ情報を追加する

```
PREFIX x3do: <https://www.web3d.org/specifications/X3dOntology4.0#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/virtualhome2kg/ontology/>
PREFIX ex: <http://example.org/virtualhome2kg/instance/>
CONSTRUCT {
    ?object :height ?height_node .
    ?height_node rdf:value ?size_y1 ;
           :unit :meter .
} WHERE {
        ?state1 :isStateOf ?object ; :bbox ?shape1 .
        ?shape1 x3do:bboxSize ?size1 .
        ?size1 rdf:rest ?size_y .
    ?size_y rdf:first ?size_y1 .
    BIND(REPLACE(STR(?object), STR(ex:) ,"") AS ?object_name)
    BIND(URI(CONCAT(STR(ex:),"height_", ?object_name)) AS ?height_node)
}
```

実行結果