

Knowledge-based Patient Data Generation

Zhisheng Huang, Frank van Harmelen, Annette ten Teije, and Kathrin Dentler

Department of Computer Science,
VU University Amsterdam, The Netherlands
{huang, Frank.van.Harmelen, annette, k.dentler}@cs.vu.nl

Abstract. The development and investigation of medical applications require patient data from various Electronic Health Records (EHR) or Clinical Records (CR). However, in practice, patient data is and should be protected and monitored to avoid unauthorized access or publicity, because of many reasons including privacy, security, ethics, and confidentiality. Thus, many researchers and developers encounter the problem to access required patient data for their research or make patient data available for example to demonstrate the reproducibility of their results. In this paper, we propose a knowledge-based approach of synthesizing large scale patient data. Our main goal is to make the generated patient data as realistic as possible, by using domain knowledge to control the data generation process. Such domain knowledge can be collected from biomedical publications such as PubMed, from medical textbooks, or web resources (e.g. Wikipedia and medical websites). Collected knowledge is formalized in the Patient Data Definition Language (PDDL) for the patient data generation. We have implemented the proposed approach in our Advanced Patient Data Generator (APDG). We have used APDG to generate large scale data for breast cancer patients in the experiments of SemanticCT, a semantically-enabled system for clinical trials. The results show that the generated patient data are useful for various tests in the system.

1 Introduction

Research and development of medical applications require the use of electronic patient data. This patient data can be obtained either from Electronic Health Records (EHR), which are systematic collections of electronic health information about individual patients or populations, or from Clinical Records (CR), which are collections of personal medical information recorded by clinicians[2]. However, in practice, patient data is protected and monitored to avoid unauthorized access or publicity, because of many reasons, such as privacy, security, ethics, confidentiality, etc. These circumstances make the use of patient data for research hard, and block the publicity of relevant patient data used in the research for public evaluation.

Therefore, an important research question is *"whether it is possible to develop a tool which can be used to create virtual and most importantly realistic patient data?"* The advantage of generated patient data is obvious, as it would not lead to any privacy problems.

We consider the following use cases of the generated patient data:

- **Availability.** The generated patient data can be used by developers to test and evaluate prototypes, without waiting for the approval from the authority or even the patients themselves. Developers mainly care about the format and the quality of the data. By defining the required format, data can easily be generated based on realistic distributions. Such required patient data would always be available for system developers for tests. Public datasets might also prove to be useful for comparison and benchmark of medical applications.
- **Publicity.** The generated patient data can be published and made available for public in any circumstance. Researchers can use those data for explanation of experiments and evaluation of their research. Those generated patient data are sufficient for the experiments in the developments of medical knowledge/information systems.
- **Complementary.** The quality of real patient data is not perfect. Some data values may be missing, erroneous, noisy, or inconsistent. The quality of generated patient data might depend on the preferences of the user. The generator could produce high-quality data for researchers and developers who prefer high quality of data as well as more realistic data.
- **Rarity.** We can use the patient data generator to create some patient data of rare diseases, because their original data is too rare to be obtained. The patient data generator can generate the required patient data, based on existing medical findings, and make those rare data available for demonstration of a prototyping system.
- **Typicality.** For the evaluation and benchmarking of an e-Health system or tool, sometimes we require that the benchmarks or tested data are not biased towards any data feature. Namely, the benchmarks should be typical enough to cover wide range of realistic data. The patient data generator can be used to create those typical data.

It is quite clear that the use cases above are useful for system developers and researchers. Thus, the next research question is: *"How can such a tool for patient data generation be built, with the generated data as realistic as possible?"* This is exactly the question we answer in this paper. The main idea is to use all the domain knowledge we can collect to control the patient data generation. Such domain knowledge can be collected from biomedical publications such as PubMed, domain knowledge from medical textbooks, and web resources such as Wikipedia and medical webpages.

In this paper, we propose a knowledge-based approach of synthesizing large scale patient data. Collected knowledge is formalized in the Patient Data Definition Language (PDDL) and used for the patient data generation. We have implemented the proposed approach in Advanced Patient Data Generator (APDG)¹, and used APDG to create patient data, which includes large data sets for breast cancer patients for the experiments in the SemanticCT system, a semantically-enabled knowledge system for clinical trials[7,6]. Those experiments include patient recruitment service (i.e., identifying eligible patients for a trial), trial finding service (i.e., finding suitable trials for a patient), and protocol feasibility service

¹ <http://wasp.cs.vu.nl/apdg>

(i.e., design eligibility criteria for a trial). The results show that the generated patient data is useful for various tests in the system.

This paper is organized as follows: Section 2 presents a framework of knowledge-based patient data generation. Section 3 proposes the patient data definition language (PDDL). Section 4 discusses the implementation of APDG. Section 5 reports several experiments of APDG. After a discussion of related work in Section 6, the last section includes conclusions and future work.

2 Framework

2.1 Patient Data

We are going to design a system which can generate patient data, based on the formalized domain knowledge. In particular we will focus on the generation of EHR data. Naturally, the first question here is what kinds of standardization for the data would be used for our system. There have been several initiatives to standardize a generic EHR architecture of patient data. Well-known EHR architectures are the archetype-based ones[1], like openEHR². Those archetype-based EHR architectures introduce the two-level approach, reference model level and archetype level, for the specification of the structure and semantics of patient data. Archetypes are reusable and domain-specific definitions of clinical concepts in the structured and constrained combinations of entities of the reference model, which represents the generic and stable properties of patient data. From the perspective of computer science, we call those entities of the reference model *slots*.

In APDG, we introduce a similar architecture of patient data, like those archetype-based EHR ones. We consider the architecture of patient data as a set of data which consists of the following three levels: Session-Archetype-Slot. Namely, sessions are considered to be a collection of archetypes which have been instantiated with slots from a reference model.

2.2 Knowledge for Patient Data Generation

As we have discussed above, we will collect relevant domain knowledge for patient data generation. This domain knowledge can be collected from the following resources.

- **Biomedical publications.** Biomedical publications such as PubMed and medical books provide rich information about diseases and patients. For example, we can find the description of distant metastases for breast cancer patients from a publication in PubMed³:

We found distant metastases at the time of primary diagnosis in 19 patients (3.9%). Bone metastases were found in 2.7%, liver metastases in 1.0%, and pulmonary metastases in 0.4%. However,

² <http://www.openehr.org/>

³ <http://www.ncbi.nlm.nih.gov/pubmed/14605816>

in breast tumors smaller than 1 cm, no metastatic lesions were found, whereas 18.2% of the patients with pT4 tumors had metastases. In 2.4% of screening imaging studies, metastases were ruled out by additional imaging.

The knowledge above can be used to define the distant metastases of breast cancer patients and their metastases sites.

- **Web resources.** Web resources such as Wikipedia and medical websites usually provide information about the distribution and its dependence on other variables (such as gender, age, etc.) for diseases. For example, we can find the following information about the distribution of breast cancer stage from the web page⁴

Data on around 17,800 women diagnosed with breast cancer in the East of England in 2006-2009 shows that, of the 92% of cancers for which a stage was recorded, 41% were Stage I, 45% stage II, 9% stage III and 5% stage IV.

This knowledge can be used to generate patient data with the stage information.

Because the information provided by various resources above may be differing, we can design a preference ordering to evaluate those data resources, so that some information would be preferred to other ones in case of inconsistencies between resources. For the temporal aspect, we would prefer the data which occurs latest to data which occurs earlier. For the trust aspect, we would prefer data which appears in scientific publications (e.g., those in PubMed or Medical textbook) to data that appears in websites.

On the other hand, the collected data may not cover exactly what we are expecting to get. However, it would not lead to a serious problem if we consider approximate patient data acceptable. For example, although the distributions above are stated for specific area (East of England) and specific period (2006-2009), we can use this knowledge to provide an approximate estimation for the distribution if we cannot find any information which states that the data is too specific for that area in that period and they are significantly different from other areas or other periods.

We formalize the collected domain knowledge in a formalism which is called *Patient Data Definition Language (PDDL)* for the procedural control of patient data generation. Namely we will embed formalized domain knowledge into the Session-Archetype-Slot structure of patient data for the control. Given a disease, embedded control knowledge is expected to be added by clinical professionals or knowledge engineers, who have trustful knowledge about the disease and know how to formalize the knowledge exactly. However, we would not expect those domain experts to have an intensive training to learn how to formalize comprehensive knowledge in Artificial Intelligence to use APDG. Thus, We design the PDDL to make it based on an XML-based text document, which would make the formulation easy for clinical professionals or knowledge engineers. Furthermore,

⁴ <http://www.cancerresearchuk.org/cancer-info/cancerstats/types/breast/incidence/uk-breast-cancer-incidence-statistics#stage>

we have also implemented a user-friendly GUI for users, so that they need no any intensive knowledge about XML to use the APDG tool[11].

We generate patient data for breast cancer with APDG by modeling the medical knowledge into PDDL by ourselves. The mapping from the knowledge from several sources into the PDDL model needs to be validated with an expert. The contribution of this paper is the possibility of generating patient data on a knowledge intensive way that exploits the available medical knowledge, and can be used in studying several medical tasks like patient recruitment for clinical trials.

3 Patient Data Definition Language (PDDL)

3.1 General Components

The Patient Data Definition Language (PDDL) is designed to be an XML-based language to define the general format of the patient data and its relevant domain knowledge to control the procedure of patient data generation. Thus, PDDL is concerned with the following information.

- **Patient Data Format.** It defines the structure of patient data by stating which Session-Archetype-Slot structure will be used for the generated patient data.
- **Domains and Ranges.** It defines what kinds of domains and ranges of patient data are allowed for the generated patient data.
- **Distribution.** It provides value distribution statements for each slot.
- **Dependence.** It defines value dependence among variables in the patient data.

In the next subsections we discuss those types of information in more detail.

3.2 Patient Data Format

We use the general structure, i.e., ‘Session-Archetype-Slot’ for patient data generation. That structure is stated with the following XML format:

```
<Session value="DemographicData">
  <Archetype concept = "Patient">
    <Slot value="LastName" type="string"/>
    <Slot value="FirstName" type="string"/>
    <Slot value="Gender" type="string"/>
    <Slot value="BirthYear" type="year"/>
  </Archetype>
</Session>
```

Each entity (i.e. session, archetype, or slot) has a value property to define the entity name. An archetype is allowed to contain other (non-recursive) archetypes or slots. Slots are termination tabs which are used to state the possible values of the slots.

3.3 Domain Ranges

Data ranges in PDDL are defined by using the `DataRange` tab and the data type specification, like those in the following format which defines the enumeration values:

```
<Slot value="Gender">
  <DataRange>
    <enumeration value="female"/>
    <enumeration value="male"/>
  </DataRange>
</Slot>
```

or in the following format which defines the range of the allowed values for the slot by using the `maxInclusive` and `minInclusive` tabs.

```
<Slot value="BirthYear">
  <DataRange>
    <maxInclusive datatype="http://www.w3.org/2001/XMLSchema#date"
    >2006</maxInclusive>
    <minInclusive datatype="http://www.w3.org/2001/XMLSchema#date"
    >1900</minInclusive>
  </DataRange>
</Slot>
```

3.4 Distribution

A data distribution is defined inside the `DataRange` with the special tab ‘distribution’. A distribution value is designed to take a real number between 0 and 100, like this:

```
<Slot value="Gender">
  <DataRange>
    <enumeration value="female"/>
    <enumeration value="male"/>
    <Distributions type="enumeration">
      <Distribution item="female" pfrom="0" pto="100"/>
      <Distribution item="male" pfrom="0" pto="0"/>
    </Distributions>
  </DataRange>
</Slot>
```

Each distribution tab defines its data type of the slot, followed by a list of the distribution which states the value (i.e., item for the enumeration type) and its distribution range by the pair *pfrom* and *pto*. The example above states that the 100 percent (i.e., from 0 to 100) of patients are female, zero percent of patients (i.e. from 0 to 0) are male. For the non-enumeration data range, we use the properties (*from* and *to*) to define the value range, like this⁵:

```
<Slot value="BirthYear">
  <DataRange>
    <Distributions type="year" variable="$birthyear">
      <Distribution from="1998" to="2006" pfrom="0" pto="0"/>
      <Distribution from="1983" to="1997" pfrom="0" pto="0"/>
      <Distribution from="1973" to="1982" pfrom="0" pto="4.36"/>
      ....
      <Distribution from="1900" to="1932" pfrom="84.35" pto="100"/>
    </Distributions>
  </DataRange>
</Slot>
```

⁵ The value of the slot `BirthYear` will be reused in other slots. Thus, we introduce a variable for it.

If a distribution statement already contains the information of the datatype for the slot (by stating the type), its reference model tabs (like enumeration, maxInclusive, minInclusive, etc.) can be ignored.

A distribution can be stated by its distribution type (e.g., uniform random, normal distribution, etc.) on an enumeration set, like those in the following example:

```
<Slot value="DiagnosisMonth" type="month">
  <DataRange>
    <Distributions type="enumeration">
      <Distribution disttype="uniformrandom"
        set="1,2,3,4,5,6,7,8,9,10,11,12"/>
    </Distributions>
  </DataRange>
</Slot>
```

or by stating a data range (with a type) over the distribution, for instance the data range (minimax) over integers from 1927 to 2000 with the uniform random distribution⁶.

```
<Slot value="BirthYear">
  <DataRange>
    <Distributions type="year" variable="$birthyear">
      <Distribution disttype="uniform" datatype="minmax(int)"
        data="1927,2000"/>
    </Distributions>
  </DataRange>
</Slot>
```

which states the data range (minmax) over integers from 1927 to 2000 with the uniform random distribution.

3.5 Dependence

The condition statements are used to state the conditions which depend on some variables which have been defined in the previous distributions slots, like this:

```
<Slot value="MenopausalStatus">
  <DataRange>
    <Distributions type="enumeration" variable="$menopausalstatus">
      <Distribution item="premenopausal" pfrom="0" pto="100"
        condition="$birthyear > 1970"/>
      <Distribution item="perimenopausal" pfrom="0" pto="80"
        condition="$birthyear < 1970 AND $birthyear > 1950"/>
      <Distribution item="postmenopausal" pfrom="80" pto="100"
        condition="$birthyear < 1970 AND $birthyear > 1950"/>
      <Distribution item="postmenopausal" pfrom="0" pto="100"
        condition="$birthyear < 1950"/>
    </Distributions>
  </DataRange>
</Slot>
```

The statements above state that the menopausal status is defined in terms of the condition of the variable '\$birthyear'. The Boolean operator 'AND' is introduced to specify the composite expressions with the comparison operators, such as '<' (less than), '>' (greater than), '>=' (greater than or equal), etc.

⁶ For the normal distribution, it needs the two additional parameters, the mean μ and the standard deviation σ , i.e., $normal(\mu, \sigma)$. If those two parameters are omitted, it means that they take the default values, i.e., $\mu = (min + max)/2$, $\sigma = 0.5$.

However, we may need some variables which do not necessarily correspond with any slot. Thus, we design a pure variable slot which are used to generate some internal information without binding its values to any slot. Those dummy slots are defined by using the tab variable, like this:

```
<Variable value="houzenumber">
  <Distributions type="string" variable="$houzenumber">
    <Distribution disttype="uniform"
      datatype="minmax(int)" data="1,1000"/>
  </Distributions>
</Variable>
```

Evaluation slots are used to define the slots those values are calculated by some built-in predicates in expressions. For example, we use the predicate "concat" to denote the concatenate of strings, like this:

```
<EvaluationSlot value="phonenummer" type="string"
source="concat($nationalcode,-,$areacode,-,$localnumber)"/>
```

We use the predicate "eval" to denote the evaluation of arithmetic expressions, like this:

```
<EvaluationSlot value="lymphocytepc" type="float"
source="eval(100*$lymphocyte/$leukocyte)"/>
```

which means that the percentage of lymphocytes is calculated by the lymphocyte count and the leukocyte count.

3.6 Semantic Interoperability

We use the tab 'ConceptMapping' to map the PDDL entities into their corresponding concepts in the ontologies[9]. For example, the following statement states that the slot 'gender' has the concept in the ontology SNOMED with a concept id '263495000'.

```
<Slot value="Gender">
  <ConceptMapping ontology="snomed" conceptid="263495000"/>
  <DataRange>
    <enumeration value="female"/>
    <enumeration value="male"/>
  </Distributions></DataRange></Slot>
```

In this section we showed how the knowledge is formalized in the Patient Data Definition Language (PDDL) for the patient data generation. In the next section we report on the proposed approach in our Advanced Patient Data Generator (APDG).

4 Implementation

APDG is designed to support different kinds of formats for the generated patient data to make them easy to be accommodated into various EHR systems. Since we have defined the Patient Data Definition Language (PDDL) in XML, it would be convenient to use XSLT to transform XML-based patient data into the required data formats. The architecture of APDG is shown in Figure 1.

In order to control the generation of patient data, users can input some generation parameters into the system. Those generation parameters include

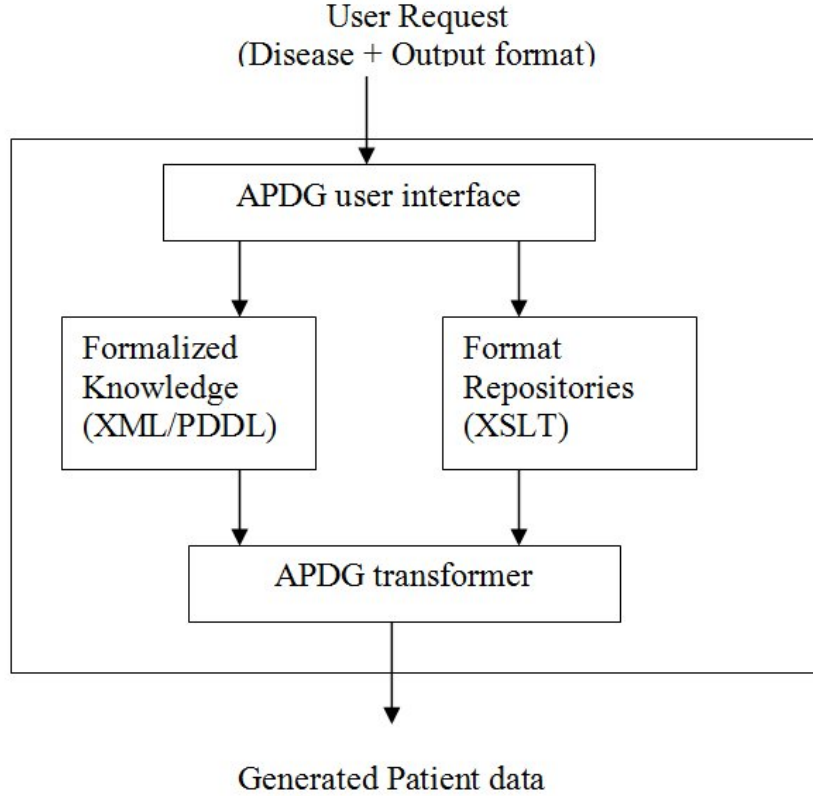


Fig. 1. The architecture of APGD.

- Number of patients. Namely, how many patients will be covered by the generated data. Usually we create a single file for each patient. With the support of the extended APDG system (i.e., the APDG system supports for the extension of the patient data based on the existing data), a single patient may have multiple data files which cover different sessions.
- Identification number of Patient (Patient ID). We want to design unique patient IDs for generated patient data, so that generated patient data can be loaded into any data store (typically, triple stores) without having to worry about ID conflict with other data. The Patient IDs consist of the following parts: the creator ID, which is used to identify the creator of the patient data, the session ID, which is used to identify different patient data sets which are generated by same creators, the disease ID, an additional ID which is designed to identify certain kinds of patients, and a patient number, which is used to identify a single patient which is created in the same session. The initial patient number (like 1000000) is used to create those patient numbers accordingly.

- Patient data format. The patient data format is used to specify what kinds of patient data format will be generated. We have provided the support for several formats of RDF data, which include the NTriple data format (with the extension names ‘nt’ or ‘ntriples’) and RDF/XML data format (with the extension name ‘xml’)⁷. Each data format corresponds to an XSLT file for the data generation.

Those generation parameters are set by editing a text file named ‘apdg.properties’, or inputting into the GUI interface of the APDG tool[11], before launching the APDG system. After the APDG system is launched, the system will load the formalized patient generation knowledge (i.e., those are formalized in PDDL) and the XSLT file which corresponds with the selected data format. The APDG transformer is a java system which calls the XSLT converter with the support of some Java libraries to interpret the patient generation knowledge which are encoded in PDDL.

5 Experiments

We have used APDG to perform several experiments on the patient data generation. In this section, we report the case of data generation for female breast cancer patients. In this case, we generated 10,000 female patients with the first diagnosis of breast cancers, and use those generated patient data for the experiments in SemanticCT, a semantically enabled system for clinical trials[7,6].

SemanticCT⁸ [7] provides semantic integration of various data in clinical trials. The system is semantically enabled for decision support in various scenarios in medical applications. SemanticCT has been semantically integrated with various data, which include trial documents with semantically annotated eligibility criteria and large amount of patient data with structured EHR and clinical medical records. Well-known medical terminologies and ontologies, such as SNOMED, LOINC, etc., have been used for the semantic interoperability.

SemanticCT is built on the top of LarKC (Large Knowledge Collider), a platform for scalable semantic data processing⁹[5,10]. With the built-in reasoning support for large-scale RDF/OWL data of LarKC, SemanticCT is able to provide various reasoning and data processing services for clinical trials, which include faster identification of eligible patients for recruitment service and efficient identification of eligible trials for patients.

The 10,000 generated breast cancer patients have been used in the tests of SemanticCT for automatic patient recruitment and trial finding. The generated patient data covers the main properties of clinical trials for female patients with the first diagnosis of breast cancer. Those main properties include:

- Gender. Since we want to create female patient data, we set the gender to ‘female’ with 100 percent in the PDDL and map the concept to SNOMED CT as follows:

⁷ We thank José Alberto Maldonado for creating the support of RDF/XML format

⁸ <http://wasp.cs.vu.nl/sct>

⁹ <http://www.larkc.eu>

The screenshot shows the SemanticCT web application interface. At the top, there's a navigation bar with 'VU UNIVERSITY AMSTERDAM' and 'SemanticCT' logos. Below this, a search bar contains 'NCT0155499' and 'SNOMED'. The patient information section shows 'CategoryID: BC_ZSH2012A' and 'PatientNo: 1000000'. The title of the trial is 'Neoadjuvant Treatment With Nab-paclitaxel For Patients With Stage II and III Luminal Breast Cancer'. The recruitment status is 'Not yet recruiting'. The main content area is divided into two sections: 'Patient Data' and 'Find the CTs for this patient'. The 'Patient Data' section contains a table with patient details.

Property	Value
Gender	Female
BirthYear	1941
Menopausal Status	postmenopausal
Currently Pregnant	no
Currently Nursing	no
Histopathology	Ductal carcinoma in situ
Diagnosis Year	1996
Diagnosis Month	September
Receptors Status	
Estrogen Receptor (ER)	positive
Progesterone Receptor (PR)	negative
HER2	positive
TNM Stage	
Stage	0
Tumor Size	1.1cm
Lymph Nodes	N0
Distant Metastases	M0

Fig. 2. The Patient Data in SemanticCT.

```

<Slot value="Gender">
  <ConceptMapping ontology="snomed" conceptid="263495000"/>
  <DataRange>
    <enumeration value="female"/>
    <enumeration value="male"/>
    <Distributions type="enumeration">
      <Distribution item="female" pfrom="0" pto="100"/>
      <Distribution item="male" pfrom="0" pto="0"/>
    </Distributions>
  </DataRange>
</Slot>

```

- Age. The age is an important variable which will be used to define the menopausal status and distribution of other properties. We collect the age distribution of female breast cancer from a cancer research website ¹⁰ and define this knowledge in PDDL as follows:

```

<Slot value="BirthYear">
  <ConceptMapping ontology="snomed" conceptid="397669002"/>
  <DataRange>
    <maxInclusive datatype="http://www.w3.org/2001/XMLSchema#date">
      1982</maxInclusive>
    <minInclusive datatype="http://www.w3.org/2001/XMLSchema#date">
      1900</minInclusive>
    <Distributions type="year" variable="$birthyear">
      <Distribution from="1973" to="1982" pfrom="0" pto="4.36"/>
      <Distribution from="1963" to="1972" pfrom="4.36"

```

¹⁰ <http://info.cancerresearchuk.org/cancerstats/types/breast/incidence/uk-breast-cancer-incidence-statistics>

```

    pto="19.32"/>
    <Distribution from="1953" to="1962" pfrom="19.32"
    pto="41.29"/>
    <Distribution from="1943" to="1952" pfrom="41.29"
    pto="67.12"/>
    <Distribution from="1933" to="1942" pfrom="67.12"
    pto="84.35"/>
    <Distribution from="1900" to="1932" pfrom="84.35"
    pto="100"/>
  </Distributions></DataRange></Slot>

```

- Menopausal. We define the menopausal status, based on the age (i.e., birth year) as follows:

```

<Slot value="MenopausalStatus">
  <ConceptMapping ontology="snomed" conceptid="161712005"/>
  <DataRange>
    <Distributions type="enumeration"
    variable="$menopausalstatus">
      <Distribution item="premenopausal" pfrom="0"
      pto="100" condition="$birthyear >1970"/>
      <Distribution item="perimenopausal" pfrom="0" pto="80"
      condition="$birthyear =<1970 AND $birthyear >=1950"/>
      <Distribution item="postmenopausal" pfrom="80" pto="100"
      condition="$birthyear =<1970 AND $birthyear >= 1950"/>
      <Distribution item="postmenopausal" pfrom="0" pto="100"
      condition="$birthyear <1950"/>
    </Distributions></DataRange></Slot>

```

- Histopathological diagnosis. The corresponding knowledge is collected from a Wikipedia page¹¹.

```

<Slot value="Histopathology">
  <DataRange>
    <enumeration value="Invasive ductal carcinoma">
      <ConceptMapping ontology="snomed" conceptid="408643008"/>
    </enumeration>
    <enumeration value="Ductal carcinoma in situ">
      <ConceptMapping ontology="snomed" conceptid="399935008"/>
    </enumeration>
    <enumeration value="Invasive lobular carcinoma">
      <ConceptMapping ontology="snomed" conceptid="444057000"/>
    </enumeration>
    <Distributions type="enumeration" variable="$diagnosis">
      <Distribution item="Invasive ductal carcinoma"
      pfrom="0" pto="55"/>
      <Distribution item="Ductal carcinoma in situ"
      pfrom="55" pto="68"/>
      <Distribution item="Invasive lobular carcinoma"
      pfrom="68" pto="73"/>
      <Distribution item="Lobular carcinoma in situ"
      pfrom="73" pto="100"/>
    </Distributions></DataRange></Slot>

```

Figure 2 shows a screen shot of the patient data in SemanticCT. For the experiment of patient recruitment service. We have picked up 10 clinical trials randomly and formalized their eligibility criteria by using the rule-based formalization [6]. We have tested the system for automatically identifying eligible patients for those selected trials. For the experiment of trial finding service, we use SPARQL queries with regular expressions over eligibility criteria to find the trials which are suitable for the patients. The results show that the generated patient data are useful for various tests in the system[7,6].

¹¹ http://en.wikipedia.org/wiki/Breast_cancer_classification

6 Discussion and Conclusion

6.1 Related Work

Clinical avatars¹² developed by the Laboratory for Personalized Medicine are virtual representations of people for the purpose of conducting personalized medicine simulations. Similar with APDG, Clinical Avatars are configured so that their statistical distribution matches the requirements of a particular population. The Conditional Probability Table which describes the distribution of the avatar attributes are used to configure Clinical Avatars.

In [4], an approach of synthetic patient data is developed. It consists of an OWL schema that describes the data needed to calculate the exemplary indicators (TBox, i.e. terminological background knowledge), and the patient data (ABox, i.e. knowledge about individuals). It can generate both the OWL schema and the patient data in OWL 2 with the OWL API. [4] uses unique random distribution only for the generation of patient data.

[8,3] propose a data-driven approach for creating synthetic electronic medical records. The approach consists of the three major steps: 1) synthetic patient identity and basic information generation; 2) identification of care patterns that the synthetic patients would receive based on the information present in real EMR data for similar health problems; 3) adaptation of these care patterns to the synthetic patient population. A distance measure is used to identify the closest patient care descriptor to the desired inject.

APDG supports for comprehensive configuration of domain-knowledge and description of statistical distribution and variable dependence. Thus, it provide a more powerful tool to generate various patient data which meets different requirements. Furthermore, in APDG, the Patient Data Definition Language PDDL is designed based on the XML format. Thus, it is much more convenient and flexible to convert the PDDL data into ones with other kinds of data formats.

6.2 Concluding Remarks

We have proposed a knowledge-based approach of synthesizing large scale patient data. This synthesis is to use various domain knowledge we can collect to control the patient data generation. Those domain knowledge can be collected from biomedical publications or web resources. The collected knowledge are formalized in an XML-based language PDDL to describe required patient data and their distributions.

There are many interesting issues for future work of APDG. The existing APDG supports for the generation of the RDF data formats (RDF/NTriples and RDF/XML) only. We are going to create more XSLT files to make APDG to cover wider range of data formats. We are going to extend the PDDL so that it can cover wider range of data distribution declaration and more powerful expressions for variable dependence description.

¹² <http://clinicalavatars.org/>

Acknowledgments This work is partially supported by the European Commission under the 7th framework programme EURECA Project (FP7-ICT-2011-7, Grant 288048). We thank José Alberto Maldonado of the Universidad Politécnica de Valencia Spain, who contributes to the XSLT file for the generation of RDF/XML in APDG. Thanks to Minghui Zhang and the team in Wuhan University of Science and Technology China, who contribute to the design and the implementation of the visual interface tools for APDG.

References

1. T. Beale. Archetypes: Constraint-based domain models for future-proof information systems. In *OOPSLA 2002 workshop on behavioural semantics*, 2002.
2. A. Bucur, A. ten Teije, F. van Harmelen, G. Tagni, H. Kondylakis, J. van Leeuwen, K. D. Schepper, and Z. Huang. Formalization of eligibility conditions of CT and a patient recruitment method, D6.1. Technical report, EURECA Project, 2012.
3. A. Buczak, S. Babin, and L. Moniz. Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making*, 10(59), 2010.
4. K. Dentler, A. ten Teije, R. Cornet, and N. de Keizer. Towards the automated calculation of clinical quality indicators. In *Proceedings of AIME 2011 Workshop KR4HC (Knowledge Representation for Health-Care)*, 2011.
5. D. Fensel, F. van Harmelen, B. Andersson, P. Brennan, H. Cunningham, E. Della Valle, F. Fischer, Z. Huang, A. Kiryakov, T. Lee, L. School, V. Tresp, S. Wesner, M. Witbrock, and N. Zhong. Towards LarKC: a platform for web-scale reasoning. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2008)*. IEEE Computer Society Press, CA, USA, 2008.
6. Z. Huang, A. den Teije, and F. van Harmelen. Rule-based formalization of eligibility criteria for clinical trials. In *Proceedings of the 14th Conference on Artificial Intelligence in Medicine(AIME 2013)*, 2013.
7. Z. Huang, A. ten Teije, and F. van Harmelen. SemanticCT: A semantically enabled clinical trial system. In R. Lenz, S. Mikszh, M. Peleg, M. Reichert, and D. R. A. ten Teije, editors, *Process Support and Knowledge Representation in Health Care*. Springer LNAI, 2013.
8. L. Moniz, A. L. Buczak, L. Hung, S. Babin, M. Dorko, and J. Lombardo. Construction and validation of synthetic electronic medical records. *Journal of Public Health*, 1(1):1–36, 2009.
9. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with snomed-rt. In *Journal of the American Medical Informatics Association*, 2000.
10. M. Witbrock, B. Fortuna, L. Bradesko, M. Kerrigan, B. Bishop, F. van Harmelen, A. ten Teije, E. Oren, V. Momtchev, A. Tenschert, A. Cheptsov, S. Roller, and G. Gallizo. D5.3.1 - requirements analysis and report on lessons learned during prototyping. Larkc project deliverable, June 2009.
11. M. Zhang, Z. Huang, and J. Gu. Visual interface tools for advanced patient data generator. *Chinese Digital Medicine*, (to appear), 2013.