# Junhao Du

**Observation:**

Although some basic information doesn’t appear in the ppt, after searching for the technology of the Xquery and Xpath. I have got something more like the following:

(1)The “book//name” operator can find all the element name under the book element wherever the name is.

(2)We can use the operator “/..” to return to the parent element

(3)We can use the **item**()* as the return type of the function. In this way, we can output the xml without the “xmlns” head.

(4)In a function there should not exist more than 1 return to output the xml, so we just need to use the “typeswitch” to return some information in a definite situation.

**Video Position**:/record.mp4

**Test Program Position:**/workspace/*

**Test Data:** **ClinicData.xml**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<clinic:Clinic xmlns:clinic="http://www.example.org/clinic"
xmlns:pat="http://www.example.org/schemas/clinic/patient"
xmlns:prov="http://www.example.org/schemas/clinic/provider"
```

```
xmlns:trmt="http://www.example.org/schemas/clinic/Treatment"
xmlns:tns="http://www.example.org/schemas/clinic/clinic_Ids"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.example.org/Clinic
file:/E:/universityGraduate/CS548EnterpriseSoftwareSystem/Assignment/Assignment
3/workspace/Clinic.xsd">

    <pat:patients>
        <pat:PatientId>12345678</pat:PatientId>
        <pat:name>jane</pat:name>
        <pat:dob>1995-06-08</pat:dob>
        <pat:treatments>
            <pat:treatment>
                <trmt:ProviderId>87654321</trmt:ProviderId>
                <trmt:diagnosis>migraine</trmt:diagnosis>
                <trmt:Surgery>
                    <trmt:surgeon>Eason</trmt:surgeon>
                    <trmt:data>2017-09-12</trmt:data>
                </trmt:Surgery>
            </pat:treatment>
            <pat:treatment>
                <trmt:ProviderId>76543218</trmt:ProviderId>
                <trmt:diagnosis>Asthma</trmt:diagnosis>
                <trmt:DrugTreatment>
                    <trmt:name>panadol</trmt:name>
                    <trmt:dosage>50.0</trmt:dosage>
                    <trmt:physician>Albert</trmt:physician>
                </trmt:DrugTreatment>
            </pat:treatment>
        </pat:treatments>
    </pat:patients>

    <pat:patients>
        <pat:PatientId>23456781</pat:PatientId>
        <pat:name>jhon</pat:name>
        <pat:dob>1995-08-08</pat:dob>
        <pat:treatments>
            <pat:treatment>
                <trmt:ProviderId>76543218</trmt:ProviderId>
                <trmt:diagnosis>hiv/aids</trmt:diagnosis>
                <trmt:DrugTreatment>
                    <trmt:name>replax</trmt:name>
                    <trmt:dosage>35.0</trmt:dosage>
                    <trmt:physician>Eason</trmt:physician>
```

```xml
            </trmt:DrugTreatment>
        </pat:treatment>
        <pat:treatment>
            <trmt:ProviderId>87654321</trmt:ProviderId>
            <trmt:diagnosis>hurt</trmt:diagnosis>
            <trmt:Surgery>
                <trmt:surgeon>Irving</trmt:surgeon>
                <trmt:data>2000-03-08</trmt:data>
            </trmt:Surgery>
        </pat:treatment>
        <pat:treatment>
            <trmt:ProviderId>76543218</trmt:ProviderId>
            <trmt:diagnosis>cancer</trmt:diagnosis>
            <trmt:DrugTreatment>
                <trmt:name>panacea</trmt:name>
                <trmt:dosage>100.0</trmt:dosage>
                <trmt:physician>Eason</trmt:physician>
            </trmt:DrugTreatment>
        </pat:treatment>
        <pat:treatment>
            <trmt:ProviderId>65432187</trmt:ProviderId>
            <trmt:diagnosis>cancer</trmt:diagnosis>
            <trmt:Radiology>
                <trmt:data>2017-09-20</trmt:data>
                <trmt:radiologist>Albert</trmt:radiologist>
            </trmt:Radiology>
        </pat:treatment>
    </pat:treatments>
</pat:patients>

<prov:providers>
    <prov:ProviderId>87654321</prov:ProviderId>
    <prov:name>Irving</prov:name>
    <prov:specialization>Surgery</prov:specialization>
</prov:providers>

<prov:providers>
    <prov:ProviderId>76543218</prov:ProviderId>
    <prov:name>Eason</prov:name>
    <prov:specialization>Oncology</prov:specialization>
</prov:providers>

<prov:providers>
    <prov:ProviderId>65432187</prov:ProviderId>
```

```
                <prov:name>Albert</prov:name>
                <prov:specialization>Radiology</prov:specialization>
        </prov:providers>
</clinic:Clinic>
```

## Xquery Code:                    testing1.xqm

```
import schema namespace p="http://www.example.org/schemas/clinic/patient" at
"Patient.xsd";
import schema namespace t="http://www.example.org/schemas/clinic/Treatment"
at "Treament.xsd";
import schema namespace c="http://www.example.org/clinic" at "Clinic.xsd";
declare namespace ps="http://www.example.org/schemas/clinic/patients";

declare    function local:getPatientTreatments ($pid as xs:integer,$database as
element(c:Clinic)) as item ()*
{
for $treatment in $database/p:patients
where $treatment/p:PatientId=$pid
return
<ps:patients>
<ps:pid>{$treatment/p:PatientId/text()}</ps:pid>
<ps:treatments>{$treatment/p:treatments}</ps:treatments>
</ps:patients>
};
let $clinic :=doc("ClinicData.xml")/c:Clinic
for $patient in doc("ClinicData.xml")/c:Clinic/p:patients
where $patient/p:PatientId/text()=23456781

  return
<ps:patients>{local:getPatientTreatments($patient/p:PatientId,$clinic)}</ps:patients>
```

## Xquery Code:                    testing2.xqm

```
(:module namespace gpd='http://www.example.org/xquery/clinic/getProviderInfo' :)

import schema namespace p="http://www.example.org/schemas/clinic/patient" at
"Patient.xsd";
import schema namespace t="http://www.example.org/schemas/clinic/Treatment"
at "Treament.xsd";
import schema namespace c="http://www.example.org/clinic" at "Clinic.xsd";
declare namespace d="http://www.example.org/schemas/clinic/drug";
declare function local:getPatientDrugs($patient-id as xs:integer,
$database as element(c:Clinic))
as element(d:drug)*
{
	for $patient in $database/p:patients
	where $patient/p:PatientId = $patient-id
	for $treatment in $patient/p:treatments/p:treatment
	where $treatment//t:DrugTreatment
	return
		<d:drug>
			<d:name>{$treatment/t:DrugTreatment/t:name/text()}</d:name>

<d:dosage>{$treatment/t:DrugTreatment/t:dosage/text()}</d:dosage>
			<d:diagnosis>{$treatment/t:diagnosis/text()}</d:diagnosis>
		</d:drug>
};
let $clinic :=doc("ClinicData.xml")/c:Clinic
for $patient in doc("ClinicData.xml")/c:Clinic/p:patients
where $patient/p:PatientId/text()=23456781
  return <d:drug>{local:getPatientDrugs($patient/p:PatientId,$clinic)}</d:drug>
```

## Xquery Code:                    testing3.xqm

```
import schema namespace p="http://www.example.org/schemas/clinic/patient" at
"Patient.xsd";
import schema namespace t="http://www.example.org/schemas/clinic/Treatment"
at "Treament.xsd";
import schema namespace c="http://www.example.org/clinic" at "Clinic.xsd";
declare namespace ti="http://www.example.org/schemas/clinic/treatmentInfo";
declare function local:getTreatmentInfo($database as element(c:Clinic))
as item()*
{
	for $patient in $database/p:patients

	for $treatment in $patient/p:treatments/p:treatment/node()
	return
	typeswitch($treatment)
	case $Drug as element(t:DrugTreatment)
	return<ti:treatmentInfo>
			<ti:name>DrugTreatment</ti:name>
			<ti:pid>{$patient/p:PatientId/text()}</ti:pid>
			<ti:ProviderId>{$treatment/../t:ProviderId/text()}</ti:ProviderId>
```

```
        </ti:treatmentInfo>

    case $Surgery as element(t:Surgery)
    return<ti:treatmentInfo>
            <ti:name>Surgery</ti:name>
            <ti:pid>{$patient/p:PatientId/text()}</ti:pid>
            <ti:ProviderId>{$treatment/../t:ProviderId/text()}</ti:ProviderId>
        </ti:treatmentInfo>
    case $Radiology as element(t:Radiology)
    return<ti:treatmentInfo>
            <ti:name>Radiology</ti:name>
            <ti:pid>{$patient/p:PatientId/text()}</ti:pid>
            <ti:ProviderId>{$treatment/../t:ProviderId/text()}</ti:ProviderId>
        </ti:treatmentInfo>
    default
    return ""
};

let $clinic :=doc("ClinicData.xml")/c:Clinic
  return <t:drug>{local:getTreatmentInfo($clinic)}</t:drug>
```

## Xquery Code: testing4.xqm

```
import schema namespace p="http://www.example.org/schemas/clinic/patient" at
"Patient.xsd";
import schema namespace t="http://www.example.org/schemas/clinic/Treatment"
at "Treament.xsd";
import schema namespace pr="http://www.example.org/schemas/clinic/provider" at
"Provider.xsd";
import schema namespace c="http://www.example.org/clinic" at "Clinic.xsd";
declare namespace ph="http://www.example.org/schemas/clinic/providerHistory";
declare function local:getProviderInfo($database as element(c:Clinic))
as item()*
{
    for $provider in $database/pr:providers
    where $provider/pr:ProviderId
    for $treatment in $database/p:patients/p:treatments/p:treatment/node()
    where $treatment/../t:ProviderId=$provider/pr:ProviderId
    return typeswitch($treatment)
    case $Drug as element(t:DrugTreatment)
    return
    <ph:providerHistory>
```

```
        <ph:providerId>{$provider/pr:ProviderId/text()}</ph:providerId>
        <ph:providerName>{$provider/pr:name/text()}</ph:providerName>
        <ph:pid>{$treatment/../../../p:PatientId/text()}</ph:pid>
        <ph:treatment>drugTreatment</ph:treatment>
        </ph:providerHistory>
        case $Surgery as element(t:Surgery)
        return
        <ph:providerHistory>
        <ph:providerId>{$provider/pr:ProviderId/text()}</ph:providerId>
        <ph:providerName>{$provider/pr:name/text()}</ph:providerName>
        <ph:pid>{$treatment/../../../p:PatientId/text()}</ph:pid>
        <ph:treatment>surgery</ph:treatment>
        </ph:providerHistory>
        case $Radiology as element(t:Radiology )
        return
        <ph:providerHistory>
        <ph:providerId>{$provider/pr:ProviderId/text()}</ph:providerId>
        <ph:providerName>{$provider/pr:name/text()}</ph:providerName>
        <ph:pid>{$treatment/../../../p:PatientId/text()}</ph:pid>
        <ph:treatment>radiology</ph:treatment>
        </ph:providerHistory>
        default
        return ""
};

let $clinic :=doc("ClinicData.xml")/c:Clinic
  return <ph:drug>{local:getProviderInfo($clinic)}</ph:drug>
```

## Xquery Code: testing5.xqm

```
import schema namespace p="http://www.example.org/schemas/clinic/patient" at
"Patient.xsd";
import schema namespace t="http://www.example.org/schemas/clinic/Treatment"
at "Treament.xsd";
import schema namespace c="http://www.example.org/clinic" at "Clinic.xsd";
declare namespace di="http://www.example.org/schemas/clinic/drugInfo";
declare function local:getDrugInfo($database as element(c:Clinic))
as item()*
{
    for $patient in $database/p:patients
    for $treatment in $patient/p:treatments/p:treatment
    where $treatment//t:DrugTreatment
    return
        <di:drugInfo>
            <di:patients>{$treatment/../../p:PatientId/text()}</di:patients>
            <di:name>{$treatment/t:DrugTreatment/t:name/text()}</di:name>
            <di:diagnosis>{$treatment/t:diagnosis/text()}</di:diagnosis>

<di:dosage>{$treatment/t:DrugTreatment/t:dosage/text()}</di:dosage>

<di:physician>{$treatment/t:DrugTreatment/t:physician/text()}</di:physician>
        </di:drugInfo>
};

let $clinic :=doc("ClinicData.xml")/c:Clinic
  return <di:drug>{local:getDrugInfo($clinic)}</di:drug>
```