

Milestone 2 of CS2310

Lin Li, lil53@pitt.edu

April 2, 2008

1 Example of Brief Steps to Define Index Cells in IC Builder

- Create icname.gra file
- Define Index Cell
 - Add states
 - Define states: id and name
 - Add transition
 - Define transition: input messages, actions, output messages
- Create other icname.gra file
- Export icname.in and .dat file

This part will be elaborated in final project report.

2 Wrapper to Generate the XML file

2.1 The Format of .in File

The .in file is generated by IC Builder, and formally describes the Index Cell.

The .in file formalize all the transitions that the Index Cell iterates, which is the only content in the .in file. For each of the transition, the following information is provided:

```
<digit>// number notation of current state
<digit>// number notation of next state
<digit>// number of input message(s)
Message.1 // Input message 1 definition
....
Message.n // Input message n definition
<digit>// 1: predicate; 0: no predicate
<digit>// number of output ic(s) to which the output message is sent to
<digit>// number of output message(s)
Message.1 // Output message 1 definition
....
```

```

Message.n // Output message n definition
<digit>// number of action(s)
<digit>, para // number notation of actions and their parameters
....
<digit>, para // number notation of actions and their parameters

```

The format of message definition is as follows:

```

<para_list> ::= <para_list>' '<item>
<para_list> ::= <item>
<item> ::= <const> | <var> | <func>
<const> ::= I<integer> | F<float> | S<string>
<var> \::= X<digit> | Y<digit> | Z<digit>
<func> ::= G<digit>'('<func_para_list>')' | H<digit>'('<func_para_list>')'
<func_para_list> ::= <f_para_list> | NULL
<f_para_list> ::= <f_para_list>', '<item>
<f_para_list> ::= <item>
<digit> ::= '0'..'9'

```

For example, this is one sample of generated .in file, which describes the Proficiency Level Index Cell in Exercise 2. The state transition diagram is shown in Figure 1.

```

0 // current state
1 // next state
1 // 1 input message(s)
0:0.60 // Access
0 // no. predicate
0 // 0 output ic(s)
0 // 0 output message(s)
1 // 1 action(s)
0,0 // Increase_by_1 increase.c
1 // current state
2 // next state
1 // 1 input message(s)
0:0.60 // Access
0 // no. predicate
0 // 0 output ic(s)
0 // 0 output message(s)
1 // 1 action(s)
0,0 // Increase_by_1 increase.c
2 // current state
3 // next state
1 // 1 input message(s)
0:0.60,0 // Access
0 // no. predicate
0 // 0 output ic(s)
0 // 0 output message(s)
1 // 1 action(s)

```

```

0,0 // Increase_by_1 increase.c
3 // current state
3 // next state
1 // 1 input message(s)
0:0.60,1 // Access
0 // no. predicate
1 // 1 output ic(s)
-1
1 // 1 output message(s)
1:0 // Proficiency_Level_Needed
1 // 1 action(s)
1,1 // Send_Message sendmessage.c

```

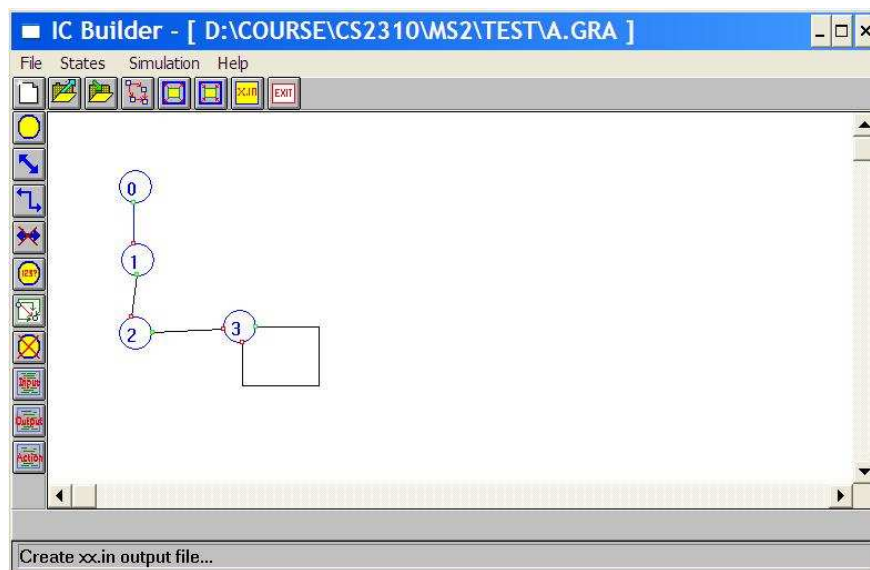


Figure 1: Sample State Transition Diagram of IC

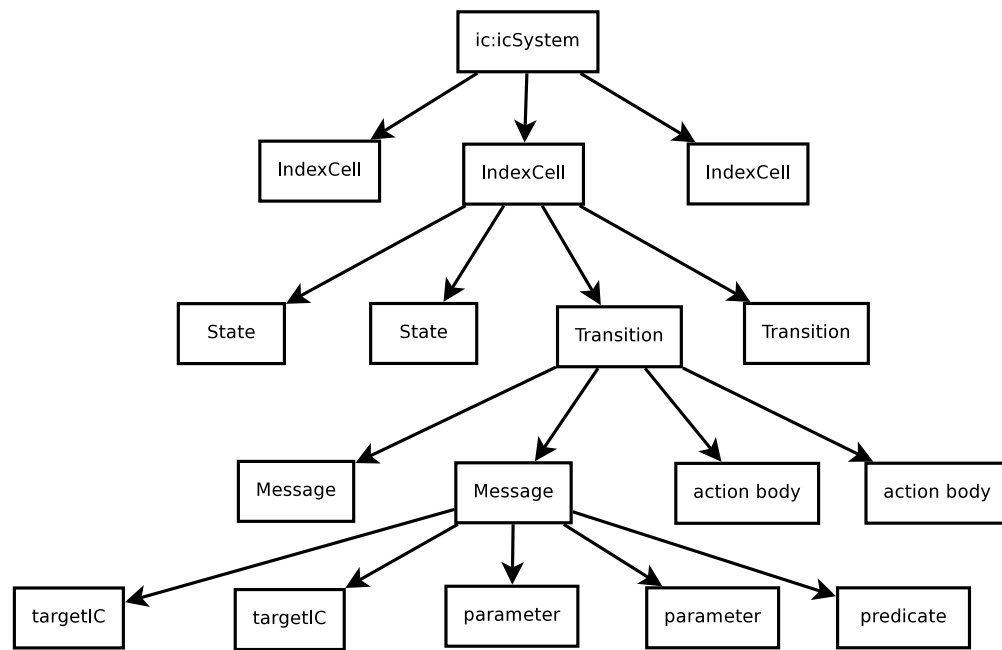
2.2 XML Specification for IC System

XML documents, with its inherited tree structure, could be obviously mapped into an AST in typical compiler. The tree structure of the IC System is shown in Figure 2.

2.2.1 XML declaration:

XML declaration is optionally preceded the root element to state the version of XML and the character encoding information.

Example:



Tree Structure of IC System in XML

Figure 2: Tree Structure of IC System

```
<?xml version="1.0" encoding="UTF-8"?>
```

2.2.2 Root element

Root element in IC System is the entity of `ic:icSystem`, which is composed of several Index Cells as its Children Elements.

Example:

```
<ic:icSystem
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns:ic="http://www.mke.org/ic/">
Children Elements
</ic:icSystem>
```

2.2.3 Index Cell

Index Cell is composed of multiple states and transitions.

Property of Index Cell include:

- id: the identity string

- maxLifeTime: expiration time (in ms) of the Index Cell
- name: name of the instance of the Index Cell type

Example:

```
<indexCell id="ic1" maxLifeTime="inf"
name="Sensor">
  Children Elements
</indexCell>
```

2.2.4 State

The Index Cell has several states. Under each state, the Index Cell accepts messages, and the transition is triggered, the Index Cell would enter other states.

Property of State include:

- id: string to identify the state
- type: entering (unique), internal(multiple), and ending(multiple).
- name: name of the state

Example:

```
<state id="state3" type="entering" name="listen"/>
<state id="state4" type="internal" name="action"/>
```

2.2.5 Transition

The Transitions are hierarchical objects composed of the input message, actions, and output messages. Transitions connect from the source state to the target state. Transitions are triggered by input messages, then perform correspond actions, and pass output messages to other Index Cells.

Property of Transitions include:

- id: string to identify the transition
- source: the id of source state
- target: the id of target state
- type: There are two types of Transitions: Internal transitions happen in the same Index Cell, and boundary transitions happen in different Index Cell.

Examples:

```
<transition id="trans2" source="state3"
target="state4" type="internal">
Children Elements
\<transition>
```

2.2.6 Action Body

Action body defines the action(s) taken by the Index Cell.
The property of Action Body include:

- id: string that identify the action body
- name: name of the action
- target: external or internal

Example:

```
<action body="alert Nurse" id="act1"  
name="emergency" target="external"/>
```

2.2.7 Message

Messages are composed of parameters, target IC and predicate. Messages pass information between multiple Index Cells.

Property of Message include:

- id: string that identify the message
- type: input or output message
- name: name of the message

Example:

```
<message xsi:type="ic:InputMessage" id="msg1"  
name="getPatientCondition">  
</message>
```

Property of parameter include dataType, dataValue and name.

Example:

```
<parameter dataType="double" dataValue="98.7"
```

Property of targetIC only include the id of target IC.

Example:

```
<targetIC>ic3</targetIC>
```

Property of predicate include the contents of the predicate

Example:

```
<predicate>condition< threshold</predicate>
```

Example of Specification of Index Cell in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<ic:icSystem
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns:ic="http://www.mke.org/ic/">
  <indexCell id="ic1" maxLifeTime="inf"
  name="Sensor">
    <state id="state0" type="entering" name="send"/>
    <transition id="trans0" source="state0"
target="state0" type="internal">
      <message xsi:type="ic:OutputMessage" id="msg0"
name="sendPatientCondtion">
        <targetIC>ic2</targetIC>
        <parameter dataType="double" dataValue="98.6"
name="condition"/>
      </message>
    </transition>
  </indexCell>
  <indexCell currentState="state1" id="ic2"
  maxLifeTime="inf" name="Sensor Emergency Alert">
    <state id="state1" type="entering" name="listen"/>
    <state id="state2" type="internal" name="alert"/>
    <transition id="trans1" source="state1"
target="state2" type="internal">
      <message xsi:type="ic:OutputMessage" id="msg2"
name="threshold reached">
        <targetIC>ic3</targetIC>
        <parameter dataType="time"
dataValue="getCurrentTime()" name="Tc"/>
      </message>
      <message xsi:type="ic:InputMessage" id="msg1"
name="getPatientCondition">
        <predicate>condition< &lt; threshold</predicate>
        <parameter dataType="double" dataValue="98.7"
name="condition"/>
      </message>
      <action body="alert Nurse" id="act1"
name="emergency" target="external"/>
    </transition>
  </indexCell>
  <indexCell id="ic3" maxLifeTime="inf"
  name="Nurse">
    <state id="state3" type="entering" name="listen"/>
    <state id="state4" type="internal" name="action"/>
    <transition id="trans2" source="state3"
target="state4" type="internal">
      <message xsi:type="ic:InputMessage" id="msg3"

```

```
name="getEmergencyRequest">
  <predicate>receive msg2</predicate>
  <parameter dataType="time"
  dataValue="getCurrentTime()" name="Tc"/>
</message>
  <action body="help patient" id="act2"
  name="dispatch" target="external"/>
</transition>
</indexCell>
</ic:icSystem>
```

2.3 Plan of Future Work

With the specification of .in file and XML file of IC System, a parser should be implemented to retrieve all the necessary information of the all the objects of the Index Cell. And the all the definition of the Index Cell objects should be formalized into XML format. Since the format of .in file is fixed, Perl script is enough to get the work done. This Perl script would parse the .in file which is generated by the IC Builder, and then generate the XML file.