[用户指南](http://www.opendial-toolkit.net/user-manual) >

对话域

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 本节详细介绍如何使用XML实际编码OpenDial的对话域。  **1.总体结构**  OpenDial中的对话域如下所示：  <domain>    <initialstate>     <!--(optional) initial state variables -->   </initialstate>    <parameters>     <!--(optional) prior distributions for rule parameters -->   </parameters>    <model trigger="trigger variables for model 1">     <!--probabilistic rules for model 1 -->   </model>    <model trigger="trigger variables for model 2">     <!-- probabilistic rules for model 2 -->   </model>    ...     <model trigger="trigger variables for model n">     <!-- probabilistic rules for model n -->   </model>     <settings>     <!--(optional) domain-specific settings -->   </settings>  </domain>  如果设置为空，则设置，初始状态和参数可以不在域规范中。规则结构模型的数量是任意的。  对于更复杂的域名，可以通过导入标记将域名规范拆分为多个文件：      <import href="path to another file" />  许多对话域的例子可以在基本目录的目录域和测试/域中找到。  **<domain>的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | <参数initialstate> | 元件 | 0-1 | 对话域的初始状态 | | <参数> | 元件 | 0-1 | 先前的参数分布 | | <import href =“...”/> | 元件 | 0-n的 | 导入其他XML文件 | | <model trigger =“...”> | 元件 | 0-n的 | 对话模式 | | <设置> | 元件 | 0-1 | 特定于域的系统设置 |   **2.初始状态**  该域的初始状态在启动对话系统时定义包括在对话状态中的变量。每个变量都有一个特定的标识符和概率分布。  具有离散值范围的变量定义为分类表：  <variable id="variable\_id">   <value prob="probability for first value">first value</value>   <value prob="probability for second value">second value</value>   ...   <value prob="probability for the nth value">nth value</value> </variable>  概率值必须介于0和1之间。如果总概率小于1，则OpenDial自动为剩余概率质量添加一个空值（无）。如果省略prob属性，则假定该值具有概率1。  这是一个状态变量的简单例子：  <variable id="userIntention">   <value prob="0.5">Want(Object\_A)</value>   <value prob="0.3">Want(Object\_B)</value> </variable>  也可以使用XML元素<distrib type =“...”>（请参见下文）为连续范围定义概率分布。  **用于<initialstate>的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | <variable id =“...”> | 元件 | 0-n的 | 状态变量 |   **为XML格式<变量>在<参数initialstate> ：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | ID | 属性 | 1 | 变量标签 | | <value prob =“p”> | 元件 | 1-n的 | 概率为p的变量的可能值。如果省略属性prob，则概率被假定为1。 | | ***或*** <distrib type =“...”> | 元件 | 0-1 | *比照 下面* |  |  | | --- | | **重要提示**： 一般来说，变量可以具有任意标识符，但应避免使用几个特殊字符。变量应该***不***包括质数（'），大括号（{，}）或方括号（[，]），因为这些在OpenDial内部使用。此外，以^ p，^ t和^ o结尾的变量有一个特殊的功能：^ p表示[预测变量](http://www.opendial-toolkit.net/user-manual/advanced-modelling-1#6._Predictive_variables)，^ t表示在每次更新循环后立即删除的临时变量，^ o表示[用户模拟器的](http://www.opendial-toolkit.net/user-manual/parameter-estimation#With_simulated_dialogues)观察变量。  一些变量值在OpenDial中也有特殊含义：“ 无 ”表示“空”值，方括号[ ]中的值表示元素集。 |     **3.参数**  概率规则可以包括其参数的值最初是未知的，并且必须根据数据进行估计。由于OpenDial采用贝叶斯学习方法，因此每个参数必须与其先前分布（通常是连续的）可能值范围相关联。  **<参数>的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | <variable id =“...”> | 元件 | 0-n的 | 状态变量 |   参数的定义与状态变量完全相同。它们的分布以参数化的方式定义：   * **均匀分布**用两个参数min和max定义。分布*U（-1,3）*因此被编码为：   <variable id="uniform\_example">   <distrib type="uniform">     <min>-1</min>     <max>3</max>   </distrib> </variable>   * **高斯分布**[[1]](http://www.opendial-toolkit.net/user-manual/dialogue-domains#%5B1%5D)用两个参数均值和方差来定义- 例如，*N（2,4）*被编码为：   <variable id="gaussian\_example">   <distrib type="gaussian">     <mean>2</mean>     <variance>4</variance>   </distrib> </variable>   * **狄利克雷分布**。狄利克雷分布是多变量连续分布。它通常用于描述分类/多项分布的先验参数分布。Dirichlet分布由*alpha*值列表（每个维度一个）定义。例如，三维分布*Dirichlet（1,1,2）*表示为：   <variable id="dirichlet\_example">  <distrib type="dirichlet">   <alpha>1</alpha>     <alpha>1</alpha>     <alpha>2</alpha>   </distrib> </variable>  **4.模型**  对话模型基本上定义为一组概率规则，与一个或多个“触发变量”相结合，定义何时应用规则：  <model trigger="trigger variable(s)">   <rule id="rule 1">       ...  </rule>   <rule id="rule 2">       ...  </rule>   ...   <rule id="rule n">       ...  </rule>  </model>  触发器变量必须用逗号分隔。这些规则可以编码概率或效用规则，正如我们下面所解释的。  **<model>的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | ID | 属性 | 0-1 | （可选）名称 | | 触发 | 属性 | 1 | 逗号分隔的触发器变量列表 | | <规则> | 元件 | 1-n的 | 概率或效用规则 |   *概率规则*  概率规则表示状态变量子集（规则的“输入变量”）如何影响其他一些状态变量（“输出变量”）的概率分布。输出变量可能已经存在于对话状态（在这种情况下，它们的内容被擦除），或者代表要包括在对话状态中的新变量。  概率规则的结构为**if ... then ... else**构造：  如果（条件c1 ）则  P （效应e1 ）= ...   P （效应e2 ）= ... ... 否则如果（条件c2 ）则... 否则...               在XML中，这些概率规则表示为案例 s的（有序）列表。每种情况都有一个（可能是空的）条件和一个备选效应列表（每个都有一个特定的概率）。  以下是概率规则的一个具体例子（对应于[Lison（2014）](http://folk.uio.no/plison/pdfs/thesis/thesis-plison2014.pdf)第65页规则r 1）：  <rule id="r1">   <case>     <condition>       <if var="Rain" value="false"/>       <if var="Weather" value="hot"/>     </condition>     <effect prob="0.03">       <set var="Fire" value="true"/>     </effect>     <effect prob="0.97">       <set var="Fire" value="false"/>     </effect>   </case>   <case>     <effect prob="0.01">       <set var="Fire" value="true"/>     </effect>     <effect prob="0.99">       <set var="Fire" value="false"/>     </effect>    </case> </rule>  规则r 1 只是表示如果没有降雨和炎热天气发生火灾的概率是0.03，而在其他情况下这个概率是0.01。  在某些情况下，人们可能希望在规则中强制执行特定的支配等级（以确保某些规则如果同时触发，则优先于其他规则）。这可以使用priority属性指定，取一个整数值（其中1表示最高优先级）。  **<rule>的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | ID | 属性 | 0-1 | （可选）规则的名称 | | 优先 | 属性 | 0-1 | （可选）整数表示规则的优先级（其中1最高） | | <壳体> | 元件 | 1-n的 | 规则案例列表 |   **<case>的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | <条件> | 元件 | 0-1 | 情况。如果省略，OpenDial将假定 一个空的（即平凡真实的）条件。 | | <效果> | 元件 | 1-n的 | 该案件的替代效果列表 |   我们现在详细说明条件和效果如何具体指定。  *条件*  如上述规则所示，条件 XML节点由一系列基本条件组成。  **<condition>的 XML格式：**[[2]](http://www.opendial-toolkit.net/user-manual/dialogue-domains#%5B2%5D)   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | 操作者 | 属性 | 0-1 | （可选）逻辑运算符。可能的值是“ 和 ”和“ 或 ”。默认值是“ 和 ”。 | | <if ...> | 元件 | 0-n的 | 基本条件。 |   每个基本条件都被写为具有三个基本属性的<if ... />标记：  **<if ... />的 XML格式：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | VAR | 属性 | 1 | 变量标签 | | 关系 | 属性 | 0-1 | （可选）满足的二元关系。默认关系是平等的。可接受的关系是：   * =（相等） * ！=（不等式） * ＆LT; （低于） * ＆GT; （比...更棒） * 包含（包含元素或子字符串） * ！包含（不包含元素或子字符串） * 在（包含在） * ！（不包含在） | | 值 | 属性 | 1 | 要检查的变量值 |   *效果*  每个案例都 包含一个或多个（替代）效果。每种效应都有特定的发生概率。这个概率可以用手来指定，如上例所示：      <effect prob="0.03">       <set var="Fire" value="true"/>     </effect>  当效应未指定任何概率 属性时，假定效应具有概率1.当所有效应的总概率低于1时，隐式假定空效应覆盖剩余概率质量。  特定效果的概率也可以是一个参数。在这种情况下，每个具有*n个*替代效应的情况与 表示影响概率的可能值的*第n*维Dirichlet分布相关联。例如，规则r 1中的效应概率可以被重写为：  <rule id="r1">   <case>     <condition>       <if var="Rain" value="false"/>       <if var="Weather" value="hot"/>     </condition>     <effect prob="firstdirichlet[0]">       <set var="Fire" value="true"/>     </effect>     <effect prob="firstdirichlet[1]">       <set var="Fire" value="false"/>     </effect>   </case>   <case>     <effect prob="seconddirichlet[0]">       <set var="Fire" value="true"/>     </effect>     <effect prob="seconddirichlet[1]">       <set var="Fire" value="false"/>     </effect>    </case> </rule>  请注意参数名称后面的括号，以指代多元Dirichlet的特定维度。  **<效果>的 XML格式（用于概率规则）：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | 概率 | 属性 | 0-1 | 效果的概率（固定或参数）。默认值是1。 | | <set ...> | 元件 | 1-n的 | 基本效果 |   每个效果内都是一个基本赋值给变量的列表。每个赋值由具有两个属性的<set ... />标记定义：var和value。  **<set ... />的 XML格式（用于概率规则）：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | VAR | 属性 | 1 | 变量标签 | | 值 | 属性 | 1 | 变量值 |   *实用规则*  规则也可以用来表达实用新型。效用规则根据特定的状态变量来定义特定动作（从系统角度）的效用。一般骨架与概率规则保持相似，不同之处在于这次的效果与特定的效用相关，而不是概率。以下是实用程序规则（规则r的示例2的[LISON（2014）](http://folk.uio.no/plison/pdfs/thesis/thesis-plison2014.pdf)，第69页。）：  <rule id="r2">   <case>     <condition>       <if var="Fire" value="true"/>     </condition>     <effect util="5">       <set var="Tanker" value="drop-water"/>     </effect>     <effect util="-5">       <set var="Tanker" value="wait"/>     </effect>   </case>   <case>     <effect util="-1">     <set var="Tanker" value="drop-water"/>     </effect>     <effect util="0">     <set var="Tanker" value="wait"/>     </effect>   </case> </rule>  规则r 2表示下水动作的效用是+5是有火（否则为-1），并且等待效用为-5的是有火，否则为0。  条件的定义与概率规则相似。效果也有类似的结构，但有一个例外：prob属性被util替换。效果中指定的变量（上例中的Tanker）是动作变量。  至于概率规则，效用可以是固定的或对应于要估计的参数。例如，规则r 2可以包括四个参数，这些参数表示系统动作的相应效用取决于情况：  <rule id="r2">   <case>     <condition>       <if var="Fire" value="true"/>     </condition>     <effect util="firstgaussian">       <set var="Tanker" value="drop-water"/>     </effect>     <effect util="secondgaussian">       <set var="Tanker" value="wait"/>     </effect>   </case>   <case>     <effect util="thirdgaussian">     <set var="Tanker" value="drop-water"/>     </effect>     <effect util="fourthgaussian">     <set var="Tanker" value="wait"/>     </effect>   </case> </rule>  **<效果>的 XML格式（用于实用规则）：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | UTIL | 属性 | 0-1 | 操作的实用程序（固定或参数）。默认值是0。 | | <set ...> | 元件 | 1-n的 | 基本效果 |   **<set ... />的 XML格式（用于实用规则）：**   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **基数** | **描述** | | VAR | 属性 | 1 | 变量标签（动作变量） | | 值 | 属性 | 1 | 变量值 |     **5.设置**  除了初始状态，参数和规则结构模型之外，对话域还可以包含特定系统设置以覆盖默认值。[[3]](http://www.opendial-toolkit.net/user-manual/dialogue-domains#%5B4%5D)  这些设置被定义为简单的元素列表：  <settings> <property1>value for property1</property1> <property2>value for property2</property2> .... </settings>  这些属性也可以通过GUI或通过向命令行添加-Dproperty = value标志来修改。  **<settings>的 XML格式：**  *（部分列表，请参阅*[*Settings.java*](https://github.com/plison/opendial/blob/master/src/opendial/Settings.java)*获取所有详细信息）*   |  |  |  |  | | --- | --- | --- | --- | | **内容** | **XML类型** | **值** | **描述** | | GUI | 元件 | 布尔 | 是否启动GUI | | 用户 | 元件 | 串 | 用户话语的变量标签 | | 系统 | 元件 | 串 | 系统话语的变量标签 | | 样本 | 元件 | 整数 | 采样时使用的样本数量 | | 时间到 | 元件 | 整数 | 最大采样时间（以毫秒为单位） | | 模块 | 元件 | 逗号分隔的列表 | 实施模块以附加到系统的类的列表 |   [1] 也可以定义多元高斯分布。在这种情况下，均值和方差的标量值由 <mean> [v1，v2，...，vn] </ mean>形式的向量值替换。多变量高斯分布仅支持具有对角协方差的时刻分布（即独立的高斯分布）。  [2]条件还可以包含嵌套运算符 <和>， <not>和 <或>（参见[高级建模：嵌套条件](http://www.opendial-toolkit.net/user-manual/advanced-modelling-1#6._Nested_conditions)）。  [3]默认设置可以在文件 resources / settings.xml中找到。 |