多维分析中的动态建模

**1 设计背景**

通常情况下，在一次多维分析中，需要向后台服务器传递执行的MDX（多维查询语句），多维模型定义（后面用Schema简称）及数据源信息，后台服务器在执行时无法改变Schema定义。

现在介绍一种在OLAP服务器执行时，动态修改Schema的设计思路，可以使用户不改变多维模型的情况下，通过上下文信息选择模型并改变执行结果，增加了多维分析的灵活性和可扩展性。

**2 技术方案**

多维模型是为了满足用户从多角度多层次进行数据查询和分析的需要而建立起来的基于事实表和维表的数据模型。

建立多维模型的作用在于，通过数据仓库可以根据不同的数据需求建立起各类多维模型，并组成数据集市开放给不同的用户群体使用，也就是根据需求定制的各类数据商品摆放在数据集市中供不同的数据消费者进行采购。

为了在相对稳定的多维模型基础上应对灵活多变的报表需求，需要根据上下文及报表请求信息，动态地修改多维模型定义进行查询分析。

对于传统的多维分析，它的流程是这样的：



如上图所示，客户端通过封装MDX、Schema和数据源信息，发送请求到OLAP服务器，获取查询分析结果。

对于上述过程的改进如下：



可以看出，改进后的OLAP服务器结构增加了规则处理层，在这个模块中，可以对Schema进行动态修改，比如根据上下文信息在定义中增加模型（立方体、指标等），以及增加对其他Schema引用的维度模型的处理（引用其他Schema中的维度模型，使多维模型更加简化，模型间实现共享）。

具体的实现逻辑是，Schema作为参数传递到后台服务器时是以XML形式存在的，在规则处理层中，解析XML文件，通过上下文信息，为XML增加或修改元素节点，达到修改Schema的目的。

**3 经验总结**

应用这个设计思路可以使多维模型定义相对简化、固定，根据报表需求，在查询分析中，把需求通过上下文形式传递到后台服务器。

对于产品来说，这个设计使应用于企业的多维分析更加灵活，可扩展性更高，使多维模型更加简化，降低建模的复杂度，简化报表开发人员的工作等。