

DSM（领域定义建模）和MDA（模型驱动架构）[3] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/141/2021_2022_DSM_EF_BC_88_E9_A2_86_E5_c29_141300.htm Defining Languages

and Interchanging Models 在OMG的MDA旗帜下还有另一个重要的技术：MOF。MOF是一个比UML更加抽象和难以理解的技术。理解了这个问题还有更难理解的术语，象metamodel（元模型）和meta-metamodel（元-元模型），感谢还没有出现meta-meta-meta-model，我们也会尽力阻止出现。MOF主要作两个工作。第一，它是一个被设计成定义建模语言的领域定义建模语言：一个MOF模型是一个MDA建模语言的定义。第二，它是一种计算一个MDA模型如何被序列化到一个XML文档或Java API的机制。一个领域的建模语言包含很多方面，它必须定义领域中的概念，必须把概念表示为图形或文本，必须定义用户如何与语言交互，必须定义一个模型是否合法，必须定义模型间如何交互。但是MOF仅仅定义了语言的基本概念，以及概念的模型如何存储和交互。一种语言的MOF说明并没有提供多少用户真正关心的东西：语言所包含的模型是什么，它看起来象什么，用户如何和模型交互。在微软，我们希望我们的语言能够整合到Visual Studio包括IntelliSense®，工具栏，菜单，属性栏，和对Debug的支持，我们发现定义如何对概念建模在整个工作中只是一个次要的方面，而且我们的语言定义工具要整合到Visual Studio中要比MOF好。事实上，尽管这是语言定义技术的通常的地位，MOF仍然是一个存储概念模型，并且使用XMI（XML Metadata Interchange）在模型和CORBA和Java API之间转换的

主要技术。如果使用MOF来定义一种语言的概念，接下来就可以使用XMI的方法来进行对语言的基于XML的自动生成。从这个方面看，这似乎是挺吸引人的，但是，还是有一些问题。首先，XML的生成基于语言的定义，这也就意味着使用UML1.4标准进行的XMI序列化将无法被基于UML2.0的实现所理解，除非用户在这些概念的观点能够保持前后一致。再者，XMI本身就在变化，也就是说可能会出现对与同一个模型的不同XMI序列化版本。第三，MOF的定义也在变化，它会为了对付不同的组合而不断加入新的元素，这将导致MOF的版本具有不同的取向，而且无法完全一致。所以，虽然XMI宣称为建模工具提供互操作性，但是实际情况是，除非每个工具都能够支持MOF，XMI，UML标准下的所有可能的组合，工具之间的交互才是没问题的。XMI的更深层次问题是，特别是对于旧版本，由机器生成的XML架构常常冗长且难以阅读，这就迫使开发者们去寻求可视化程度更高的，可转换的技术来维护XML文档。来源：www.examda.com 我们不认为XMI对于模型的序列化来说是正确的方法。XML正在变的成熟，市场上有大量的模式和工具。我们认为正确的方法是，对特定的建模语言有他自己特定的XML架构，并且提供工具来管理语言和序列化格式间如何自动解释和映射。如果对一个特定领域进行标准化，XML架构就可以是标准的，这是业界广泛存在的观点。之后，如果语言的定义发展了，可以在旧的XML架构上扩充，进行移植。XMI有效地阻止了这个清晰的思路发展，并导致了大量不兼容的XML架构标准，和它的互操作的目的完全背离了。简而言之，微软不支持MOF是因为下面的原因：1. 它还不是个稳定的标准 2. 使用

它来作为设计我们的工具的语言会产生我们不愿看到的结果。

3. 支持MOF所没有提供的元素需要商业级的实现，我们会继续引入MOF定义的改动。

4. MOF没有实现自己的目标。

结论：本文讨论了模型在软件开发中的角色，特别是domain-specific languages的定义和使用，以及在产品线中的使用，同时对OMG的MDA作了总体的评价。我们确信在敏捷软件开发过程中模型会得到更多的使用，我们正在构筑工具和技术来支持这些开发。我们看到UML作为重要的一步，它的未来是基于图释的开发者间的约定，而且可以作为面向特定问题领域的领域定义语言的灵感。也看到XML是模型的表现和交互的关键技术，我们期望对领域内容的标准化能尽早开始。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com