

AltiumTM

新一代元器件管理



一个设计元器件的传统模型如实地延展渗透到电子设计过程的各个方面。然而，如要将电子设计过程作为一个整体无缝融入到最终产品开发过程中，这种模式还需要发展 – 扩展到包括其他设计进程的其他方面，特别是 MCAD 和工业设计，以及业务流程 — 如采购和制造 — 这些与产品开发过程相关的方面。

将所有这些建模对象的不同方面连接起来做为一个统一的整体设计过程的不同部分。拥有一个单一的在各个不同的方面有不同模型的元器件对于设计师管理维护设计的真实的历史视图是至关重要的。

本白皮书将展示 Altium 公司如何通过其统一的元器件模型来解决这个问题的，而且运用这种模式的力量展示了一个元器件管理的新方法。

Altium 设计解决方案中的元器件管理的历史

Altium 公司多年来一直提供创新，领先的设计方案。在软件的各种产品 – 从 DOS 时代到现在与 Altium Designer — 设计中的元器件的管理一直是并将继续是软件中的基本要素。毕竟，没有元器件，就没有设计。如果没有这些元器件的有效管理，就没有其他的能力再利用它们进行一个新的设计。

如果我们退后一步思考多年来出现的不同的元器件管理方案，我们可以看到，最新的元器件模型和管理理念，其实是第四代。

第一代：简陋的原理图和 PCB 库模型（*.SchLib 和 *.PcbLib），普遍应用于八十年代中期的软件中。

第二代：集成化的元器件模型，通过集成库实现。在此模型中，元器件是更高层次的在原理图库文件（*.SchLib）中的原理图符号内建模。其他的链接到原理图符号和元器件参数的模型都添加到该原理图符号中。所有的源库 – 符号和连接的模型 — 都在一个库包项目（*.LibPkg）中定义，该项目将随后被编译为一个单一的文件，即一个集成库（*.IntLib）。该库编译成一个集成库的优势是，所有元器件的信息都在一个方便，安全的文件中可供使用。

第三代：出现了数据库的库（*.DbLib）以及版本控制的 SVN 数据库的库（*.SVNDbLib）。通常被冠名为“基于表格的库”，元器件的所有信息都按照一定格式存储在一个 Altium Designer 之外的文件中，比如 ODBC、ADO，或一个 Excel 电子数据表。在数据库中的每一条记录都代表一个元器件，存储有基于模型链接的所有参数、数据表参考信息或其他元器件信息。记录信息还可以包含到存货总量或其他同类元器件数据的链接。所有关于一个元器件的完整而详尽的信息都存储在数据表本身，而原理图符号仅作为一个图形化的表示，或仅仅是其中的一个模型。

第四代：即是“下一代”。在这个最新一代的元器件模型管理理念中，熟悉的原理图和 PCB 库模型继续扮演重要而基础的子部件角色。然而，元器件模型现在不仅仅为设计师作为工程设计领域提供信息，还在更广泛的产品级领域为采购和制造部门的人员提供访问。“统一的元器件”模型（在那里有元器件的基于文件的定义）是在设计层面进行执行操作，然后发布到一个安全的存储库，或称数据保险库。在这里，一个元器件被存储为一个唯一可识别的元器件条目的一系列修订本。每个修订本是生命周期管理的，提供被鉴定的元器件的搜集信息，并且被授权为可重新例化入新的设计项目，进而被制造为原型或者用于产品运作。简而言之，统一的元器件目录是通过基于数据保险库的库来实现的。

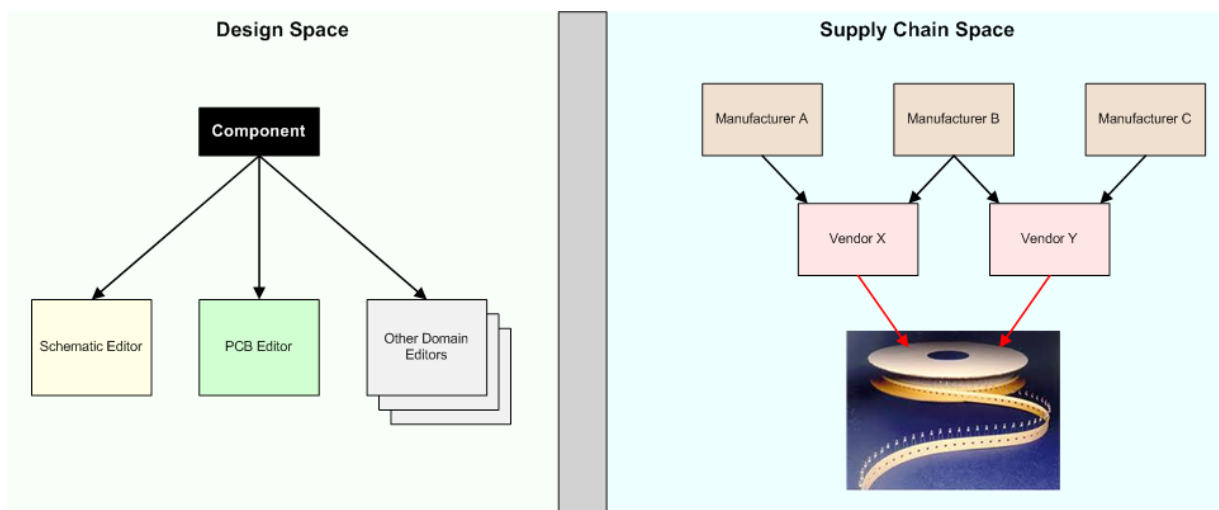
元器件的其他名称 ...

该术语“元器件”根据不同人的理解会有不同的意思。对于设计师而言元器件的概念与对于采购人员而言的元器件概念是有很大差别的。

在设计阶段，元器件是一个逻辑化的实体，设计师用它来解决一些问题或者实现某些功能。在原理图编辑领域，它有一个图形化的表示。在 PCB 编辑领域，它有另一个表示。在更多设计领域，它也将被重新以不同的形状或形式出现。但是到最后，对于设计师来说，领域是不相关的，而元器件都有不同的形状和表现形式，这个是元器件，那个还是元器件。

因此设计师基本上是对一个黑匣子实体在工作，该黑匣子必须拥有 特定的特征，并被设计师们以参数表来按照特性进行定义。而与这个黑匣子实体相关联的，即是“这是我们的元器件，而这是它所应该提供给我的用途”。

对于采购团队，元器件在更广泛的设计领域的功能性知识并不是必须的。采购专家们仅仅需要了解那些真正的物理元器件所应提供的信息。而采购专家需要同时对付制造商条目和供应商条目。前者是物理可制造的实体，而后者则是可购买的实体。在某些情况下，元器件的制造商也可能同时是销售该元器件的供应商。



该元器件从两个不同的观点来展示。一般来讲，外部沟通的关键环节是采购团队保证采用这些器件对应的供应商那里得到可制造的元器件，并且在设计层面，这些现实物理意义上的元器件可以实现相应的功能，一般要求这两个空间是由一个虚拟的“砖墙”分隔开来的。

因此，例如，设计人员指定了一个具有特殊性能的电阻 – 即 0405 封装的 10K 的电阻，5% 公差。这些参数都是在所设计的产品完整电路中需要实现其应有的功能特性。这可能会映射到从 Digi-Key 以某种价格购买的一卷 5000 片 Panasonic 电阻，或从 Mouser 购买的一卷 10000 片的 Yageo 电阻。

甚至进一步复杂化，设计工程师需要说其中哪些是确定以“补槽”的方式使用这种设计的元器件。这可能包括需要 1% 精度的电阻，即使设计部分一般都是 5% 的电阻。而最终，采购小组可能选择不买某些供应商的电阻，因为他们发现他们的交货是靠不住的。

因此，设计师的元器件视图的确与需要对该元器件的采购负责的人有不同的视角。但是，这两个世界并不一定互相隔离。如果这样隔离只能是费时又费钱。试想一个设计师在最最重要的时间节点，才发现来自采购小组的电子邮件说，部分需要的元器件已不能提供，或有需要几个星期的时间 – 而这几个星期的时间足以严重影响该公司关于产品上市的计划时间！

如果有这样一种方法该多好啊，它可以提供给设计师一个更好的“前期”图片可选择，选择采用哪种物理制造的元器件条目而使它在板级装配时可实现设计中的特定功能 ...

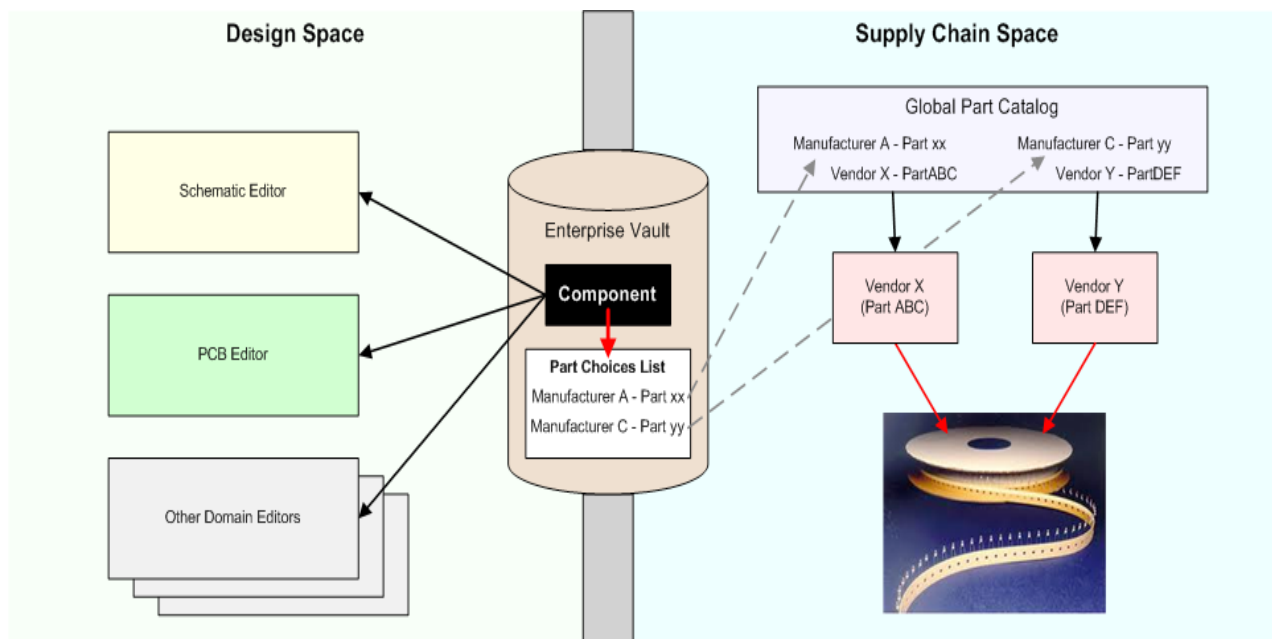
统一的元器件模型

Altium 的新一代元器件模型有效地绘制了设计元器件的概念 – 在传统的电子设计空间 – 并且可被更大的“产品空间”的组织者中的其他人查看。一个真正的“统一的元器件”模型，不仅可以代表不同的设计领域（原理图捕获，以及 2D/3D PCB 设计，仿真，信号完整性）的元器件模型，还可以促成在设计阶段对于所需的物理元器件（即现实世界的可制造条目）的选择，并在制造和装配方面提供对于采购成本和时间的优化配置。

根据这个模型范例，由设计师可见的设计元器件，将从具体的制造商或者供应商器件中独立出来。这些信息不仅包含元器件的信息，还会有一个单独的文件用来映射一个设计元器件到一个或多个元器件制造商（被列在全球元器件目录 Global Part Catalog 中）元器件，该制造商元器件反过来又可以映射到一个或多个供应商的元器件，使设计人员可以在前期了解哪些显示世界的元器件可以用于某个给定的设计中的设计元器件。

而元器件本身，以及对于它们的选择信息，都存储在一个中央储存库 – 一个企业数据保险库 – 能够有效地提供双方的共同点，是穿越该虚拟砖墙的“连接门”！

数据保险库促进了数据的高度完整的安全处理，同时提供给设计师和供应链相关人员在需要时进行数据的访问。不再通过电子邮件或电话对外联络。这些需要的信息都被定义为适合双方查看和操作。



术语“统一的元器件”描述了设计元器件的定义通过元器件选择功能被延展到更大的产品空间。元器件选择映射其设计元器件到必须由采购团队的人员购买的现实世界的可制造元器件。这给了采购团队可实时了解应该采购哪些元器件的灵活性。

以下各段落将在不同的方面近距离展示这种新的建模方案，以及它所带来的好处。

企业数据保险库服务器

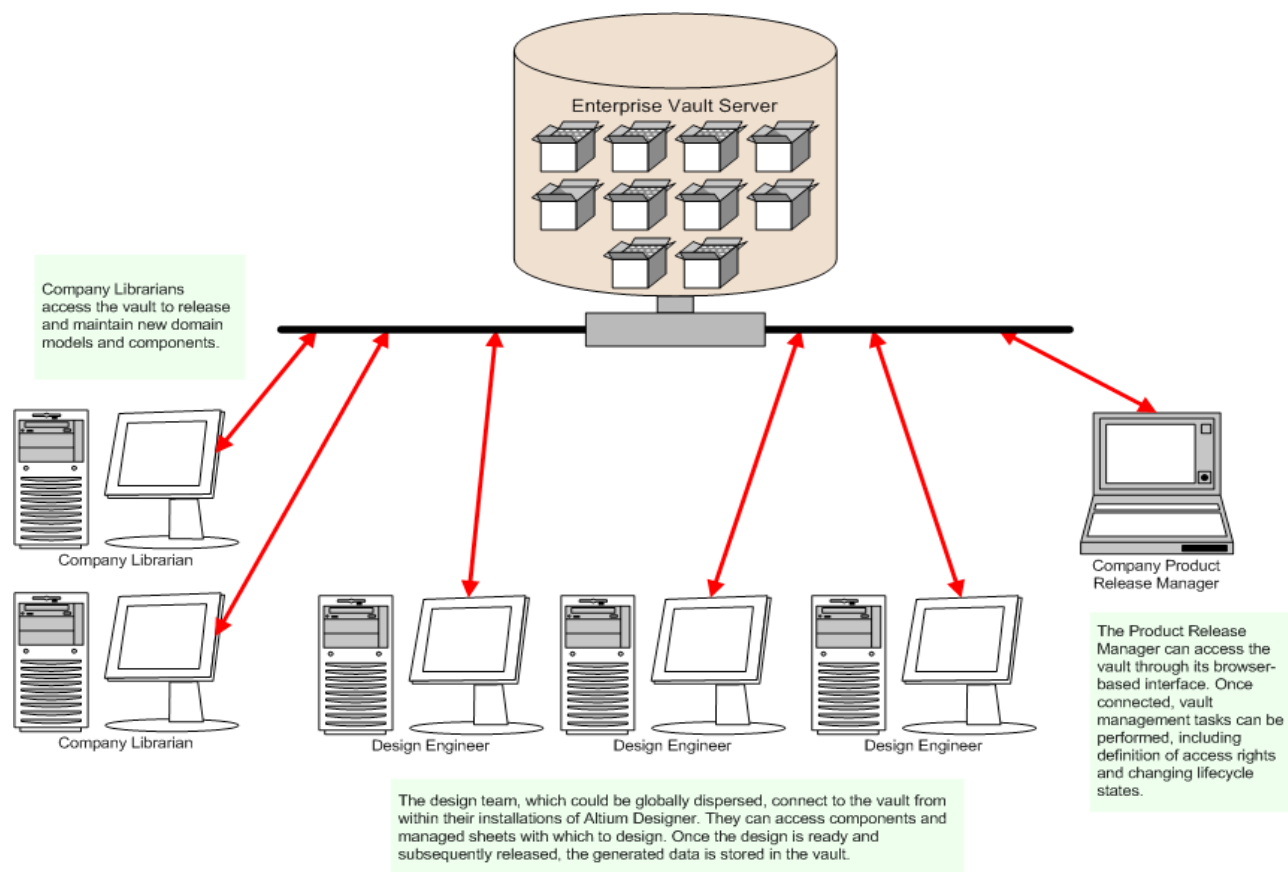
在了解元器件本身如何定义之前，更需要认真考虑他们在什么地方，以及那些相关的元器件选件列表都实际存在哪里。

在 Altium 的设计数据管理系统这个伞下，横向被分为两个部分 – 设计空间和供应链空间 – 类似于花园围墙边的两个邻居。在设计层面，我们的设计师团队（有时甚至只有一个人！）在进行创新设计。一个产品，在某种意义上说，应该可以满足并丰富购买和使用它的人的生活。在供应链层面，我们所有的团队都在为将一个设计最终变成一个物理产品而努力和负责。这些团队包括制造，采购，装配，测试等等。

供应链主要作用于，或使用的元器件数据大部分来自设计空间。此数据被认为是从设计空间“发布”过来。然而，究竟这些发布的数据发生了什么？它是如何处理，更何况，处理过程安全吗？换句话说，能保证数据的完整性吗？

需要有一个切实有效的手段来高度安全的方式，而且是容易被接受的方式处理这些数据。一个解决方案，不仅提供了坚如磐石的数据存储，而且对于不同的修改过的修订本提供数据的重新发布 – 可按时间跟踪设计变更，而不会覆盖任何以前发布的数据。一个解决方案，也可以对这些数据的生命周期进行管理，使那些需要使用这些数据的人只需一眼，就可以确定它到达其生命周期的哪个阶段，以及它可被安全的用作什么用途。

基于所有这些给定的因素，Altium 提供基于服务器的工程内容管理系统解决方案 — Altium 的企业数据保险库服务器。作为其本身独特的设计解决方案，企业数据保险库服务器与 Altium Designer 一起为数据的安全管理方面的各种问题提供了完美的答案。



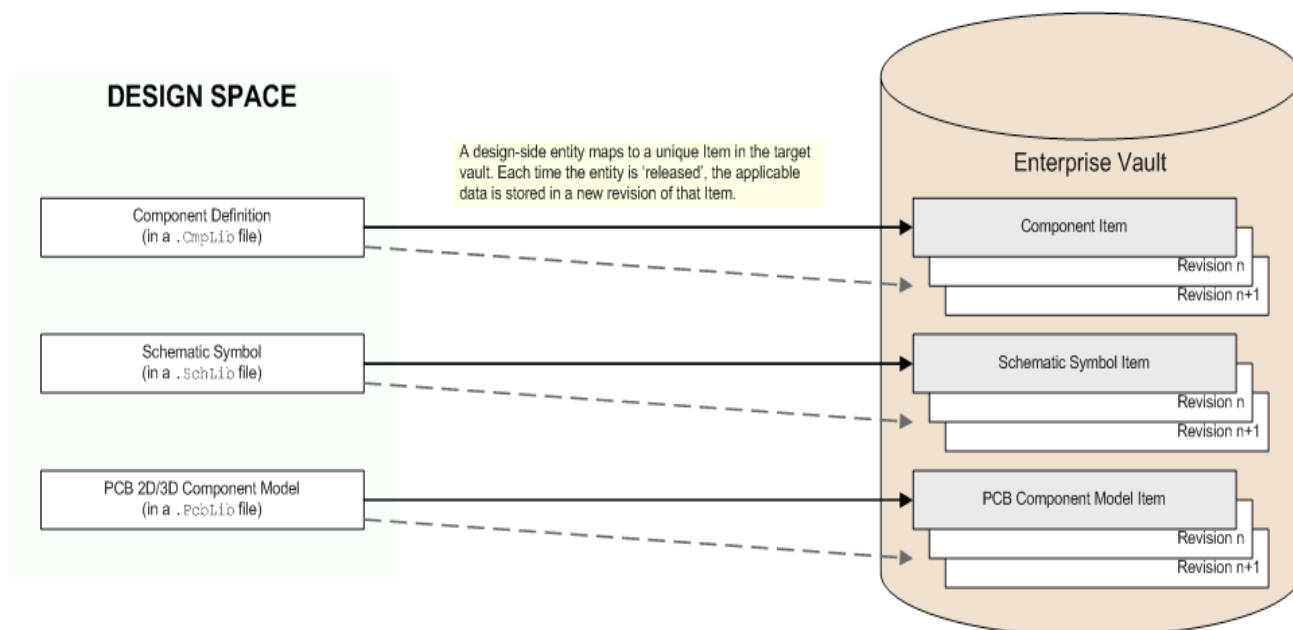
一个基于局域网的 Altium 公司的企业数据保险库服务器的概念。

事实上，一个企业数据保险库是用来存储不只是从设计空间发布过来的数据。它也可以用来处理其他数据，数据也可能来自供应链的空间。元器件选择列表是一个很好的例子，它是基于数据保险库的数据，不都是从设计方面而来，因此也不是一个“发布”实体。

各种数据实体（从设计空间或其他方式发布）在数据保险库里以唯一的条目来代表。一个条目仅代表一个单一特定的对象，而且被一个分配的条目 ID 唯一识别。企业数据保险库，是一个中央存储系统，将所有数据，以及每个条目储存其中。

每一个条目所代表的和所存储的内容究竟有何不同，取决于该条目所映射的是什么。在元器件管理方面，将有专门的条目类型来存储元器件本身的数据，以及它们所参考的各领域的模型。

在数据保险库，每个条目都存储为一系列的修订本。每一个修订本包含用于表示，或者构建该条目的特定版本的数据。每一次向源设计数据做了更改，数据保险库中将新建一个该条目的新版本，准备接受（存储）所生成的数据。



条目修订本的概念，展示了元器件的发布和它所参考的各领域模型。

设计元器件的定义

在我们关注一个设计元器件是如何将各个部分映射到真实世界中，并构成一个扩充的“统一元件”之前，我们先来看看它是如何定义和管理的。

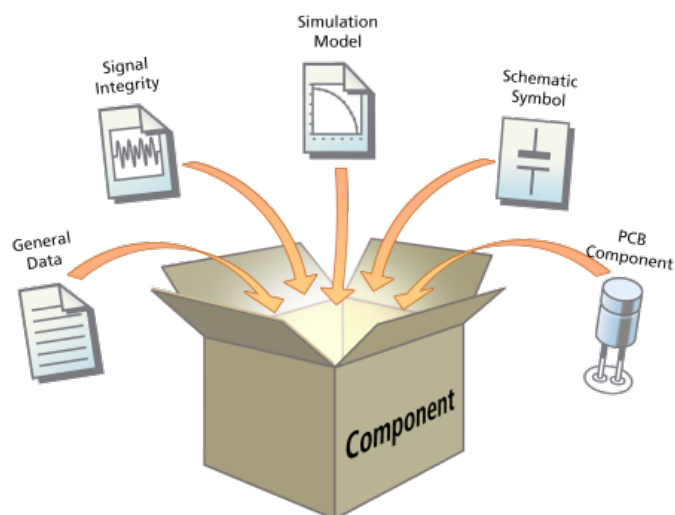
以工程设计或者研发的视角看来，从本质上讲，所有在设计领域中用于描述元件的模型信息，以及参数信息都是存储在一个“容器”中的（原理图符号，PCB 2D/3D 模型，SI 模型，Sim 模型，等），此外，未来需要为元件在其它设计领域建模，这种抽象的“斗式”建模方式是很容易扩展的。

但是如何具体指定一个元件？或者更准确地讲，如何确定箱子中的内容呢？在设计研发角度，答案是创建一个元件定义。

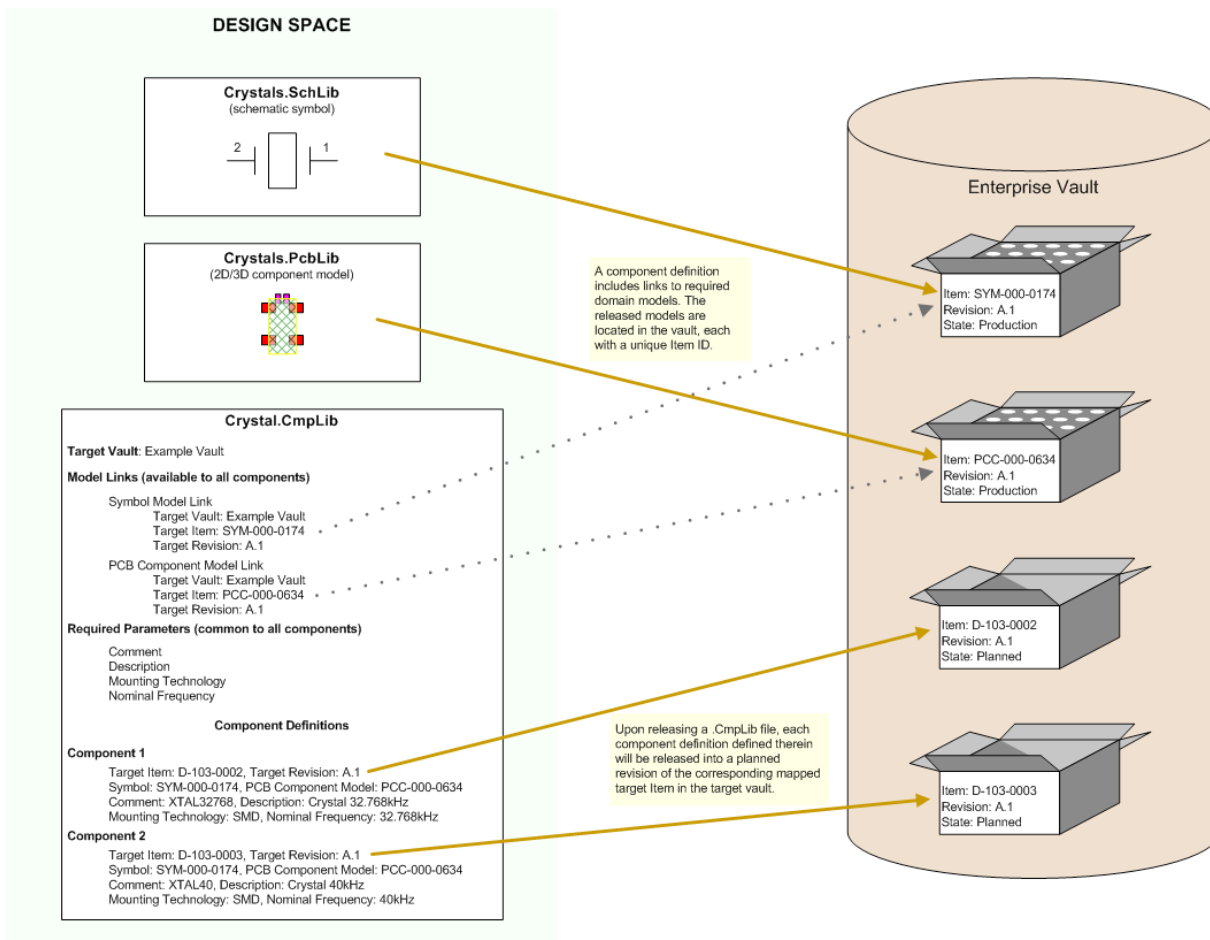
元件定义简单地讲就是——一个关于特定设计元件的定义。一个将元件必须的模型和参数捆绑在一起，并清晰有序地表达出来的定义。每个元件通过元件条目由设计端映射到发布端。换言之，是在详细说明源定义，发布之后，规定在设计中可以重复使用的元件。

从存储的方面来说，元件定义是在一个专用的元件库文件 (*.CmpLib) 中创建和管理的。单个的元件库文件，通过输入一个或多个元件定义，可以用来创建（或者说映射）一个或多个唯一的元件条目。每个元件定义关于不同领域模型的参数和链接都有相同的设置。

在元件库文件中按照要求定义好元件定义后，可以很简易地进行发布。发布过程仅仅是为每个目标条目生成新的版本号这一简单动作。



以设计研发视角看元件——一个将不同设计领域模型信息包含在内的整洁的包装箱。



元件的定义。一个元件在设计端所需的所有数据，都是在专用的元件库文件中指定的，包括链接更低一级的模型信息和参数信息。

各种模型

对于各个领域的表达方式，元件并不包含这些模型本身，而是包含到这些模型的链接。在设计端这些链接是作为元件定义（用来生成元件发布）的一部分进行指定的。同样在深入探究元件的定义和发布的过程之前，必须确保各个领域的模型本身已经创建好并发布出来了。

原理图符号 — 在设计端原理图符号库文件 (*.SchLib) 中已绘制好的一个原理图符号，被映射到发布端的一个原理图符号条目。一个原理图符号库中可能包含若干个定义好的原理图符号。发布过程中会将这些符号打散到独立的原理图符号库中去，每个库中只包含一个原理图符号。库文件的每个发布都会根据对应的链接条目，将符号模型数据保存为一个新的版本。这里要注意的是仅仅生成了原理图符号，也就是在原理图编辑中元件更高一级的表达方式。并不是像集成库中的“原理图元件”那样，将其他模型和参数作为原理图元件的一部分进行指定。更高级设计元件需要的仅仅是关于这个符号的图形描述。和其他领域模型的链接以及参数定义，将包含在元件库文件中。

如果符号来源于早期的库中，并不需要删除任何模型链接和参数。在发布过程中系统将剥离它们。这种“剥离”适用于已发布的数据，设计端的原理图元件源定义并没有改变，从而容许在 Altium Designer 支持的其他元件和库管理方式下，继续使用这些库。

PCB 元件模型 — 在设计端 PCB 库文件 (*.PcbLib) 中的一个 PCB 2D/3D 元件模型，被映射到发布端的一个 PCB 元件模型条目。一个 PCB 库中可能包含若干个定义好的模型。发布过程中会将这些模型打散到独立的 PCB 库中去，每个库中只包含一个模型。库文件的每个发布都会根据对应的链接条目，将模型数据保存为一个新的版本。

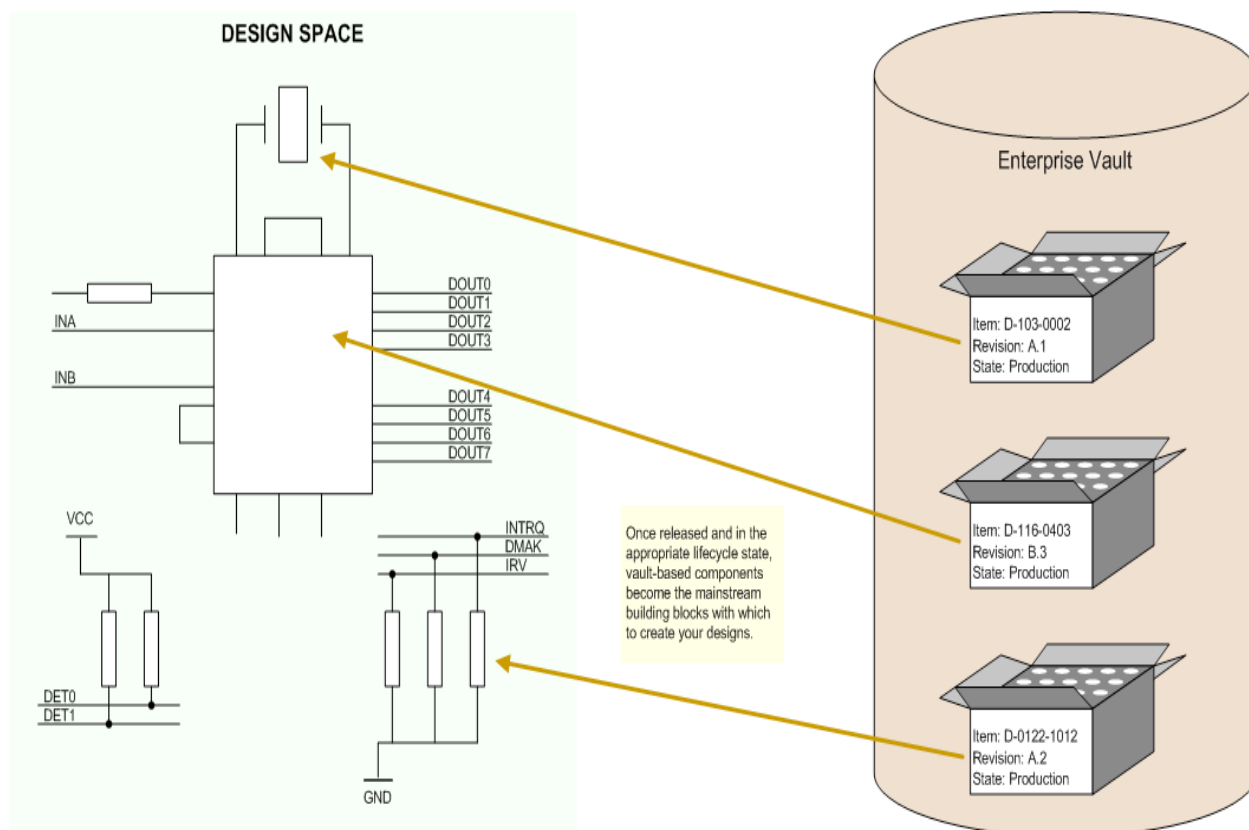
仿真模型 — 在设计端仿真模型文件 (*.SimModel) 中的一个仿真模型，被映射到发布端的一个仿真模型条目。文件的每次发布都会根据对应条目，将模型数据保存为一个新的版本。发布的数据中包含了引用 .SimModel，.mdl 以及 .ckt 文件定义的模型。对于模数混合元件，对应着 .mdl 以及 .scb 文件。

信号完整性模型 — 在设计端信号完整性模型文件 (*.SiModel) 中的一个信号完整性模型，被映射到发布端的一个信号完整性模型条目。文件的每个发布都会根据对应条目，将模型数据保存为一个新的版本。发布的数据中包含了由 .SiModel 以及 .SiLib 文件定义的模型。

针对原理图符号和 PCB 2D/3D 模型，系统提供了批处理的方式，可以将一个或多个库批量发布到某个文件夹中去。

应用经过验证的元件进行设计

一旦元件在数据保险库中发布并可用 — 或者叫做基于数据保险库的元件 — 可以被重新实例化到任何一个新的设计项目中、做成样机或者产品。另外元件验证的这一概念，使对元件的正规地修订和生命周期管理成为可能。这样用户可以指定元件的状态和元件的用途（设计、样机、产品，等）。从设计的角度，这么做的结果是创建基于数据保险库的库，库中包含了经过公司验证过的正规元件。



应用经过验证的元件进行设计！

在设计中使用经过验证的元件的好处是，当需要更改设计的生命周期的状态时，设计的完整性变得非常容易，因为一个设计只能发布为“样机”或者“产品”，同时规定了设计中所用元件的相关状态。换句话说，如果元件的状态处于“设计”级别，那么就不要开始生产这个装配好的板子。

另外，如果我们将这种方式从细粒度级进行元件管理的话，除非元件参考的所有模型都处于正确的状态，否则系统是不允许更改数据保险库中元件的生命周期的。换言之，父一级元件的生命周期是不能大于它的子一级模型的。

在设计阶段确定加工制造信息

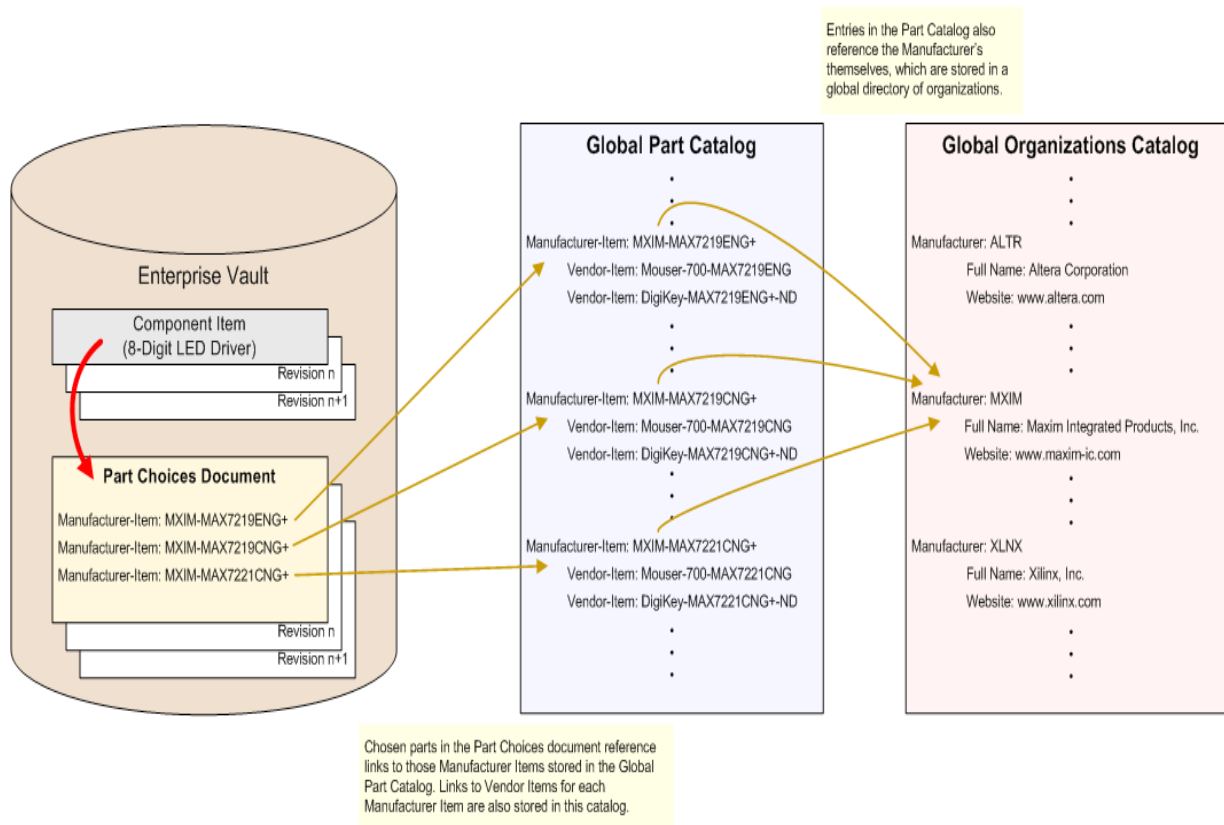
作为电路板设计者，应用跨越多页原理图上的元件的逻辑连线来捕捉设计的灵感。所用的元件应是已经定义好符号（以及其它领域的模型）和相关关键参数的，但是归根结底这些元件仅仅是“设计意向上的符号”，包含了项目设计中的抽象含义，却不包含实际的无理信息。无论是量产前进行采购，还是按照库存进行生产，都需要将每个元件“在真实世界具体化”。

在供应链中，采购专员经常弄不清楚（甚至他们需要或者希望了解！）某个元件在设计中的含义。他们迫切需要的是一个采购指示，哪个实际的物理器件可以用来实现那个抽象的元件。做出这个指示的最优人选，当然就是设计师本人。

最终要在生产和装配的电路板上指定这些元件，如果能将公司发布和管理元件，以及“批准”制造等附属信息在设计阶段进行确定该是多么惬意啊？ Altium 可以精确地提供全球器件分类和器件选择的服务。

全球器件分类是按照现货供应、器件的可采购性、厂家条目进行管理和维护的。每个厂家条目都链接到相应的父一级全球厂商目录的。此外，每个厂商条目还包含了到供应商条目的链接。供应商条目可以链接到某个器件销售公司。有时某些厂商是进行直销的。

由数据保险库中的一个元件条目，到全球器件分类中的厂商条目，是通过同样存储在数据保险库中的专用器件选择文档来完成它们之间的映射的。每个元件条目参照自己的器件选择文档。元件条目的版本同样利用这个相同的文档。器件选择文档是可以修改并可以很方便地进行生命周期管理的。并且器件选择文档作为一个单独的实体，是可以独立于参照它的元件条目进行更新的。



通过器件选择映射一个基于数据保险库的元件到真实世界的厂商。

设计者可以在设计阶段选择通用的器件。并且元件的映射是智能的，将基于数据保险库的元件，映射为真实世界统一标准的器件。

实时供应链信息

基于数据保险库元件的统一本质,通过对精选器件厂商和供应商的选择,最终生成一个到该器件的链接。在设计者看来,元件已经和供应链直接挂钩。Altium 没有忽略这一点是一件多么令人兴奋的事啊!

从供应商网站传回的数据是实时可用的,因此设计者可以从各个供应商那里获知所选器件(全球器件分类中定义的)的当前成本和可用性。而且并非只有设计者才能看到这些信息。采购专员也会关注这些供应链信息,同时可以确认数据保险库中每个元件的可用性。

当一个器件不再有效或者意外断货了该如何?没关系,一旦这类变化发生,实时的更新会立刻发送给设计者。有了这一重要提示,设计者可以避免选择这些不可用的器件。并且在任一时刻都可以将那些可靠的、可用的、经济的等价器件添加到清单里。

强大的用途反查能力

数据保险库中发布数据的高度关联结构,使其具有强大的“用途反查”能力。任何时候,根据数据保险库中的父条目,可以查看子条目用在何处。所以针对某个模型,可以快速确认哪些元件条目参考了它。针对某个元件条目,可以查看哪些设计、哪些原理图应用了它。

如果某个器件出于某种原因弃用或者过期了,您可以快速确认有哪些原理图和电路板应用了该器件,并且应用新版本器件或者其它等价的器件,来更新或重新发布设计。

如果设计处于生产阶段,所用器件已经经过验证并准备就绪,那么在下一个设计中您就可以充满信心地使用它了。

宏大系统的基石

下一代元件管理本身即是一个强大的体系。然而这一体系仍是更为宏大的系统的一部分。原理图设计复用管理和板级设计制造高度集成的发布,弥补了互利性的不足,与独立功能的子系统一道构成了 Altium 设计数据管理系统。

随着一些系统对元件可用性的依赖和使用,经过验证的元件不仅能够在本版本修订和生命周期进行跟踪,而且能够使用他们作为一个数据迁移的高级激励要素用于数据保险库元件中。

但并不是强制进行迁移。对这些元件可靠、集中的应用,打个比方,可以逐步地走入水里,而不是直接垂直的跳入水中。也许是进行元件生命周期的管理,或者是倾向对供应链信息的直接访问。现行的元件管理方法跟新的系统是可以兼容的。Altium 提供了相关工具,以便当您一切准备就绪,并且对系统的知识储备和操作充满信心,您可以很轻松的应用现有的元件库去创建新的基于数据保险库的元件库。而且一旦您使用了下一代元件管理系统,您将对其爱不释手。

Altium 的下一代元件管理系统是在不断发展的。它现在就像处于胚胎状态,提供了发展壮大成为一个重大系统的全部基础。一个通向基于 WEB 制造的系统,一个一设计 IP 形式提供分享和交易的系统。展望未来,板级设计中将集成更多的现场可升级器件,从 FPGA 到嵌入式 IP。真正基于云计算的器件管理,支持全球互联的智能元件的生态体系。