**OpenUniverse：一个全新的颠覆式的WinForm应用开发技术**

OpenUniverse是一个开源项目，其目标是：

* 1、给Win32应用提供一个灵活、易于描述、面向互联网的UI布局引擎；
* 2、给每个Win32应用提供一个内置的现代浏览器模型
* 3、让Win32应用全面支持.NET Framework(设计时以及运行时)

OpenUniverse允许软件开发者用宇宙结构的层次感重新看待每一个WinForm应用，使得：

1. 彻底突破桌面开发与Web开发的边界，以适应新一代互联网环境下的软件需求；
2. 淡化桌面应用与互联网浏览器之间的差异，每一款Win32或者WinForm应用事实上都可以内置现代浏览器模型，使得应用在默认状态之下就是一款面向互联网的浏览器；
3. 每一个WinForm或者Win32应用都具有一个充分大的对象模型，拥有自己的基于Web技术的DOM体系
4. 每一个二进制的组件（.NET Component或者Win32组件）都具备Web特征，以此形成一种对偶模式；
5. 每一种桌面软件都具备面向Web的内容服务基础，互联网浏览器是一种特别的桌面应用，由于其支持Web页面，所以其内容生态奠定了互联网浏览器现实的地位，我们认为这个模式限制在浏览器范围之内是对生产力的一种禁锢，Universe的使命之一是将这种页面支撑能力拓展到大众化的桌面应用之中，使得互联网内容服务是一种公共的而不是浏览器独有的特征。

**OpenUniverse: a brand-new subversive WinForm application development technology**

**OpenUniverse is an open source project whose goals are:**

1. Provide a flexible, easy-to-describe, Internet-oriented UI layout engine for Win32 applications;
2. Provide a built-in modern browser model for each Win32 application
3. Let Win32 applications fully support .NET Framework (design time and runtime)

OpenUniverse allows software developers to re-look at every WinForm application with a sense of hierarchy of the universe structure, making:

1. Thoroughly break the boundary between desktop development and Web development to adapt to the software needs of the new generation of Internet environments;
2. To downplay the difference between desktop applications and Internet browsers, every Win32 or WinForm application can actually have a built-in modern browser model, making the application an Internet-oriented browser by default;
3. Every WinForm or Win32 application has a sufficiently large object model and its own DOM system based on Web technology
4. Every binary component (.NET Component or Win32 component) has Web characteristics, thus forming a dual mode;
5. Every desktop software has the basis of Web-oriented content services. Internet browser is a special desktop application. Because it supports Web pages, its content ecology has established the real status of Internet browsers. We believe this model Being restricted to the browser is a kind of restriction on productivity. One of the missions of Universe is to extend this page support capability to popular desktop applications, making Internet content services a kind of public rather than browser-only Some characteristics.

**OpenUniverse: a New Subversive WinForm application development technology**

**OpenUniverse is an open source project with the objectives of:**

1. Provide a flexible, easy to describe and Internet Oriented UI layout engine for Win32 applications;
2. Provide a built-in modern browser model for each Win32 Application
3. Make the.net framework (design time and runtime) fully supported by Win32 applications

OpenUniverse allows software developers to take a fresh look at each WinForm application with a sense of hierarchy of the universe structure, making:

1. Break through the boundary between desktop development and web development to adapt to the software requirements in the new generation Internet environment;
2. The differences between desktop applications and Internet browsers are diluted. In fact, every Win32 or WinForm application can have a built-in modern browser model, so that the application is an Internet Oriented browser by default;
3. Each WinForm or Win32 Application has a sufficiently large object model and its own DOM system based on Web technology
4. Each binary component (. Net component or Win32 component) has web characteristics, which forms a dual pattern;
5. Every desktop software has a web oriented content service foundation. Internet browser is a special desktop application. Because it supports web pages, its content ecology has established the status of Internet browser in reality. We think that this mode limited within the scope of browser is a kind of constraint on productivity. One of the missions of universe is to support this kind of page To expand to popular desktop applications, Internet content service is a public rather than a browser unique feature.

**从伽利略到赫歇尔再到哈勃，宇宙观测的每一次突破都是一次认知的颠覆**

伽利略发明了人类第一个真正意义上的天文望远镜，使得人类可以一睹太阳系的风采，赫歇尔则将人类的视野扩大到银河系，哈勃的出现，使得人类得以一窥宇宙的一斑，哈勃带来的影响是极其深刻的，即使是爱因斯坦，都不得不因为哈勃的发现而修改其相对论的关键环节。截至到今天为止，我们认知的宇宙，是以星系团以及更多更大的天体结构为结构成员的宇宙体系，这一切得益于NASA的Hubble，没有Hubble天文望远镜，我们很难想象这一切。

肉眼观天的时代，许多环节只能靠想象解释，所以人们创造了很多“神话”，这一点中国与外国几乎没有区别。由于肉眼的局限性，人们只能看到有限的宇宙，以今天的尺度看，肉眼之下的宇宙非常非常的“小”，我们知道，一些由哈勃拍摄的天体图像，图像中一个像素的大小估计就有5光年，远远大于我们太阳系的尺度，很多时候，我们看到的全部，可能距离真实的场景相差极大。

回到WinForm程序，许多人会说，我们的软件只有有限的几个WinForm，这些与你所说的天文观测有什么关系？历史告诉我们，如果我们不去观天，地球就是一个囚笼，我们会被永远的囚禁在这个小小的星球之上，同样，对于WinForm应用开发而言，如果我们不能从宇宙观测之中找到启发，那么，我们的世界就会是另外一个囚笼，会极大的限制我们的创造力。

**From Galileo to Herschel to Hubble, every breakthrough in cosmic observation is a cognitive subversion**

Galileo invented the first astronomical telescope in the true sense of mankind, allowing mankind to have a glimpse of the solar system, Herschel expanded the field of vision of mankind to the Milky Way, and the emergence of Hubble gave mankind a glimpse of the universe. The influence of Hubble is so profound that even Einstein had to modify the key links of his theory of relativity because of the discovery of Hubble. Up to now, our understanding of the universe is a cosmic system with galaxy clusters and more and larger celestial structures as its structural members. All this benefits from NASA's Hubble. Without Hubble telescope, we can hardly imagine all this.

In the era of observing the sky with the naked eye, many links can only be explained by imagination, so people have created a lot of "myths", which is almost the same in China and foreign countries. Due to the limitations of the naked eye, people can only see a limited universe. At today's scale, the universe under the naked eye is very, very "small". We know that in some celestial images taken by Hubble, the size of a pixel in the image is estimated to be 5 light years, which is much larger than the scale of our solar system. In many cases, everything we see may be very different from the real scene.

Going back to the WinForm program, many people will say that our software has only a limited number of WinForms. What does this have to do with the astronomical observations you mentioned? History tells us that if we don’t observe the sky, the earth is a cage, and we will be imprisoned forever on this small planet. Similarly, for WinForm application development, if we cannot find inspiration from observations of the universe, then our world will be another cage, which will greatly limit our creativity.

**From Galileo to Herschel to Hubble, every breakthrough in cosmic observation is a cognitive subversion**

Galileo invented the first astronomical telescope in the true sense of mankind, so that people can have a glimpse of the splendor of the solar system. Herschel expanded the vision of human beings to the Milky Way system. The emergence of Hubble enabled human beings to have a glimpse of the universe. The impact of Hubble was extremely profound. Even Einstein had to modify the key rings of his theory of relativity because of Hubble Section. Up to now, our understanding of the universe is a cosmic system with galaxy clusters and more and larger celestial structures as its structural members. All this benefits from NASA's Hubble. Without Hubble telescope, we can hardly imagine all this.

In the era of naked eye view of the sky, many links can only be explained by imagination, so people have created a lot of "myths", which is almost no different between China and foreign countries. Due to the limitations of the naked eye, people can only see a limited universe. In today's scale, the universe under the naked eye is very, very small. We know that the size of a pixel in some images of objects taken by Hubble is estimated to be 5 light years, which is far larger than the scale of our solar system. In many cases, what we see may be very different from the real scene 。

Back to the WinForm program, many people will say that our software only has a limited number of WinForms. What does this have to do with what you call astronomical observation? History tells us that if we don't look at the sky, the earth will be a cage, and we will be forever imprisoned on this small planet. Similarly, for WinForm application development, if we can find inspiration from cosmic observation, then our world will be another cage, which will greatly limit our creativity.

按照OpenUniverse的视角，每一个WinForm应用都是一个某种意义上的宇宙。或者都包含一个自己的宇宙体系结构。听起来，这个观点似乎很荒谬，一个普通的WinForm应用程序，怎么可能会有自己的宇宙体系结构？

According to **OpenUniverse**'s perspective, every WinForm application is a universe in a sense. Or both contain their own cosmic architecture. It sounds like this point of view seems absurd. How can an ordinary WinForm application have its own universe architecture?

From the perspective of openuniverse, every WinForm application is a universe in a sense. Or they all have their own cosmic architecture. It sounds ridiculous. How can an ordinary WinForm application have its own cosmic architecture?

**WinForm应用是一个某种意义上的宇宙**

如果一个WinForm应用拥有类似宇宙一样的层次结构，我们需要回答几个关键的问题：

* 1、WinForm对象在这个宇宙里面处于什么位置？
* 2、这个宇宙里面有星系吗？
* 3、这个宇宙里面的各类星体在哪里？
* 4、我们怎样看待这个宇宙？

根据OpenUniverse提供的技术规格，每一个WinForm应用，都是WinForm可执行程序本身外加一个巨大的“组件空间”，这个组件空间至少包含“所有包含于dll库里面的UI组件”，这是一个足够大的组件集合

一般说来，每一个具体的WinForm上，至少包含一个控件，其Dock属性要么是“Fill”，要么是“None”。如果WinForm可以解释为一种宇宙之中的天体结构，那么这种天体结构就是现代宇宙领域里面的“星系团”，而星系团的每一个星系的核心，就是Form窗体上那些Dock属性为“Fill”或者“None”的控件。我们在OpenUniverse里面给出一种方案，让环绕在这类控件周围的那些形形色色的对象得以呈现。

Generally speaking, every specific WinForm contains at least one control, and its Dock property is either "Fill" or "None". If WinForm can be interpreted as a kind of celestial structure in the universe, then this kind of celestial structure is the "galaxy cluster" in the modern universe, and the core of each galaxy in these galaxy clusters corresponds to the .NET UI elements whose Dock attribute value is "Fill" or "None". We give a solution in OpenUniverse, so that all kinds of objects surrounding such controls can be presented.

在OpenUniverse项目里面，我们实现了一个非常重要的核心对象，我们将之命名为“Hubble”，这个对象就是针对Dock属性值为“Fill”或者“None”的那些.NET控件、WPF对象等等。

In the **OpenUniverse** project, we implemented a very important core object, which we named "Hubble". This object is aimed at the. Net controls and WPF objects whose dock property value is "fill" or "None".

**WinForm applications are a universe in a sense**

If a WinForm application has a hierarchy similar to the universe, we need to answer several key questions:

1. Where is the WinForm object in this universe?
2. Are there galaxies in this universe?
3. Where are the various stars in this universe?
4. How do we view this universe?

**WinForm application is a universe in a sense**

If a WinForm application has a hierarchy similar to the universe, we need to answer several key questions:

1. Where is the WinForm object in this universe?
2. Are there any galaxies in this universe?
3. Where are all kinds of stars in this universe?
4. How do we view the universe?

如果WinForm应用真的就是一个相对独立的宇宙，那么其中的日月星辰在哪里？

If the WinForm application is really a relatively independent universe, where are the sun, moon and stars?

我们知道，WinForm开发的最基本的可编程元素包含.NET Control、WinForm、WPF……，这些对象隶属于.NET Framework，对某个特定的应用而言，通常情况之下，一个具体的.NET Project只能包含有限个我们刚刚提到的元素，每个工程都有一个自己的边界，如同一个小的笼子，将自己的元素放在里面。这样的工程，要么是一个独立的可执行文件，要么是一个组件库。自从.NET Framework诞生以来，一直就是这个状态。

We know that the most basic programmable elements of WinForm development include. Net control, WinForm, WPF,..., These objects belong to the. Net framework. For a specific application, a specific. Net project can only contain a limited number of elements we just mentioned. Each project has its own boundary, just like a small cage, in which its own elements are placed. Such a project is either an independent executable file or a component library. This has been the state since the birth of the. Net framework.

从数学的角度看问题，现代数学习惯于将同一类东西放在一起，进而形成一个全新的集合，然后在这个集合之上定义新的规则，统一的作用在集合的元素之上，例如，向量空间就是一个典型的例子，群也是一个例子，数学的这种同类对象聚合进而形成一类新的数学结构是现代数学的一个明显的法则。如果我们按照类似的思维看待.NET WinForm开发，我们尝试着想象一下，如果将.NET UI对象都聚合在一个新的集合之中，这个集合会有多大？

From a mathematical point of view, modern mathematics is accustomed to putting the same kind of things together to form a brand new set, and then define new rules on this set, and the unified effect on the elements of the set, for example, vector Space is a typical example. Groups are also an example. The aggregation of similar objects in mathematics to form a new mathematical structure is an obvious law of modern mathematics. If we look at .NET WinForm development in a similar way, we try to imagine how big the collection will be if all the .NET UI objects are aggregated in a new collection?

From the point of view of mathematics, modern mathematics is used to putting the same kind of things together to form a new set, and then define new rules on this set and unify the elements of the set. For example, vector space is a typical example, and group is also an example. This kind of mathematical objects aggregate to form a new set Aspect is an obvious feature of modern mathematics. If we look at. Net WinForm development in a similar way, we try to imagine how large the collection would be if we aggregated all the. Net UI objects into a new collection?

很显然，这是一个包含无穷多个元素的超大的集合，我们将这个集合里面的成员分成两大类，第一类，元素隶属于一个dll库，第二类，元素隶属于exe。在数学上，这个集合的结构类似于一种数学结构：纤维丛，我们会在后续的介绍中给出这类结构一个系统的介绍。就单一一个WinForm应用而言，一个比较明显的“纤维”是指由这个程序本身做为纤维的基点，在这个基点上“生长出”一根庞大的纤维：就是那些存在于所有dll库之中的.NET UI组件形成的集合。从抽象的角度看，这个由包含在dll库里面的UI组件形成的集合足够庞大，差不多可以说“包罗万象”了。我们认为，这里存在一个“宇宙”，以应用程序可执行体本身包含的成员，以及整个巨大纤维里面的成员形成的一个新的“宇宙空间”。

Obviously, this is a very large collection containing infinitely many elements. We divide the members in this collection into two categories, the first category, the elements belong to a dll library, and the second category, the elements belong to the exe. Mathematically, the structure of this set is similar to a mathematical structure: fiber bundles. We will give a systematic introduction to this type of structure in the subsequent introduction. As far as a single WinForm application is concerned, a more obvious "fiber" refers to the program itself as the base point of the fiber, and a huge fiber is "grown" from this base point: those that exist in the dll library A collection of .NET UI components. From an abstract point of view, this collection of UI components contained in the dll library is large enough to be almost "all-encompassing". We believe that there is a "universe" here, a new "cosmic space" formed by the members of the application executable itself and the members of the entire huge fiber.

Obviously, this is a very large collection of infinitely many elements. We divide the members of this collection into two categories. The first category is that the elements belong to a DLL library, and the second category is that the elements belong to exe. Mathematically, the structure of this set is similar to a mathematical structure: fiber bundle. We will give a systematic introduction to this kind of structure in the following introduction. For a single WinForm application, a more obvious "fiber" refers to the program itself as the fiber base point, on which a huge fiber "grows": the collection of. Net UI components existing in the DLL library. From an abstract point of view, the collection of UI components contained in the DLL library is large enough to be said to be "all inclusive". We believe that there is a "universe", a new "cosmic space" formed by the members contained in the application executable itself, as well as the members within the entire giant fiber.

如果我们有了我们心目中的宇宙，Form对象是什么？我们的答案是，Form应该是这个宇宙之中的星系团，是不是听起来比较荒诞？

If we have the universe in our mind, what is the Form object? Our answer is that Form should be a cluster of galaxies in this universe. Does it sound absurd?

If we have the universe in our mind, what is the form object? Our answer is, form should be a galaxy cluster in the universe, doesn't it sound absurd?

一切从Dock属性开始

我们知道，每个.NET UI对象（WinForm、Control、 WPF Control）都有Dock属性，这个属性有6个值，由此，我们将.NET UI对象按照Dock属性分为两类，第一类是Dock属性值为“left、right、top、bottom

”这类控件只能环绕在某一个UI对象的边缘；另外一类Dock属性值为“Fill、None”，第二类对象是值得我们关注的，事实上，在.NET设计逻辑里面，第二类对象会被其他对象环绕着，在我们的宇宙里面，第二类对象非常重要，这些对象相当于宇宙里面的星系核，听起来非常不可思议，星系核应该被大量的星体环绕着，我们却没有可见。

**Everything starts with the Dock property**

We know that every .NET UI object (WinForm, Control, WPF Control) has a Dock property. This property has 6 values. Therefore, we divide the .NET UI object into two categories according to the Dock property. The first category is Dock attribute values are "left, right, top, bottom

"This type of control can only surround the edge of a certain UI object; another type of Dock attribute value is "Fill, None", the second type of object is worthy of our attention, in fact, in the .NET design logic, the second Class objects are surrounded by other objects. In our universe, the second class objects are very important. These objects are equivalent to the nucleus of the galaxy in the universe. It sounds incredible. The nucleus of the galaxy should be surrounded by a large number of stars, but we don’t visible.

It all starts with the dock property

As we know, each. Net UI object (WinForm, control, WPF control) has a dock property, which has six values. Therefore, we divide the. Net UI objects into two categories according to the dock property. The first category is the dock property with values of "left, right, top, bottom"

”This kind of control can only surround the edge of a UI object. The other type of dock property value is "fill, none". The second type of object is worthy of our attention. In fact, in the. Net design logic, the second type of object will be surrounded by other objects. In our universe, the second type of object is very important. These objects are equivalent to the galactic nuclei in the universe. It sounds like this It's incredible that the galactic nucleus should be surrounded by a lot of stars, but we don't see it.

类星体与Dock属性

在近代天文学之中，类星体是一类最重要的天体，最初被发现的时候，这类天体是一类强射电源，一个偶然的机会，人们发现了这类射电源的光学对应物，看上去是一个有“圆面”的星体，异常明亮，由于有巨大的红移，所以按照哈勃定律，这类天体远离我们的速度非常之快，一个直径4光年的类星体，远离我们的速度可以达到光速的70%，非常难以想象，人们发现了这类天体，相当长的时间之内不知道它们究竟是什么，从上世纪60年代发现，一直到2011年哈勃分离出环绕在类星体周围的那些“旋臂”，人们才意识到，类星体是一类剧烈活动的星系的“核”，从其名字可以看出，最初，人们认为这些天体似乎是一种类似“恒星”的天体，但远比恒星要大得多。

Quasars and Dock properties

In modern astronomy, quasars are one of the most important celestial bodies. When they were first discovered, such celestial bodies were a kind of strong radio source. By chance, people discovered the optical counterpart of this kind of radio source. It is a star with a "round face", which is extremely bright. Due to the huge redshift, according to Hubble's law, this kind of celestial body moves away from us very fast. A quasar with a diameter of 4 light years is far away from us. The speed can reach 70% of the speed of light, which is very unimaginable. People have discovered such celestial bodies, and they don’t know what they are for quite a long time. From the discovery in the 1960s, until Hubble separated from the surrounding class in 2011 People realize that quasars are the “nucleus” of a class of violently active galaxies. As you can see from their names, people initially thought that these celestial bodies seemed to be similar to “stars”. Celestial bodies, but much larger than stars.

Quasars and dock properties

Quasars are one of the most important celestial bodies in modern astronomy. When they were first discovered, quasars were a kind of strong radio source. By chance, people found the optical counterpart of this kind of radio source. It seems that it is a star with a "round surface", which is extremely bright. Due to the huge red shift, according to Hubble's law, such objects are far away from our speed The speed of a quasar with a diameter of 4 light-years can reach 70% of the speed of light far away from us. It is very hard to imagine. People found such objects and did not know what they were for a long time. From the discovery in the 1960s, until the separation of the "spiral arms" around the quasars by Hubble in 2011, people realized that quasars are The names of the nuclei of a class of violently active galaxies suggest that at first, these objects seemed to be similar to "stars", but much larger than stars.

在WinForm开发过程中，那些Dock属性为“Fil”l或者“None”的控件是一类非常常见的可编程对象，这类对象与类星体差不多，许多关键的信息并没有被人们发现，OpenSource项目的关键要素之一是，如果WinForm应用本身具有一个宇宙结构，那么这类控件相当于宇宙之中的类星体，问题的关键是我们如何看到这些类星体周围环绕的那些星体

In the process of WinForm development, those controls whose Dock property is "Fill" or "None" are a very common type of programmable object. This type of object is similar to a quasar, and many key information has not been discovered by people. The OpenSource project One of the key elements is that if the WinForm application itself has a cosmic structure, then this type of control is equivalent to the quasars in the universe. The key to the problem is how do we see the stars surrounding these quasars

In the development process of WinForm, the control whose dock property is "fill" or "None" is a very common kind of programmable object. This kind of object is similar to quasar, and many key information has not been found. One of the key elements of opensource project is that if WinForm application itself has a cosmic structure, then this kind of control is equivalent to the class in the universe Stars, the key is how we see the stars around these quasars

**如果你拥有哈勃，你将打开一个全新的宇宙**

非常不幸，Hubble归属于NASA，所以整个天文界都要服从NASA的规范才可能使用Hubble，Hubble造就了NASA的垄断地位。在WinForm世界里，每一款应用，都有自己的宇宙体系，通过OpenUniverse，我们给每一款软件装配了属于应用本身的Hubble，在那里，开发者可以运用Hubble观测他感兴趣的一切

**If you own Hubble, you will open up a whole new universe**

Unfortunately, Hubble belongs to NASA, so the entire astronomy community must obey NASA's specifications before using Hubble. Hubble has created NASA's monopoly. In the WinForm world, each application has its own universe system. Through OpenUniverse, we equip each software with Hubble that belongs to the application itself, where developers can use Hubble to observe everything they are interested in

If you have Hubble, you will open up a whole new universe

Unfortunately, Hubble belongs to NASA, so the entire astronomical community has to obey NASA's specifications before they can use Hubble. Hubble creates NASA's monopoly. In the WinForm world, every application has its own universe system. Through openuniverse, we assemble each software with its own Hubble, where developers can observe everything they are interested in

一个全新的面向互联网的可描述的多层次动态UI引擎

为了充分的挖掘环绕在那些属性为Fill或者None的.NET Control的周围环绕物，我们需要一种全新的“虚拟布局引擎”（VLE: Virtual Layout Engine），我们需要几个基本概念。