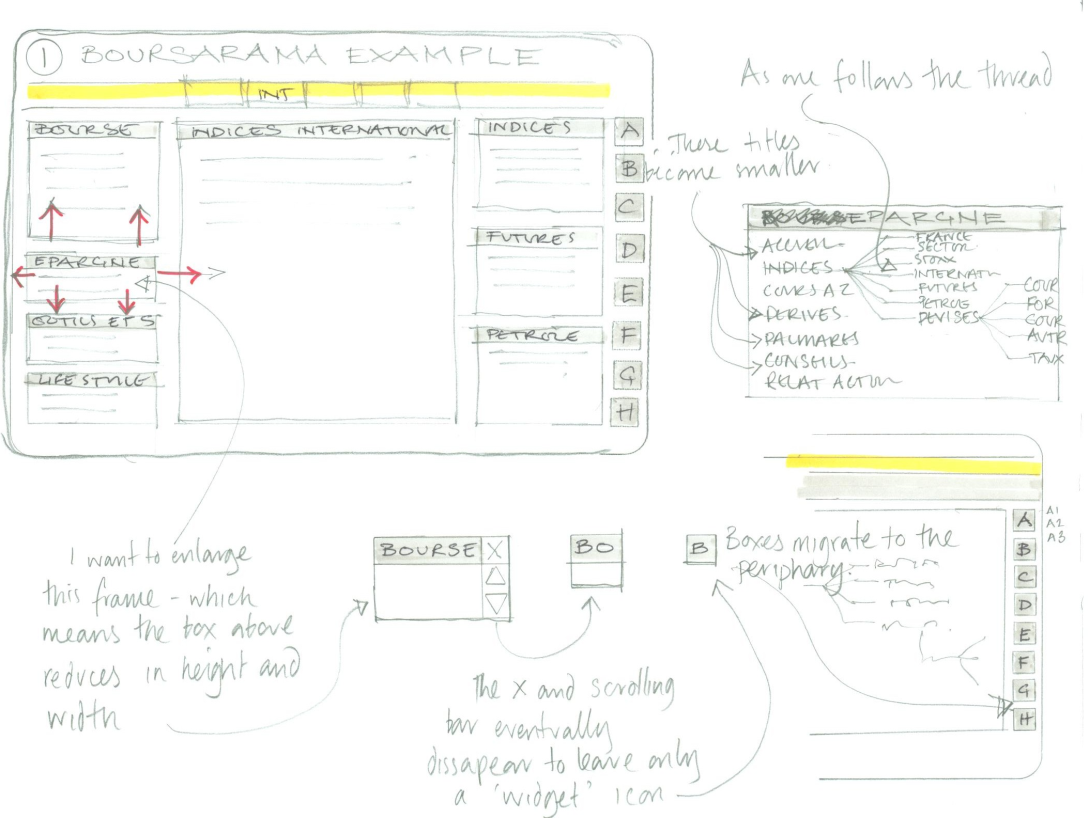
# The ZDragon Desktop concept

*Zoomable User Interface (ZUI): The graphical user interface (GUI) is based on a graphical environment where users can change the scale of the viewed area to see more detail or less*

The name resembles the SeaDragon technology created by Microsoft. The Z letter means Zoomable. The 3D PhotoSynth application uses the 2D SeaDragon to visualize the images.

## The sketches that were sent

  
Figure 1

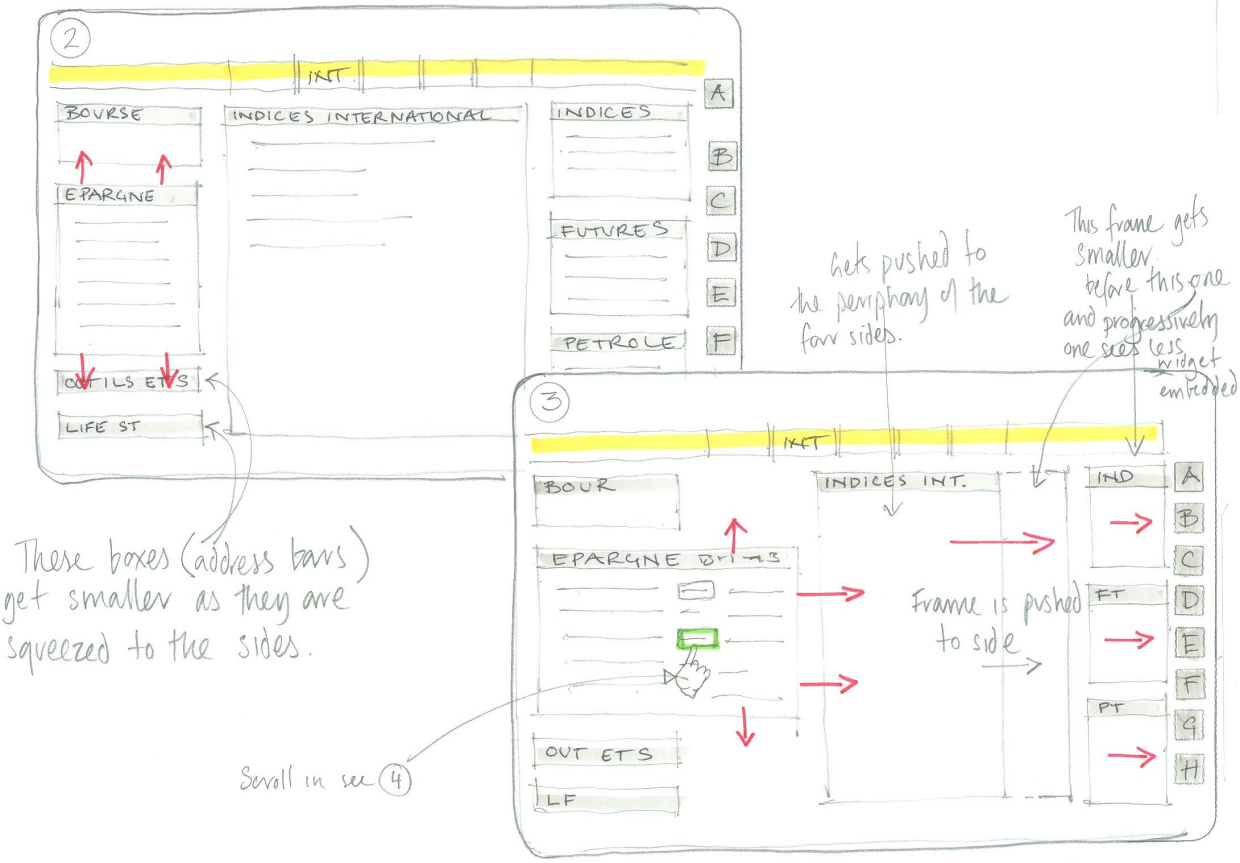


Figure 2



## 1.Preface

The good news is that the steps that are necessary for creating the ZDragon Desktop starts to materialize in my mind. And in my opinion, we can do it.   
The downside is:

* Many prototypes (PoC, Proof of Concept) are needed
* And some research, experimenting, feedback and correction-cycles are on the road
* These takes some time.

The whole concept is similar to the SphereBrowser (developed from 2006-06 till 2007-02 = 9 months, 2-3 guys), but simpler in some sense, because it is only 2D, however I expect the similar time frame, because this time we have a very concrete application (HQ Desktop) that should be built on the top of the ZDragon Desktop and ZDragon should handle various 2D Widgets (combobox, labels, buttons) (that was not the case in SphereBrowser).  
  
So, approx. 9 months can be a good estimation, but I will concretise a little bit further with an implementation plan in a later section.

Before starting a projects, it is worth taking note of the risks. What concerns me are the following:  
**Risk 1:**

1. The most efficient way to develop everything if one guy can concentrate and really focus on only 1 topic at a time. I am guy like this. Switching between task (2 days for this, 2 days for another completely different task) requires a couple of hours of “get-in” time/ warm-in time at every occasion. (every Monday after the weekend)  
   If all time, all days are spent on focusing only a single task, the mind can go deeper into the topic, new ideas appears more frequently; I even like the fact that I dream about the topic and on the morning I find out what could be a solution for a yesterday problem.   
   Another drawback of switching between tasks is that we lose this efficiency that will lengthen the project.
2. The development of HedgeQuant Desktop and the Portfolio selection strategies have to continue, because it generates money. I cannot really abandon the project. None of us can.

So, these are the ambivalent feelings of mine. I couldn’t see how we can insert the ZDragon Desktop project into our current projects, because it seems to be a long project on its own.

**Risk2:**

As always, if we make some something that has never been done, it contains some dangers that something is not possible with the chosen technology (WPF). For example writing an operation system cannot be implemented in Visual Basic.

**Risk3:**

“other challenges to make it **easily programmable** and smooth enough to interact.”

* Building the inner content of the windows – the inside of the ZWindows – is more problematic. I am still quite pessimistic that RAD (Rapid Application Development) is achievable in the ZDragon Framework. As a good example for RAD is developing a basic WPF application. Placing WPF widgets onto a static WPF window, where their positions and sizes are fixed. That is RAD. Microsoft worked a lot to make RAD development for WPF developers. They invested thousands of man-years into the project of .NET 3.5, WPF, Visual Studio 2008, etc. Leveraging on their work, ordinary developers can quickly design 2D dialog boxes that contains widgets. Just simple click here – or -- there, move the WPF label to the right place, drag&drop a combobox to the dialog, etc.  
    
  However, in our case the widgets can be moved or resized and animated. This just simply doesn’t fit into the RAD environment the Microsoft developed. On the other hand, in theory it is possible to develop a Tool that help in RAD for ZDragon, but we don’t want to develop a tool like a Visual Studio Dialog Designer, because of its time it requires.

So, we should keep an eye on this and make steps one at a time. For example, I plan to create a case study (PoC) for example only just for the current MarketSummary dialog. How can this small dialog be developed under the ZDragon framework.

In the next chapter I made an outlook on some technologies:

## SeaDragon and usage of it.

SeaDragon: The technology that only loads from the HDD what’s required as the user goes to that part of the overall image.

<http://labs.live.com/Seadragon.aspx>

It appeared sometime around 29th January 2005.

<http://www.metacafe.com/watch/637132/this_technology_will_blow_your_mind/>

2.1 One usage of SeaDragon is PhotoSynth  
  
PhotoSynth = powered by SeaDragon

You can install the tech preview of PhotoSynth: (it is not a productions quality)

<http://labs.live.com/photosynth/installer/sysreq.aspx?collection=sanmarco/index1.sxs&st=coll>

Internet Explorer is recommended. I played with that. There is a “More Collections” section. I tried “Grassi Lakes” but I wasn’t impressed with the user-navigation. It was smooth, but something wasn’t right. Quite difficult to explain.

2.2 Another usage of SeaDragon is DeepZoom

You can try the DeepZoom feature of SilverLight 2.0 (based on SeaDragon) in a production environment.

<http://memorabilia.hardrock.com/>

It is very nice to try. from here:  
<http://www.techcrunch.com/2008/03/05/seadragon-in-silverlight-the-coolest-bit-from-mix-keynote-one/>

“The demonstration featured the [Hard Rock Memorabilia[http://i.ixnp.com/images/v3.30/t.gif](http://memorabilia.hardrock.com/)](http://memorabilia.hardrock.com/) site. It started with what looked like some basic memorabilia shots, then zoomed out to a button on a suit. The seamless image was 2 billion pixels created from many separate images with Silverlight natively providing the stitching.

The crux of the functionality is to provide the ability to zoom in and out quickly without the need to download an entire picture; Silverlight only loads what’s required as the user goes to that part of the overall image, saving on bandwidth and in theory providing a quicker and more pleasurable end user experience. “

Or the BBC DeepZoom application:

<http://www.bbc.co.uk/radio1/bigweekend/2008/zoom/>

But Deep Zoom is designed for Silverlight, and still not available to WPF apps.  
  
There is a Deep Zoom Composer, that can merge individual jpeg images to a new file format called Deep Zoom Image. DeepZoom can handle only this file format.

I would like to emphasize that the main point of the SeaDragon technology is that the user can see a huge (10-50GB size, cannot fit into RAM ) image without loading the whole image into the RAM. However, we don’t plan to do the same, because it is quite difficult. In our ZDragon Desktop we plan to keep every widget in memory. The smart thing from us is we **only render** those that are visible enough.

## Other noteworthy programs

### 3.1 3D Cube for RSS reading

<http://www.microsoft.com/emea/msdn/thepanel/featured/universs.aspx>

You can even download it and try, but it is catastrophic to use and the when the text of RSS feeds is rendered is switch back to a 2D window. So, it is technically, not 3D.  
This is how it looks like, when the text of the RSS is rendered:

<http://blogs.msdn.com/llobo/archive/2006/12/06/3d-rss-reader-in-wpf.aspx>

### 3.2 Ubuntu Linux’s Compiz 3D Desktop

<http://www.youtube.com/watch?v=i9JC5NQ7G0o>

It looks very nice, but I have to comment that after installing Ubuntu linux (which is not trivial in its own), it takes approx. 2 days for an inexperienced guy to install the 3D Desktop. For an experienced one, maybe half day or just one day depends on the video card and your hardware. So it is not for the masses. Another note is: the cube is wired into the system. It cannot be another basic primitive. Of course, I understand why they did it. The cube contains only 6 squares. However, a smooth sphere can contain thousands of triangles. So, it is for performance reason. Another reason is: the mapping the 2D space of a desktop to a cube is easier than mapping 2D space to a sphere.

### BumpTop 3D Desktop Prototype by a Canadian university

<http://www.youtube.com/watch?v=M0ODskdEPnQ>  
from which we can gain some ideas. But for example implementing 3D shadow is not the easiest task.

## Features of ZDragon Desktop

Definitions:

We define ZWindow = ZUniverse = Zoomable Universe as a basic window in our framework. The ZWindow not necessarily has a border, a title bar, etc...The ZWindow is a 2D container that can contain :

* General 2D widgets (*support the IResizable interface*) (button, label, combobox, listbox, checkbox, vector and pixel images) . Painted into the background: layer 0, (class CustomMagnificationRule:  
  *“There are preference rules for text blocks inside a window when magnifying/minifying as well. For example: a Paragraph title will appear /disappear before/later the relevant paragraph text.”*)
* Nested ZWindows that can push/move/resize each other when zooming or dragging, painted to layer1 (class ToTheCenterMagnificationRule)
* TopZWindows, that are overlayed to on top of the other contents. Rendered to layer2 (class UniformMagnificationRule)

Features:

* built on top of the WPF (C#)
* The root UI (the top ZWindow) can be mapped to a 3D surface, so you can rotate it in 3D like in Flip3D feature of the Vista op.sys.  
  (I currently think that WPF will help in this. Research needed. WPF 3.5 sp1 may help.)
* The nested ZWindows can be moved and dragged inside its containers.
* It supports animation (when moving/dragging/zooming the ZWindows/ZUniverses, the transition of the original place and the target space is continuous in time. E.g. it happens in 0.5seconds.
* When double-clicking on the “sensitive area”(titlebar) of a regular ZWindow it will “move-animate” into the centre
* Supports 2D panning the UI like in DeepZoom demos and in PhotoSynth
* Transparent background where there is no content, so the underlying windows or the Vista Desktop is seen-through.

## Open Questions

*“When the mouse hovers on a window and the user scrolls with the mouse wheel the window is resized and moved towards the target space (that is the centre).”*   
I assume that when the user zooms in, the target position of the currentWindow is ”Always” the centre of the Containing universe. Does it?   
(Except the TopZWindows, that magnified in place, because TopZWindows don’t disturb the other windows)

*“Which virtual windows can be dragged? (for example, can the user drag a window if it is almost full-screen .”*

Yes  
*“ or only when it is 64x64 pixel size)”*

If the almost full-screen ZWindows can be dragged, what should happen? The other ZWindows are on the periphery, and they cannot move, because they already reached their minimum size.

## Implementation notes

* The TopZWindows will hide the content that is under them. (maybe they hide another ZWindows that are not top-windows)
* If a regular (not TopZWindow) ZWindows are dragged and moved, other ZWindows become smaller and move away. One idea is to implement the whole animation with spring theory. Not like this   
  <http://www.myphysicslab.com/molecule4.html>  
  , but this is just a draft idea
* Another animation idea is a the Physics animation (like in BumpTop 3D Desktop Prototype)
* Another animation idea is a deterministic animation computed by a mathematical equation
* Another animation idea is a deterministic pre-determined keyframe animation.

## Milestones and the required times

Currently ,we distinguish the following steps: PoC= Proof of Concepts  
  
PoC 0: (2 weeks)

* The current “HedgeQuant 3D background” (the spheres) can be on the background, but it is not compulsory to have a background like that
* Experiment: how can this 2D Universe mapped to a 3D plane, or a 3D cube
* Implement what happens when the zoom happens on the background (something similar to SeaDragon, it should magnify uniformly)
* Use static images (jpeg) in this PoC for ZWindows
* Can handle different screen-resolutions (1024x764 and 1920x1200)

PoC 1: (2 weeks + some (2?) weeks for feedback)

* Implement what happens when the zoom happens on a ZWindow. The ZWindow moves to the centre when zooming, at the same time, other Zwindows gets smaller and moves to the periphery
* because the Main (3D spheres) window is in the background, we can capture the mouse wheel scrolling even after the window is moved away from the mouse cursor. If the mouse scrolling event happens during the next 500-1000msec from the previous mouse scrolling, we magnify the window even further in spite of the fact that the mouse is not over the window.

PoC2: (2 weeks)

* animation support
* When double-clicking to the “sensitive area” of a regular ZWindow it will “animate” into the centre

PoC3: (1 week)

* How can regular 2D widgets (button, label, DataGrid, etc.) be placed inside the ZWindow
* Implement UniformMagnificationRule for them

PoC4: (1 week)

* Experiment with nested behaviour: the ZWindows can contain nested ZWindows (hierarchy depth is 2)
* When the mouse is over a nested ZWindow a highlight (what is it?) or a green border appears.

PoC5: (2 weeks + some (2?) weeks for feedback)

* Experiment with drag&drop the ZWindows. When moved by the user, the ZWindows push each other in that way that none of them overlap

PoC6: (1 week)

* The user can drag&drop a Zwindow from a nested ZWindow to an upper layer in the ZWindow hierarchy
* The hierarchy of the ZWindows are saved and loaded (versioning!)

PoC7: (2 weeks)

* Develop the MarketSummary dialog inside ZDragon
* Make conclusions based on the experience of developing it.
* Suggest modifications to the ZDragon framework.
* Implement the modifications.

PoC8: (4 weeks)

* The 2D TreeViewer which will be a usual 2D widget is implemented (sketch was given some months ago)

After the last PoC, the PoC phase is over. We are satisfied with the framework and confident that we can build applications on this ZDragon framework. So we use the whole ZDragon framework for HedgeQuant.

Total: 17 (+4?) weeks estimation for the PoC.

Each of the PoC can contain multiple breaks in the implementation. Feedbacks from you, brainstorming sessions, ideas, research into another direction. Back-steps are also possible. So, the times are only rough estimates. It estimates 1 person that is similar to me (George) and as a full-time. But, if I don’t develop anything in HQ from now, just team-managing time: 15%, some company administration: 15%, communication time: 10%, I can only regard my time as 60% of the full-time. At the first phases, which are critical for the viability of the project, I currently don’t think I can use Zoltan (how can I say without spite him: the lack of creativity).   
And by the way, putting 2 guys to the project doesn’t halve the implementation time as we know it well enough.

## 8.Conclusion:

That’s it for now. I expect obstacles on the road. Maybe there will be big problems we face when we get to PoC 5 for example, that cannot be seen now. But I see a good chance that eventually, it can be a success.