



An Introduction to JeeWiz for Business Managers

Brahm van Niekerk

Business Manager - *JeeWiz!*

Abstract

JeeWiz enables organisations to rapidly turn out enterprise systems. It removes many of the risks from implementing these systems. By enabling you to produce Microsoft .Net and Java or J2EE systems from the same set of specifications, it enables you to truly future-proof your system. JeeWiz not only takes care of up to 95% of the development effort in a secure environment, it also creates the basis for sustainable software systems for the next 10 years. This paper will show how this kind of development has become a reality in October 2003. It is assumed that the reader will have some background in having managed, budgeted or overseen substantial enterprise system development.

Introduction & Goals

The pressures put on business managers these days are unlike any in the past. Systems need to be produced in a few months on ever changing platforms. JeeWiz eases this process by creating a higher level of abstraction. By creating a specification that is *not* intimately tied to the technology platform, sustainable systems can be created on a range of technology platforms.

The following is a business manager’s introduction to JeeWiz that will flesh out

- **What is JeeWiz?**
- **What will it do for your business?**
- **What JeeWiz is not?**

The answers follow.

What is JeeWiz?

JeeWiz is an MDA process tool. Model Driven Architecture (MDA) is described in a separate article. JeeWiz shares many of the goals of MDA and surpasses many of them at the time of writing.

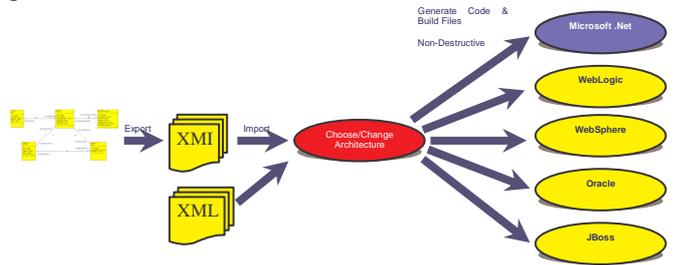
JeeWiz has the following stated goals:

- Improve Application Portability – deploy the same application on both Microsoft and IBM platforms.
- Promote platform independent applications. This is critical in organisations that are part of a group that needs to operate with systems in other members of the group.
- Enable business experts to operate across multiple technology domains. This means that we no longer need to combine business and technology expertise in the designers and developers we use.
- Improve productivity while still enabling full fine-grained control. Many products in the past have claimed to raise the productivity, but they prevent us from doing the fine-grained changes needed to accomplish our aims. JeeWiz is different in this.

In a nutshell, JeeWiz enables you to design an application in a technology platform independent manner. This means that these technology agnostic specifications are used to generate an executable application in a range of deployment environments. JeeWiz enables technology specialists to work independently from the business experts to generate the application.

High level overview

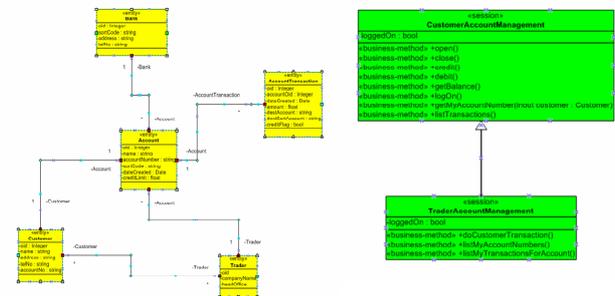
The operation of JeeWiz is graphically illustrated below. By taking either UML or XML as a specification, JeeWiz generates one or more system(s) on different technology platforms.



Specifications or models

JeeWiz accepts as input a number of different forms of input. The following tools can all be used to create the specification or model*. Since natural language, such as English, is ambiguous in defining specifications, it is unsuitable for defining the limits of a system. If we could create a “template”, then we could enable these business experts to express the requirements unambiguously. UML and XML is such a “template”. Using either a graphical UML model or a more precise XML structure, we create an unambiguous template for expressing requirements. Here is an example of both

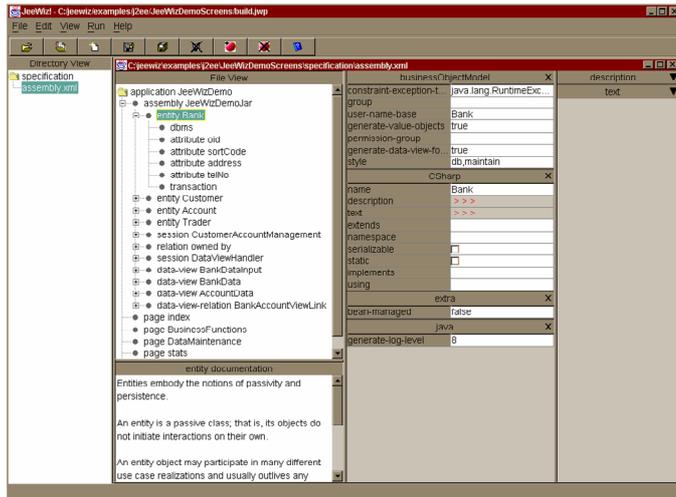
UML Model



The model above shows us an extract of the UML model. By enabling us to graphically link different entities, data elements, we can indicate for example that a customer can have several bank accounts and that many accounts will be linked to each branch.

* A model is a scaled down version of the real system. A specification on the other hand should be unambiguous definition of what is required.

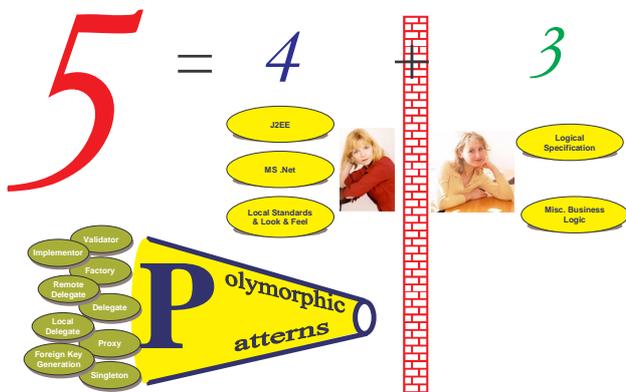
XML Specification



JeeWiz includes an editor that enables you to define specifications of the application in XML. Every property is accompanied by help text to enable you to see the impact of the property.

How is it different from other systems

JeeWiz creates a clear separation between technical and business experts. By allowing each of these expert groups to concentrate on their relevant areas of expertise, we actually accelerate the whole development process.



In contrast to this, many of the so-called Rapid Application Development (RAD) tools simply convey the same complexity that already exists from code to modeling. This is what is known as the *law of containment of misery*. In other words, we transfer the pain or misery we experience in complex coding to the modelling tool, but all the time maintaining the same amount of pain or misery. This kind of modelling now ceases to clarify anything due to the complexity and the value of simple modelling is lost.

The “Polymorphic Patterns” in the picture are the complex patterns that are used in the guts of JeeWiz to generate complex systems from simple specifications such as those shown in **UML Model** and **XML Specification**. This complexity is totally invisible to the business expert.

What will it do for your business?

Many so-called “cutting edge” business applications these days tend to be out of date before they are completed. As

soon as an enterprise system has been completed, the technology used to create it is outdated, flawed. What we need is a way to potentially integrate new technologies without destabilising the application all together. This is precisely what JeeWiz offers.

The benefits can be summarised as follows:

- **Reduce costs.** We do this by
 - Simplifying the specification of systems.
 - Reduce the number of developers needed by simplifying the whole development process.
- **Reduce development time.** Because we can rapidly create prototypes, we reduce the time it takes to produce software. Reduced time equates to reduced risks and costs.
- **Increase application quality.** Since up to 95% of the system is automatically generated, the quality of this code is much better than handwriting all the code. Better quality equates to far fewer technology bug fixes.
- **Improve the return on IT investment.** Since the generated systems can be modified by your own technology experts to adapt to legacy libraries or expand to include more channels, your investment in IT is protected. You can actually keep your systems up to date, technologically speaking.
- **Rapidly include new technologies.** It has been said that the only constant in IT, is that it will change. JeeWiz allows you to integrate the new technologies – many of which we do not even know about yet.

What JeeWiz is NOT?

JeeWiz has been compared to many other systems, all in an attempt to make sense of this revolutionary technology. In describing it, it sometimes helps to know what it is not.

NOT: Programming without coding

This was a popular mantra in the nineties. JeeWiz does not remove coding. We will always need coders. JeeWiz simply allows us to generate the boring repetitive code quicker. This frees our coders up to focus on the areas that really add value to our business – the complex business rules.

NOT: Rapid application development (RAD) Tool

Although the promises made by RAD have been far reaching, it has largely failed to deliver. RAD takes the complexity in enterprise systems and enables us to make our applications complicated quickly. This really simply perpetuates the *law of containment of misery*. We take the misery of coding complex systems and convert them into a different shape or form, such as UML.

JeeWiz actually simplifies the development of complex systems by working from platform neutral specifications. Although JeeWiz accelerates software development, it actually makes it simpler to specify the system.

NOT: UML to code converters

Most of the UML tools that make it to market will convert UML to code. However, to design the complex software

needed for secure, reliable systems require us to design complex UML. What have we done? We have simply transferred the misery of coding to UML without simplifying anything.

JeeWiz works from simple unambiguous technology agnostic specifications. The complexity is contained under the covers of JeeWiz, not in your face.

NOT: Code level Abstraction

This is similar to the previous point. Microsoft Visual Basic was one of the first MS Windows style programs to enable one to “program visually”. For simple systems this worked fine, but if you wanted to move from one version of Visual Basic to another or from Visual Basic to C++ or Java you had to rewrite the systems. We have simply transferred the misery of creating a static system from coding to visual point and click. Same effort needs to be applied or perhaps even more.

JeeWiz simplifies the development process and puts up a wall between technical and business experts.

NOT: Profiled UML

One of the developments in UML is to create “profiles”. A profile will have all the needed properties for you to set, for example EJB. We are once again simply transferring the misery without making any gains in productivity.

Although JeeWiz can be configured to expose the needed *platform independent* properties using profiles, the complexity is still kept to a minimum. If it takes the same amount of time to create the specification as it takes to write the entire system by hand, then we will have failed in our

purpose. It can therefore be stated that JeeWiz goes much further than the tools currently exposed by UML editors.

NOT: Reversed engineered legacy code

Many UML tools, such as Rational Rose, allow us to take a large body of legacy code and reverse engineer it back to UML. These models are horribly complex and frequently not understandable.

JeeWiz simplifies the development process and gives us tangible executable results within a short period of time.

Conclusions

As we have seen, JeeWiz certainly hails a revolution in the development of software. By using it we are able to build technology systems economically viable even in first world countries. By simplifying the process and separating the efforts of technology and business experts, we can maximise the efforts of both of these groups. No, we would not have to fear new technology, but rather reach out for it. The question in future will be: “How quickly can we convert JeeWiz to use it on my systems?”

About the author

Brahm van Niekerk has been active in the software industry for over 15 years. Most of this time was spent doing both enterprise architecture as well as hands-on development. After running his own Software House for 10 years in South Africa, he moved to London, UK where he has been consulting since 2000. He now works for NT/e where he is involved with JeeWiz.