

## Convenience = Efficiency

The PC continues its triumphant march under the flag of "more functionality for everyone." Front-end integration of legacy systems with desktop applications is closing the gap with the host system-or the database at the back-end.

But how does more functionality add up to greater productivity?

PC applications have grown closer together with such interfaces as OLE, DDE, and ActiveX, and the HLL API has even made it possible to reach host systems in the back end. Nonetheless, tools created by individual users long dominated the desktop, and the topic of front-end integration received only moderate attention in IT concepts.

**But it is just that perceived gap in ease-of-use that is proving to be the main factor holding back efficiency.**

### Fundamental questions

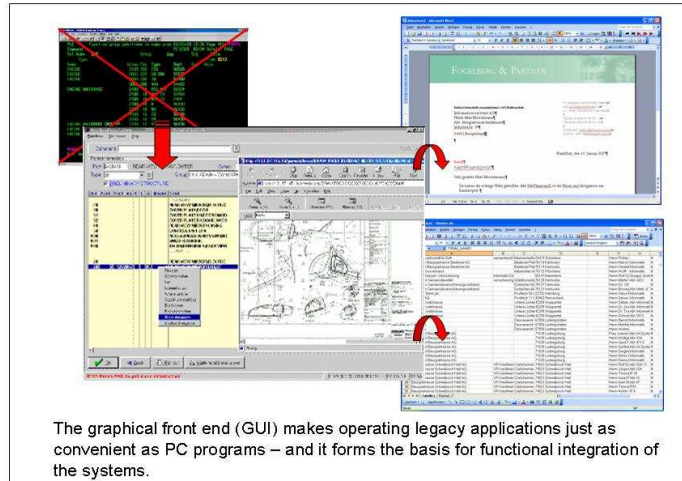
- How does functionality turn into productivity?
- Is back-end integration always the better solution?
- Where is the happy medium between central requirements and individual needs?

### From the paper interface...

Dealing with programmatic interfaces presents a challenge in linking the "old" world of the mainframe with the "new" world of client desktop applications. In contrast to the *paper interface*, "copy and paste" at least provided a rudimentary solution to the problem of manually importing data.

### ...to the clipboard...

Other procedures such as copying data into a temporary file as well as subsequent reformatting and importing into the target environment are time-consuming and inefficient. Real-time integration solutions relegate such procedures to the past.



The graphical front end (GUI) makes operating legacy applications just as convenient as PC programs – and it forms the basis for functional integration of the systems.

### ...to an integration solution

The stated aim is a professional tool customized for individual users' usage patterns, and the accompanying clean separation with data-management systems and business logic at the back end and the presentation layer at the front end is proving to be extremely practical.

### Front-end integration with NetPhantomX

Form letters in Microsoft Word with data from legacy systems? No problem! NetPhantomX makes data from the host system available on the client. OLE calls need only a few lines of Java code and save time and reduce errors for users.

**Flexible interface implementation between different clients and local desktop programs keeps costs manageable, and the perceived benefit is even greater.**

Taking this one step further is the integration of the graphical interface: Administrators who often must juggle countless client applications at the same time find working "under one roof" is substantially clearer and more efficient.

Two different approaches have proven themselves in practice:

- Several applications are combined in the NetPhantomX client window.
- An overall framework integrates the NetPhantomX GUI or even the NetPhantomX terminal component.

The NetPhantomX GUI is therefore not only a graphical wrapper for host systems, but it likewise allows the integration of additional Swing components, like exporting an individual GUI in Java frameworks.

Another aspect is client-side data exchange. These tasks can be handled easily using NetPhantomX APIs, standard Java APIs (e.g., JMS, JAX, or sockets), or third-party APIs (e.g., jacoZoom or HSSF to communicate with MS Office).

**The GUI is far more than just the face of the applications. For administrators, it is the intuitive access point to functionality and information-the link between business processes and IT systems. Here convenience and efficiency go hand in hand. NetPhantomX is a valuable asset with long-term relevance, even when other systems are operating in the background.**

