



The Interoperability Stack

an overview of
how AT and IT interact

Peter Korn

Accessibility Architect

Sun Microsystems, Inc.



What AT needs from IT

- For magnification, speech, and/or Braille; and to automatically read & highlight content, AT needs:
 - > To know where all of the user interface elements are
 - > Know what UI elements are (radio button, checkbox, ...)
 - > Know where the text is, what the text is, text attributes
 - > Know where the focus is, the text caret, text selection
 - > Know how elements related to each other (labels, etc.)
 - > Know the precise text content location, to highlight it
 - > Know document content structure for rich content access
 - > Have final control over what is drawn to the screen

What AT needs from IT (cont.)

- To replace user input via speech, switches, head-mouse/eye-gaze, AT needs:
 - > Ability to control all the UI elements
 - > Ability to synthesize keyboard & mouse input
 - > Ability to enter/delete/replace text content
- Additional, general AT needs:
 - > Ability to interdict keyboard & mouse input
 - > Sufficient audio control/priority to not be pre-empted
 - > General I/O services (USB switches, Braille devices, etc.)

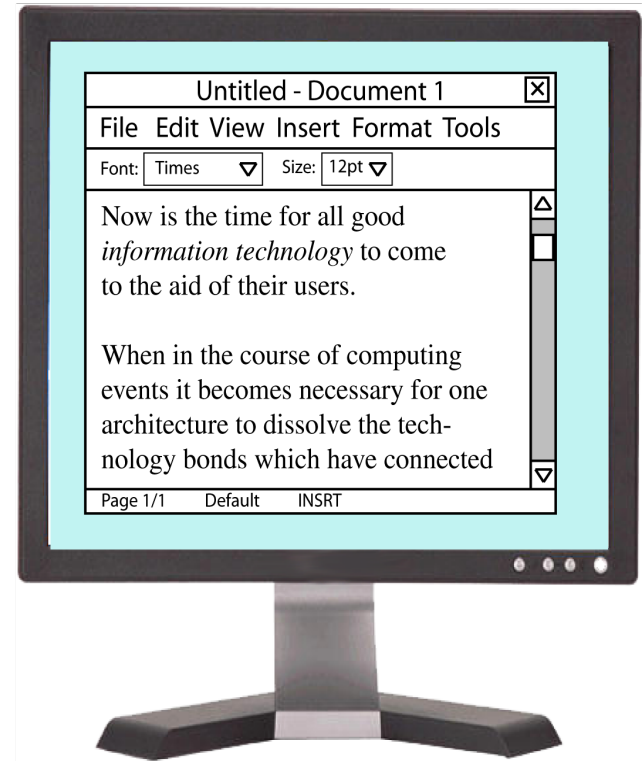
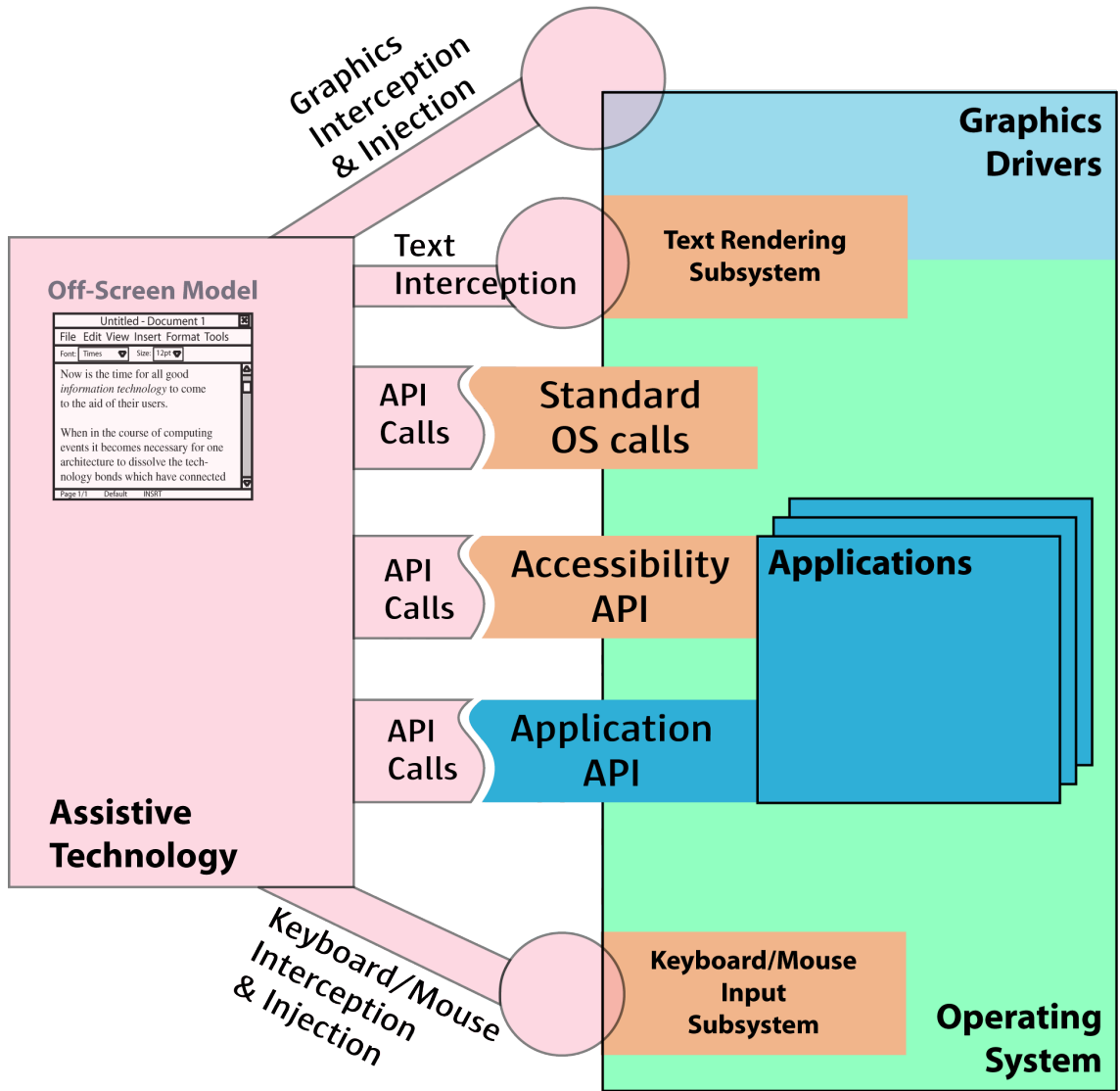
How AT gets what it needs

- To get needed information about screen contents:
 - > Replace/chain/patch graphics subsystem, in order to:
 - > Build an Off-screen model representing meaning of GUI
and/or
 - > Use general system APIs, app-specific interfaces, and/or Accessibility APIs, to obtain the information directly
- To control keyboard/mouse input:
 - > Replace/chain keyboard driver
and/or
 - > Use supported services/APIs to obtain that control
 - > Use app-specific interfaces, Accessibility APIs to manipulate user interface elements programmatically

How AT gets what it needs (cont.)

- To magnify the screen:
 - > Use the replacement/chained video driver to control what is drawn (and maybe turn off some IT drawing options)
and/or
 - > Use system-supplied magnification services
- To provide rich document content access:
 - > Re-interpret the raw document format
and/or
 - > Hook into the application with app-unique API calls
and/or
 - > Use Accessibility APIs to content that the app(s) support

Interoperability Stack



What is an API?

- Acronym stands for “Application Programming Interface”
- A contract between two different pieces of software, expressed as a set of function or method calls
 - > Another view: a set of questions that can be asked, or requests that can be made
- Example of an individual call in an API:

```
int err = Open(String name, FILE &file)
```

Promise to open the file “name”, place a reference to it in “file”, and return a success/error code

What is an Accessibility API?

- Set of calls designed specifically to provide AT with information they need to make applications accessible
- Encompass questions like:
 - > What objects are in this window?
 - > What is the role of this particular object?
 - > What are the text contents of this particular object?
- Also include “events” - notification of changes:
 - > The focus has moved to this new object!
 - > The caret is in a new location!

What is the Accessibility API's role?

- AT today uses many techniques to obtain the information needed to provide access:
 - > Make standard system calls for things that help them
 - > Patch/replace/chain video drivers to build Off-Screen Models of the contents of the screen; keyboard drivers in order to redirect keyboard control
 - > Use application-specific APIs to get information from specific applications
- Accessibility APIs replace some/most/all of these with supported programming interfaces
 - > Some Accessibility APIs may be comprehensive enough to replace the need for an Off-Screen model



The Interoperability Stack

an overview of
how AT and IT interact

Peter Korn

Accessibility Architect

Sun Microsystems, Inc.

