TECHNICAL REPORT

ISO/IEC TR 13066-4

First edition
2015-11-01

Information technology — Interoperability with assistive technology (AT) —

Part 4: Linux/UNIX graphical environments accessibility API

Technologies de l'information — Interopérabilité avec les technologies d'assistance —

Partie 4: Accessibilité API des environnements graphiques lInux/UNIX

Reference number
ISO/IEC TR 13066-4:2015(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 35, *User interfaces*.

ISO/IEC 13066 consists of the following parts, under the general title *Information technology — Interoperability with Assistive Technology (AT)*:

— *Part 1: Requirements and recommendations for interoperability*

— *Part 2: Windows accessibility application programming interface (API)* [Technical Report]

— *Part 3: IAccessible2 accessibility application programming interface (API)* [Technical Report]

— *Part 4: Linux/UNIX graphical environments accessibility API* [Technical Report]

— *Part 6: Java accessibility application programming interface (API)* [Technical Report]

Introduction

Assistive technology (AT) is specialized information technology (IT) hardware or software that is added to or incorporated within a system that increases accessibility for an individual. In other words, it is special purpose IT that interoperates with another IT product enabling a person with a disability to use the IT product.

Interoperability involves the ability to add or replace Assistive Technology (AT) to existing components of Information Technology (IT) systems. Interoperability between AT and IT is best facilitated via the use of standardized, public interfaces for all IT components.

This part of ISO/IEC 13066 describes the following.

AT-SPI

The Assistive Technology Service Provider Interface (AT-SPI) API, which can be used as a toolkit agnostic framework to support software to software IT-AT interoperability on Linux and UNIX graphical desktop environments.

ATK

The Accessibility Toolkit (ATK) library provides a set of interfaces in support of AT-SPI on the GUI application side. The interfaces are toolkit-independent implementations could be written for any widget set, such as GTK, Motif, or Qt.

Information technology — Interoperability with assistive technology (AT) —

Part 4:
Linux/UNIX graphical environments accessibility API

1 Scope

This part of ISO/IEC 13066 provides an overview to the structure and terminology of the Linux/UNIX graphical environments accessibility API.

It will provide the following:

— a description of the overall architecture and terminology of the API;

— further introductory explanations regarding the content and use of the API beyond those found in ISO/IEC 13066-1:2011, Annex A;

— an overview of the main properties, including
  — of user interface elements,
  — of how to get and set focus, and
  — of communication mechanisms in the API;

— a discussion of design considerations for the API (e.g. pointers to external sources of information on accessibility guidance related to using the API);

— information on extending the API (and where this is appropriate);

— an introduction to the programming interface of the API (including pointers to external sources of information).

It will provide this information as an introduction to the Java API to assist the following:

— IT system level developers who create custom controls and/or interface to them;

— AT developers involved in programming “hardware to software” and “software to software” interactions.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

2.1
A11Y
short form of “accessibility” or of “accessible”

2.2
accessible object
part of the *user interface* (2.21) that is accessible by and exposes the Java accessibility API

Note 1 to entry: An accessible object is represented by an object of the “AccessibleContext” Java class.

2.3
application programming interface
API
collection of invocation methods and associated parameters used by one piece of software to request actions from another piece of software

2.4
application software
software that is specific to the solution of an application problem

EXAMPLE A spreadsheet program is application software.

2.5
assistive technology
AT
hardware or software that is added to or incorporated within a system that increases accessibility for an individual

EXAMPLE Braille displays, screen readers, screen magnification software, and eye tracking devices are assistive technologies.

[SOURCE: ISO 9241-171:2008, 3.5]

Note 1 to entry: Within this part of ISO/IEC 13066, where Assistive Technology (AT) is used, it is to be considered as both singular and plural, without distinction. If it is to be used in the singular only, it will be preceded by the article "an" (i.e. an Assistive Technology). If it is to be used in the plural only, it will be preceded by the adjective "multiple" (i.e. multiple AT).

2.6
ATK
The Accessibility Toolkit
library which describes a set of interfaces that support the AT-SPI on the GUI application side

Note 1 to entry: The ATK interfaces are toolkit-independent - implementations could be written for any widget set, such as GTK, Motif or Qt.

2.7
AtkObject
base object class for the Accessibility Toolkit API

Note 1 to entry: This is the ATK analog of MSAA's and IAccessible2's "accessible object". The AtkObject structure is not accessed directly.

2.8
AT-SPI
The Assistive Technology Service Provider Interface
API that can be used as a toolkit agnostic framework to support software to software IT-AT interoperability on Linux and UNIX graphical desktop environments

2.9
clients
components that use the services of another component

Note 1 to entry: In this part of ISO/IEC 13066, client refers more specifically to a component that uses the services of either or both AT-SPI and/or ATK to access, identify, or manipulate the UI elements of an application.

2.10
daemon
software application that is not invoked explicitly, but lies dormant waiting for some condition(s) to occur

2.11
embedded device
embedded system
computer system designed to perform one or a few dedicated functions often with real-time computing constraints

Note 1 to entry: It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Note 2 to entry: In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability, e.g. hand-held computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded," large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions," and is thus appropriate to call "embedded."

2.12
function
defined objective or characteristic action of a system or component, e.g. a system has inventory control as its primary function

[SOURCE: IEEE Std. 610.12-1990]

2.13
interface

shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical interconnections, signal exchanges, and other characteristics, as appropriate

2.14

interoperability
capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

2.15
inter-process communication
IPC
mechanism by which different software processes communicate with each other – across process boundaries, runtime environments, and sometimes also computers and *operating systems* (2.16)

2.16
operating system
OS
software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input-output control, and data management

Note 1 to entry: Although operating systems are predominantly software, partial hardware implementations are possible.

2.17
servers
in the context of this part of ISO/IEC 13066 and of assistive technology, servers are components (applications, libraries, etc.) that have UI and expose information about the UI and/or allow it to be manipulated

2.18
service
functionality made available to a user electronically, e.g. airline reservation service, currency translation services, weather forecasting, restaurant recommendations are all services

[SOURCE: ISO/IEC 24752-1:2014, 4.27]

2.19
software
all or part of the programs, procedures, rules, and associated documentation of an information processing system

Note 1 to entry: Software is an intellectual creation that is independent of the medium on which it is recorded.

2.20
system software
platform software
application-independent software that supports the running of *application software* (2.4), e.g. an *operating system* (2.16), a Web browser, or a programming environment – Java can be used as a platform for application software

2.21
user interface
UI
mechanisms by which a person interacts with a computer system

Note 1 to entry: The user interface provides input mechanisms, allowing users to manipulate a system. It also provides output mechanisms, allowing the system to produce the effects of the users' manipulation.

2.22
user interface element
user interface object
user interface component
entity of the user interface that is presented to the user by the software

[SOURCE: ISO 9241-171:2008, 3.38]

Note 1 to entry: User interface elements may or may not be interactive.

Note 2 to entry: Both entities relevant to the task and entities of the user interface are regarded as user interface elements. Different user interface element types are text, graphics, and controls. A user interface element may be a representation or an interaction mechanism for a task object (such as a letter, a sales order, electronic parts, or a wiring diagram) or a system object (such as a printer, hard disk, or network connection). It may be possible for the user to directly manipulate some of these user interface elements.

EXAMPLE 1 User interface elements in a graphical user interface include such things as basic objects (such as window title bars, menu items, push buttons, image maps, and editable text fields) or containers (such as windows, grouping boxes, menu bars, menus, groups of mutually-exclusive option buttons, and compound images that are made up of several smaller images).

EXAMPLE 2 User interface elements in an audio user interface include such things as menus, menu items, messages, and action prompts.

EXAMPLE 3 User interface elements in tactile interfaces include such things as tactile dots, tactile bars, surfaces, knobs, and grips.

3 Overview

3.1 General description

AT-SPI/ATK was originally developed by Sun Microsystems for the GNOME platform, a common and accessible graphical desktop for Linux / UNIX graphical environments. GNOME is an open source project delivering a collection of software libraries and applications. It was formerly rationalized as an acronym meaning GNU Network Object Model Environment. From the beginning, AT-SPI/ATK has served as a platform-neutral, toolkit agnostic framework for providing bi-directional communication between AT and applications. Through the use of AT-SPI, an application's components' state, property, and role information is communicated directly to the end user's AT, thereby facilitating bi-directional (input and output) user interactivity with, and control over, an application or compound document instance.

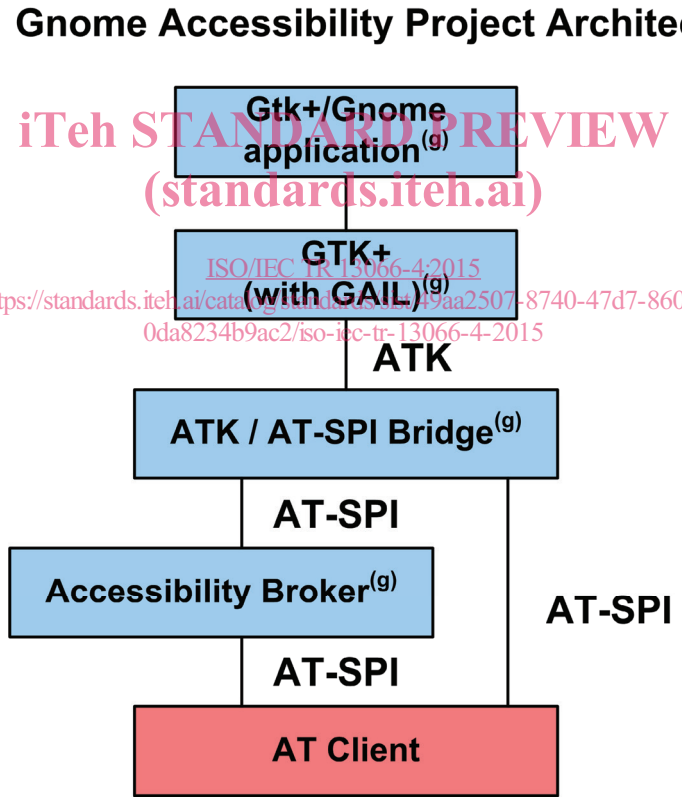
It includes support for rich text, tables, and relationships between objects, self-describing actions, application-specific information, and extensible object properties to support Web 2.0 applications.

As of this writing, both the GNOME and KDE platforms directly support AT-SPI/ATK on Linux/UNIX graphical desktops. For applications, support is available through several specific user interface toolkits, including GTK+ (a library for creating graphical user interfaces which works on many UNIX-like platforms, Windows, and on framebuffer devices), UNO, XUL, and Java/Swing (as described in ISO/IEC TR 13066-6). It can also be achieved by implementing support for ATK or the Java Accessibility API directly, or by AT-SPI directly.

3.2 Architecture

Accessibility support on Linux/UNIX graphical desktops implements the familiar client-server paradigm. The GNOME Accessibility Architecture provided on most Linux and UNIX graphical desktops -the only accessibility architecture available for Linux and UNIX graphical desktops as of this writing- distinguishes among ATK aware applications, assistive technologies and an accessibility broker.

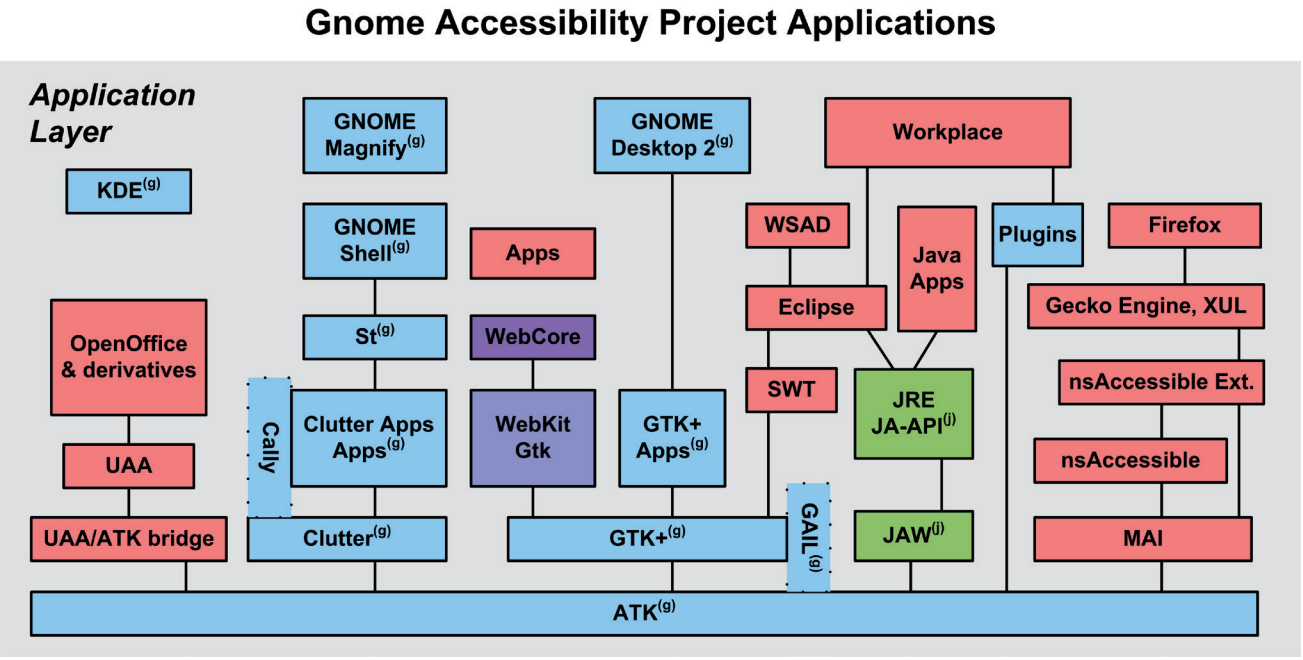
Communications between applications and assistive technologies (AT) is achieved using AT-SPI, which facilitates communications with ATK aware applications on the one hand, and with AT on the other. Generally, only AT is directly aware of AT-SPI.



Key

(g) GNOME

Figure 1 — AT-SPI/ATK Architecture Overview



Key

(g) GNOME

(j) Java; everything else is Application

Figure 2 — Application Toolkits Supporting AT-SPI/ATK

3.2.1 ATK Aware Toolkits

ATK aware applications are applications that offer information about their user interface via the AT-SPI protocol. This can be achieved in several ways.

— GNOME applications get AT-SPI support through the GTK+ toolkit they are based on. GTK+ optionally loads theGNOMEAccessibility Implementation Library (GAIL), an implementation of the accessibility interfaces defined by ATK, which bridges between the GNOME widgets and ATK. GAIL is provided by GTK in libgail-util. A second library is used in order to bridge between ATK and AT-SPI.