Table of Contents

- A short history of the project.
- SIP Communicator – existing features.
- Work in progress
- The SIP communicator architecture
short history
Work on SIP Communicator started in early 2003. It was initially one of the examples in the NIST-SIP open source stack.

June 2003 it moved to java.net for a life of its own where it is still being hosted.

2004 and 2005 it didn’t change a lot. Most changes were bug fixes. The application was mostly used by SIP (research) developers for testing purposes.

November 2005 we started work on SIP Communicator 1.0 which was a completely rearchitectured version of the old one. The current version is a lot more user friendly and designed for daily usage.

December 30th 2006 – first official 1.0-alpha1 release.

November 2nd 2007 – 1.0-alpha2 release.
Progress during the last 2 years

/trunk: Lines of Code

Date

Lines

0 25 000 50 000 75 000 100 000 125 000 150 000 175 000 200 000 225 000 250 000
Progress during the last 2 years
According to ohloh

**SIP Communicator**

Large, active development team

Over the past twelve months, **10 developers** contributed new code to SIP Communicator.

This is a relatively large team, putting this project among the top 10% of all project teams on Ohloh.

For this measurement, Ohloh considered only recent changes to the code. Over the entire history of the project, 13 developers have contributed.
JAVA.NET Top Ranked Projects

October 2007

This listing will be updated monthly. Each project that is #1 for any category that month gets a gold star for their accomplishments. If your project is not listed here and you are interested in getting listed, check out our publicize your project page for ideas of things you can do to be a gold star project too! You can also view our archive of previously top ranked projects.

Top 10 projects by membership:
1. jdk
2. glassfish
3. java-net
4. sip-communicator
5. lg3d
6. lg3d-core
7. open-esb
8. jain-sip
9. OpenJFX
10. opensoo

Top 10 projects by accesses:
1. glassfish
2. appfuse
3. OpenJFX
4. jaxb
5. jxta
6. JDK6
7. jax-ws
8. ajax
9. blueprints
10. glassfish-theme

Top 10 projects by cvs commits:
1. jade
2. jaxb2-reflection
3. hudson
4. glassfish-corba
5. open-jbi-components
6. glassfish
7. opensso
8. sailfin
9. bean-properties
10. jdnc-incubator

4th largest community out of 4,448 projects
Google Summer of Code 2007

- Google Summer of Code™ is a program that offers student developers stipends to write code for various open source projects.

- Google works with several open source, free software and technology-related groups to identify and fund several projects over a three month period. [http://code.google.com/soc/](http://code.google.com/soc/)

- In 2007 SIP Communicator was selected for participation.

- We received:
  - 8 student slots
  - Funding in the amount of **$40 000**
    - as much as Mozilla and OpenOffice
    - more than VideoLAN (VLC), Fedora, X.org
  - GSoC gave us 50 000 lines of code and a lot of publicity
existing features
SIP Communicator is an open source (LGPL) Audio/Video software phone and instant messenger. Among others, we currently support:

- **Audio calls with SIP and Jabber, and Video calls with SIP**
- Instant messaging with: *Jabber / Google Talk, ICQ/AIM, Yahoo! Messenger, SIP, Bonjour, Microsoft MSN, IRC, and RSS*
- IPv6 support for SIP and Jabber
- Support for multiple accounts and meta contacts
- Basic NAT & Firewall Traversal with STUN.
- Modularity, extensibility, and flexibility with OSGi
- Platform specific installers for Windows, Debian, Fedora, Mac OS X

Visit [http://sip-communicator.org](http://sip-communicator.org) for more details.
SIP Communicator Overview
SIP Communicator Overview

Instant Messaging History

![Instant Messaging History Image]

Wednesday, November 21, 2007

JRES 2007, Strasbourg
Emil Ivov, SIP Communicator
SIP Communicator Overview

Multi Party Chat Conferencing

[Image of SIP Communicator interface]

Subject: Forum for the SIP Communicator project, the OpenSource

Nov 05, 2007 damencho at 10:33:32
yes
Nov 05, 2007 rweires at 10:33:53
ok, fine
Nov 05, 2007 rweires at 10:33:55
thanks
Nov 05, 2007 damencho at 10:33:59

np
features currently in progress
Features Currently in Progress

- **Powerful and scalable firewall support** – Combine technologies like TURN, STUN, ICE and P2P in order to achieve powerful, secure, and scalable firewall support.

- **Security** – Encrypt all media. Usage of P2P for firewall traversal imposes reliable security of all media transmitted by the SIP Communicator in order to prevent relaying nodes from eavesdropping.

- **Automatic updates and one-click plug-in installation** - Provide a user interface and an online repository for SIP Communicator plug-ins.

- **Shared Whiteboards** – Use Jabber’s XEP-0113 – Simple Whiteboarding. Using SVG for collaborative work with Jabber.

- **Robust and seamless IPv6 support** – We aim to achieve more than a mere “support for IPv6”. We would like to guarantee its transparent and seamless usage. In other words, usage of IPv6 should not require any special configuration and should not cause any problems.
Shared Whiteboards with Jabber

1) GPS coordinates
2) 
3) ...
The SIP Communicator Plug-in Architecture
A Look Inside – Modularity & Flexibility

- Jabber
- SIP
- MSN
- Media
- User Interface
- Message History
- Meta Contact List
- Firewall Traversal
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** - All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** - It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** - The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** – All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** – It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** – All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** – It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- Ease of maintenance and deployment – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator
Application Design

Abstract APIs

Protocol Service
Jabber Impl

History Impl

Msg History

Stats Plug-in

OSGi

Some Plug-in

Protocol Service
SIP (jain-sip-ri)

Media Service
JMF

UI Service
Java Swing

Bundle
Specifics

JRES 2007, Strasbourg
Emil Ivov, SIP Communicator
SIP Communicator Architecture
A Simple Scenario

Dude (Online) _ o x
Hi dude, what r u up to?

Send

UserInterfaceService
Swing UI Service I mpl Bundle

ProtocolProviderService
Jabber Protocol I mpl Bundle

network

1 actionPerfomed(evt)
2 getService(providerRef)
SIP Communicator Architecture
A Simple Scenario

Dude (Online) _ o x

Hi dude, what r u up to?

Send

1 `actionPerformed(evt)`

Swing UI Service Impl Bundle

UserInterfaceService

2 `getService(providerRef)`

ProtocolProviderService

Jabber Protocol Impl Bundle

3 `sendMsgTo(Contact)`

network

4 `send message`
SIP Communicator Architecture
A Slightlly More Complex Scenario (1)

SIP Communicator          _ o x          CallTransfer Plug-In

network

OSGi Framework
SIP Communicator Architecture
A Slightly More Complex Scenario (1)

```
SIP Communicator          O x

1 getService(uiServiceRef)

2 registerComponent(this)

CallTransfer Plug-In

UserService

Transfer

3 container.addComponent(button)
    container.addComponent(field)

Swing UI Service! mpl Bundle

network

OSGi Framework

JRES 2007, Strasbourg
Emil Ivov, SIP Communicator
SIP Communicator 1.0 Architecture
A Slightly More Complex Scenario (2)

SIP Communicator

network

OSGi Framework

1. `getService(providerRef)`
2. `registerCallListener(this)`

User Interface Service

Swing UI Service Implementation Bundle

Protocol Provider Service

SIP Implementer Bundle

Transfer
A Slightly More Complex Scenario (3)

**SIP Communicator Architecture**

**CallTransfer Plug-In**

**UserInterfaceService**

**ProtocolProviderService**

**Swing UIService mpl Bundle**

**Network**

**OSGi Framework**

1. **INVITE req received**

2. **incomingCallReceived( evt )**

3. **container.addComponent(cmp)**

Alerting … The Duke is calling you!

Transfer
SIP Communicator 1.0 Architecture
A Slighty More Complex Scenario (4)

SIP Communicator

Alerting …
The Duke is calling you!

+3592166166
Transfer

5 transferCall(call, dst)

CallTransfer Plug-In

UserInterfaceService

Swing UI Service mpl Bundle

4 actionPerformed(evt)

ProtocolProviderService

S SIP Implementation Bundle

network

6 send REFER req

OSGi Framework