AUSTRIAN ACADEMY OF SCIENCES

ACOUSTICS RESEARCH INSTITUTE

Piotr Majdak

Open Source and Licensing: Publish your code – it’s good enough

AI Winter School 2025

Open source?

- A methodology to provide free access to a product's design and implementation details.
  - Open as in free, open as in access
  - Design and implementation details:
    - Source code
    - Data
    - Literature
    - Graphics
    - Audio
    - Hardware
    - Buildings

History of the “open source” idea

- pre 1950s: Source code released with the hardware, used mostly by academic researchers, modifications required
- 1953: A-2 system (UNIVAC): Users were invited to send back the improvements
- Late 1950s: IBM mainframe: Software exchange groups
- 1969: “Request for Comments” by researchers within the ARPANET
- 1970s: Software costs increased
  ⟶ Begin of closed- and/or restricted-source software
  - 1976: Microsoft (Altair BASIC)
  - 1979: AT&T (Unix V7)
  - 1983: IBM (PC-XT)
- 1980s: Sources listed in magazines and books (e.g., ATARI OS)

IBM 704 mainframe at NACA in 1957, Wikipedia (2025)

History of “open source” idea

- 1983: Richard Stallman (MIT, USA)

History of “open source” idea

- 1983: Richard Stallman (MIT, USA): GNU Project
- 1985: Free Software Foundation
- 1989: GNU General Public License (GPL) published
- 1991: Linus Torvalds: Linux kernel released
- 1993: FreeBSD
- 1997: A book about the open-source culture: Eric Raymond “The Cathedral and the Bazaar” → Netscape released the sources
- 1998: Open Source summit (California): “freeware”, “sourceware”, “**open source**”, and “freed software”?
- 2000: StarOffice sources released (Sun Microsystems) → OpenOffice, LibreOffice...

Open source

- Developer:
  - Compiled application → Distribution
  - Source code & license → Distribution
- Users:
  - Access to the source code: Read (use? modify?)
  - Feedback to the developer

Advantages (users' perspective)

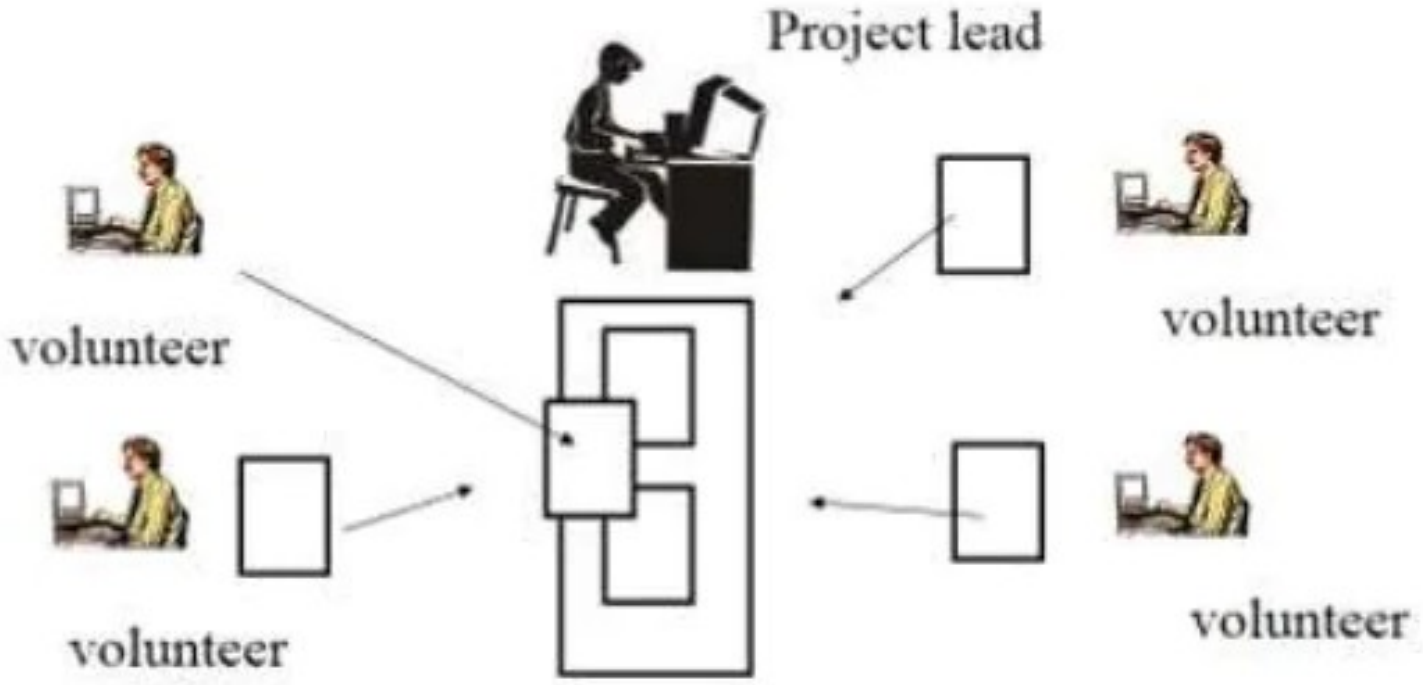
- Availability:
  - Source code as a template to learn from
  - Prevents reinventing the wheel
- Independence of the vendor:
  - Free choice of support
  - Potential to fix bugs
  - Potential to customized improvements
- (No costs)

Advantages (developers' perspective)

- Easy distribution system:
  - More wide distribution
  - Better visibility
  - Higher level of awareness
  - Quicker build-up of the community
- (Potential) feedback from the users:
  - Distributed testing → Faster reaction time to bugs → Software improves faster
- (Potential) improvements by the users
  - Distributed development → Software evolves quicker

Open source

- Development:



Ritwick Halder (2013)

Open versus closed source

- Closed source:
  - Source code access only on request and/or after signing a non-disclosure agreement (NDA)
- Open source:
  - Free access to view the source code of an application
    - No discrimination against persons, groups, or fields
  - Potential to use in commercial applications
  - Potential to modify and reuse the source
  - Potential to redistribute the modified version
  - Potential for feedback and improvements from the users

Famous open-source software examples

- Linux, Android
- OpenOffice, LibreOffice
- GIMP, Blender
- 7-Zip, Eclipse
- Firefox, Chromium
- Perl, PHP, Python, Ruby, Java
- Apache, Joomla!, Wordpress
- Octave, R, Torch

Open source: Let's start!

- Step 1: Choose a license!

Open-source licenses

- Open Source Initiative (OSI):
  - Academic Free License 3.0 (AFL-3.0)
  - Adaptive Public License (APL-1.0)
  - Apache Software License 1.1 (Apache-1.1) (superseded)
  - Apache License 2.0 (Apache-2.0)
  - Apple Public Source License (APSL-2.0)
  - Artistic license 1.0 (Artistic-1.0) (superseded)
  - Artistic License 2.0 (Artistic-2.0)
  - Attribution Assurance License (AAL)
  - Boost Software License (BSL-1.0)
  - BSD Licenses: 0-clause, 1-clause, 2-clause, 2-Clause-Patent, 3-clause, 3-Clause-LBNL

Open-source licenses

- CERN Open Hardware Licence Version 2 – Permissive, - Weakly Reciprocal, - Strongly Reciprocal
- CeCILL License 2.1 (CECILL-2.1)
- Common Development and Distribution License 1.0 (CDDL-1.0)
- Common Public Attribution License 1.0 (CPAL-1.0)
- Common Public License 1.0 (CPL-1.0) (superseded)
- Computer Associates Trusted Open Source License 1.1 (CATOSL-1.1)
- Cryptographic Autonomy License v.1.0 (CAL-1.0)
- CUA Office Public License Version 1.0 (CUA-OPL-1.0) (retired)
- Eclipse Public License 1.0 (EPL-1.0) (superseded)
- Eclipse Public License 2.0 (EPL-2.0)
- eCos License version 2.0 (eCos-2.0)
- Educational Community License, Version 1.0 (ECL-1.0) (superseded)
- Educational Community License, Version 2.0 (ECL-2.0)
- Eiffel Forum License V1.0 (EFL-1.0) (superseded)
- Eiffel Forum License V2.0 (EFL-2.0)
- Entessa Public License (Entessa)
- EU DataGrid Software License (EUDatagrid)
- European Union Public License 1.2 (EUPL-1.2)
- Fair License (Fair)
- Frameworx License (Frameworx-1.0)
- Free Public License 1.0.0 (0BSD)
- GNU Affero General Public License version 3 (AGPL-3.0)

Open-source licenses

- GNU General Public License version 2 (GPL-2.0)
- GNU General Public License version 3 (GPL-3.0)
- GNU Lesser General Public License version 2.1 (LGPL-2.1)
- GNU Lesser General Public License version 3 (LGPL-3.0)
- Historical Permission Notice and Disclaimer (HPND)
- IBM Public License 1.0 (IPL-1.0)
- Intel Open Source License (Intel) (retired)
- IPA Font License (IPA)
- ISC License (ISC)
- Jabber Open Source License (retired)
- LaTeX Project Public License 1.3c (LPPL-1.3c)
- Lawrence Berkeley National Labs BSD Variant License (BSD-3-Clause-LBNL)
- Licence Libre du Québec – Permissive (LiLiQ-P) version 1.1 (LiliQ-P)
- Licence Libre du Québec – Réciprocité (LiLiQ-R) version 1.1 (LiliQ-R)
- Licence Libre du Québec – Réciprocité forte (LiLiQ-R+) version 1.1 (LiliQ-R+)
- Lucent Public License ("Plan9"), version 1.0 (LPL-1.0) (superseded)
- Lucent Public License Version 1.02 (LPL-1.02)
- Microsoft Public License (MS-PL)
- Microsoft Reciprocal License (MS-RL)
- MirOS Licence (MirOS)

Open-source licenses

- MIT License (MIT)
- MIT No Attribution License (MIT-0)
- MITRE Collaborative Virtual Workspace License (CVW) (retired)
- Motosoto License (Motosoto)
- Mozilla Public License 1.0 (MPL-1.0) (superseded)
- Mozilla Public License 1.1 (MPL-1.1) (superseded)
- Mozilla Public License 2.0 (MPL-2.0)
- Mulan Permissive Software License v2 (MulanPSL - 2.0)
- Multics License (Multics)
- NASA Open Source Agreement 1.3 (NASA-1.3)
- Naumen Public License (Naumen)
- Nethack General Public License (NGPL)
- Nokia Open Source License (Nokia)
- Non-Profit Open Software License 3.0 (NPOSL-3.0)
- NTP License (NTP)
- OCLC Research Public License 2.0 (OCLC-2.0)
- Open Group Test Suite License (OGTSL)
- Open Software License 1.0 (OSL-1.0) (superseded)
- Open Software License 2.1 (OSL-2.1) (superseded)
- Open Software License 3.0 (OSL-3.0)
- OpenLDAP Public License Version 2.8 (OLDAP-2.8)

Open-source licenses

- OSET Public License version 2.1
- PHP License 3.0 (PHP-3.0) (superseded)
- PHP License 3.01 (PHP-3.01)
- The PostgreSQL License (PostgreSQL)
- Python License (Python-2.0) (overall Python license)
- CNRI Python license (CNRI-Python) (CNRI portion of Python License)
- Q Public License (QPL-1.0)
- RealNetworks Public Source License V1.0 (RPSL-1.0)
- Reciprocal Public License, version 1.1 (RPL-1.1) (superseded)
- Reciprocal Public License 1.5 (RPL-1.5)
- Ricoh Source Code Public License (RSCPL)
- SIL Open Font License 1.1 (OFL-1.1)
- Simple Public License 2.0 (SimPL-2.0)
- Sleepycat License (Sleepycat)
- Sun Industry Standards Source License (SISSL) (retired)
- Sun Public License 1.0 (SPL-1.0)

Open-source licenses

- Sybase Open Watcom Public License 1.0 (Watcom-1.0)
- Universal Permissive License (UPL)
- University of Illinois/NCSA Open Source License (NCSA)
- Upstream Compatibility License v1.0
- Unicode Data Files and Software License
- The Unlicense
- Vovida Software License v. 1.0 (VSL-1.0)
- W3C License (W3C)
- wxWindows Library License (WXwindows)
- X.Net License (Xnet)
- Zero-Clause BSD (0BSD)
- Zope Public License 2.0 (ZPL-2.0) (superseded)
- Zope Public License 2.1 (ZPL-2.1)
- zlib/libpng license (Zlib)

- Only those approved by the Open Source Initiative!

Open-source licenses

- Most popular (according to Open Source Initiative):
  - Apache License 2.0 (Apache-2.0)
  - 3-clause BSD license (BSD-3-Clause)
  - 2-clause BSD license (BSD-2-Clause)
  - **GNU General Public License (GPL)**
  - GNU Lesser General Public License (LGPL)
  - MIT license (MIT)
  - Mozilla Public License 2.0 (MPL-2.0)
  - Common Development & Distribution License 1.0 (CDDL-1.0)
  - Eclipse Public License 2.0 (EPL-2.0)

Open-source licenses

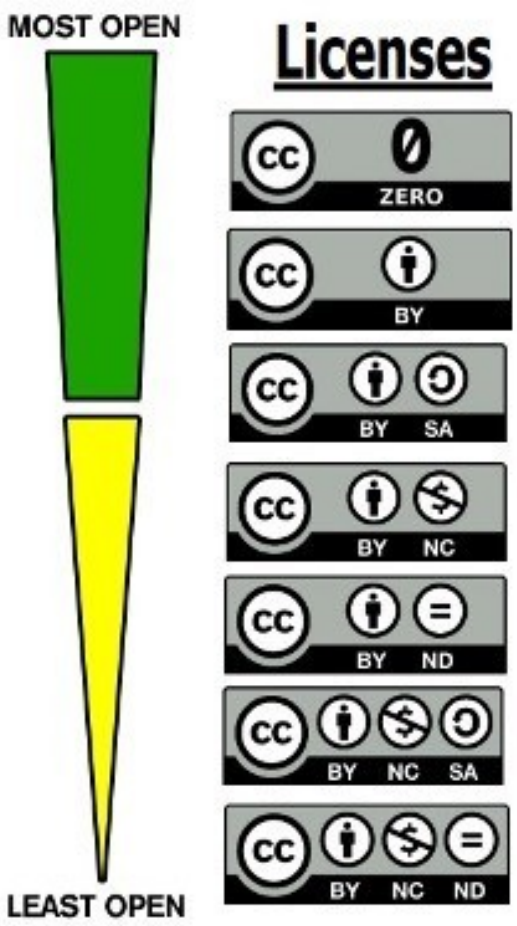
- International:
  - Germany: Deutsche Freie Software Lizenz (D-FSL-1.0)
  - France: CeCILL License 2.1
  - Quebec: Licence Libre du Québec – Permissive (LiLiQ-P), Réciprocité (LiLiQ-R), Réciprocité forte (LiLiQ-R+)
  - China: Mulan Permissive Software License v2
  - EU: **European Union Public License (EUPL-1.2)**
- Tools for selection:
  - E.g., https://joinup.ec.europa.eu/collection/eupl/jla-joinup-licensing-assistant-select-and-compare-open-licences

Open-source licenses

- International:
  - Germany: Deutsche Freie Software Lizenz (D-FSL-1.0)
  - France: CeCILL License 2.1
  - Quebec: Licence Libre du Québec – Permissive (LiLiQ-P), Réciprocité (LiLiQ-R), Réciprocité forte (LiLiQ-R+)
  - China: Mulan Permissive Software License v2
  - EU: **European Union Public License (EUPL-1.2)**
- Tools for selection:
  - E.g., https://joinup.ec.europa.eu/collection/eupl/jla-joinup-licensing-assistant-select-and-compare-open-licences
- No license: Default copyright of your country applies, usually, “all rights reserved”

Open-source licenses

- Other data (research data, graphics, audio, etc)?
  - Creative Commons licenses:
    - CC0
    - CC BY
    - CC BY SA
    - CC BY NC
    - CC BY ND
    - CC BY NC SA
    - CC BY NC ND



This work is a CC0 Public Domain Dedication work.

Open source: Let's start!

- Step 1: We have a license!
- Step 2: Set up a public code repository

Code repository

- Code repository: Archive for the code
  - Storing the code
  - Version control
  - Publishing and distribution
  - User management
  - Issue tracker
  - Automatic testing capabilities
  - Preparation and distribution of releases
- Examples: Github, Bitbucket, Assembla, Backlog, Sourceforge
- Important: Choose a version-control system
  - Simple system: Subversion
  - Distributed system: Git

Open source: Let's start!

- Step 1: We have a license!
- Step 2: We use a public code repository
- Step 3: We did some programming

Open source: Let's start!

- Step 1: We have a license!
- Step 2: We use a public code repository
- Step 3: We did some programming
- Step 4: Documentation!

Documentation

- Types of documentations:
  - Learning oriented (tutorials)
  - Goal oriented (how-to guides, manuals)
  - Understanding oriented (discussions)
  - **Information oriented** (reference material, indexes, syntax descriptions)
- Embed it in the code:
  - Easier to keep up-to-date
  - Consistent with the code
  - Documentation compiler required
  - Examples: Doxygen, Javadoc, Sphinx, PyDoc, (Visual Expert)

Open source: Let's start!

- Step 1: We have a license
- Step 2: We use a public code repository
- Step 3: We did some programming
- Step 4: We documented the code
- Step 5: Want to create a release? Version number!

Software versioning

- Date of release, e.g., MyApp 20210924
- Tex: number of Pi, e.g., 3.14159265
- Apple, Windows, … : marketing strategy
- Sequence based: e.g., 1.4.2.1234
  - Semantic versioning (SemVer):
  - Major: not backwards-compatible changes
  - Minor: new features
  - Patch: bug fixes
  - (development state)

Stage	SemVer	Num 90+
Alpha	1.2.0-a.1	1.1.90
Beta	1.2.0-b.2	1.1.93
Release candidate	1.2.0-rc.3	1.1.97
Release	1.2.0	1.2.0
Post-release fixes	1.2.5	1.2.5

Open source: Let's start!

- Step 1: We have a license
- Step 2: We use a public code repository
- Step 3: We did some programming
- Step 4: We documented the code
- Step 5: We have a version number
- Step 6: Prepare a package for download

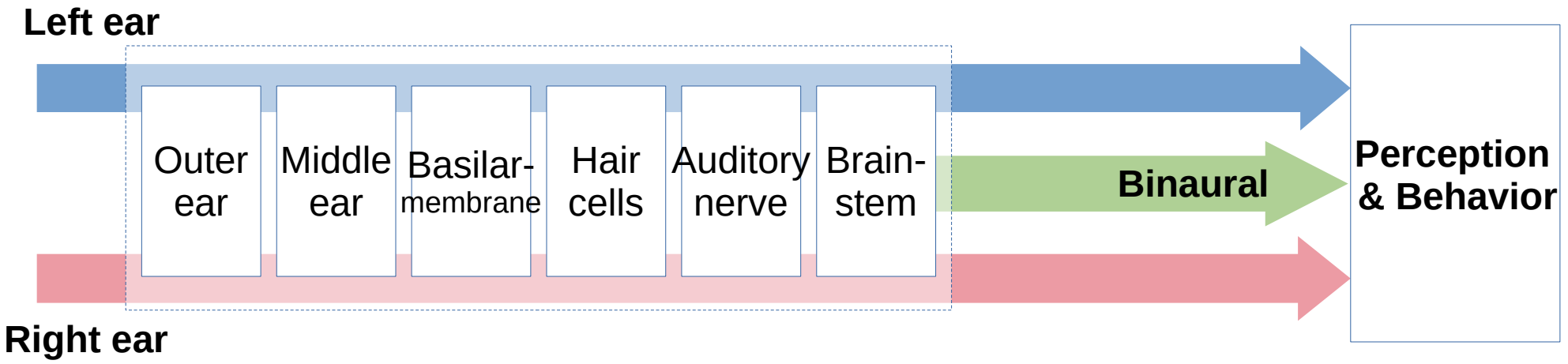
- Three examples: AMT, ExpSuite, and the SOFA Toolbox
  - Concept, Statistics, History
  - License, Code repository, Versioning, Documentation

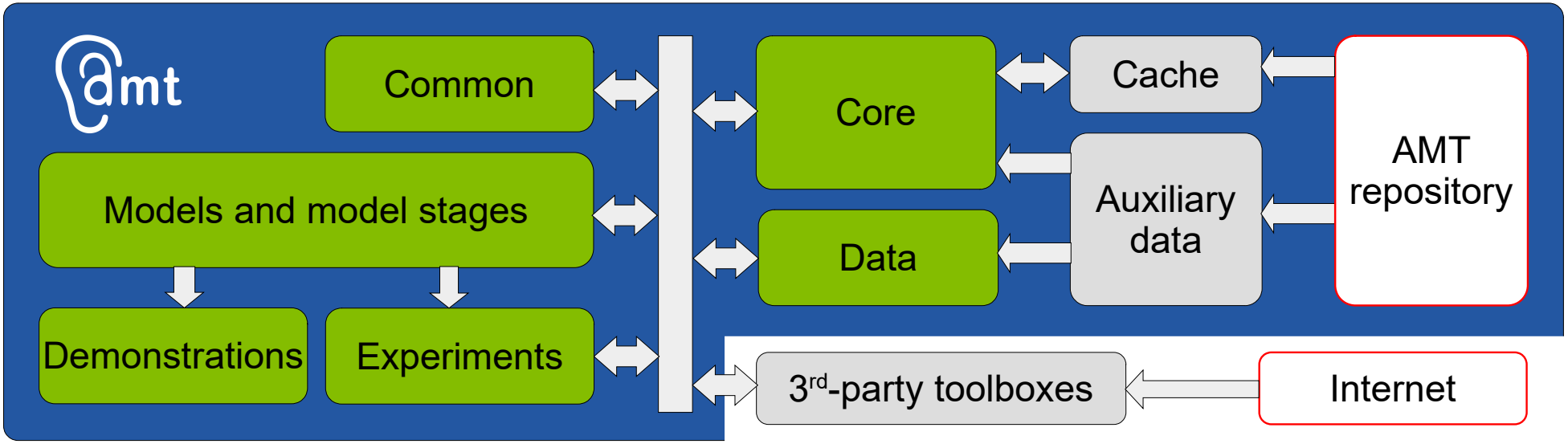
The Auditory Modeling Toolbox (AMT)

- An open-source toolbox for auditory modeling:
  - A framework for developing **new models** by providing verified components
  - A tool for performing scientific experiments with **existing models**
  - Strong focus on **reproducible research***
  - An instrument to make a large number of models available in **common programming languages**:
    - Common: Matlab/Octave
    - Embedded: C, C++, Python
    - Supported: Any language
  - Combines **code and data**

*) Jasny, B. R., Chin, G., Chong, L., and Vignieri, S. (2011). “Again, and Again, and Again …,” Science, 334, 1225.

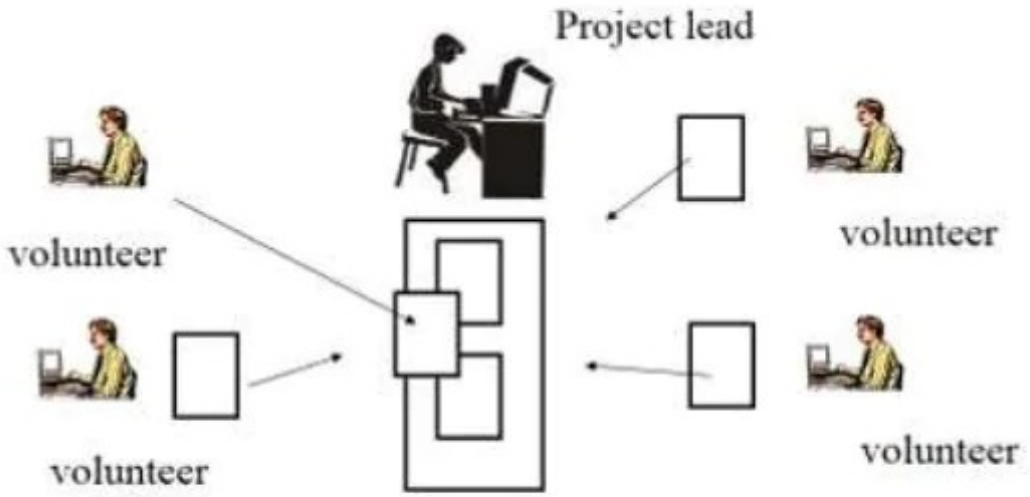
The Auditory Modeling Toolbox (AMT)





The AMT – Statistics

- Code: approximately 700 files, 75 auditory models
- Data: 48 data sets (80 GB in total)
- Structure:
  - 1 project leader
  - 50 contributors
- Development:
  3771 commits
- Download: over 17000
  (since 2009)



Ritwick Halder (2013)

The AMT – History

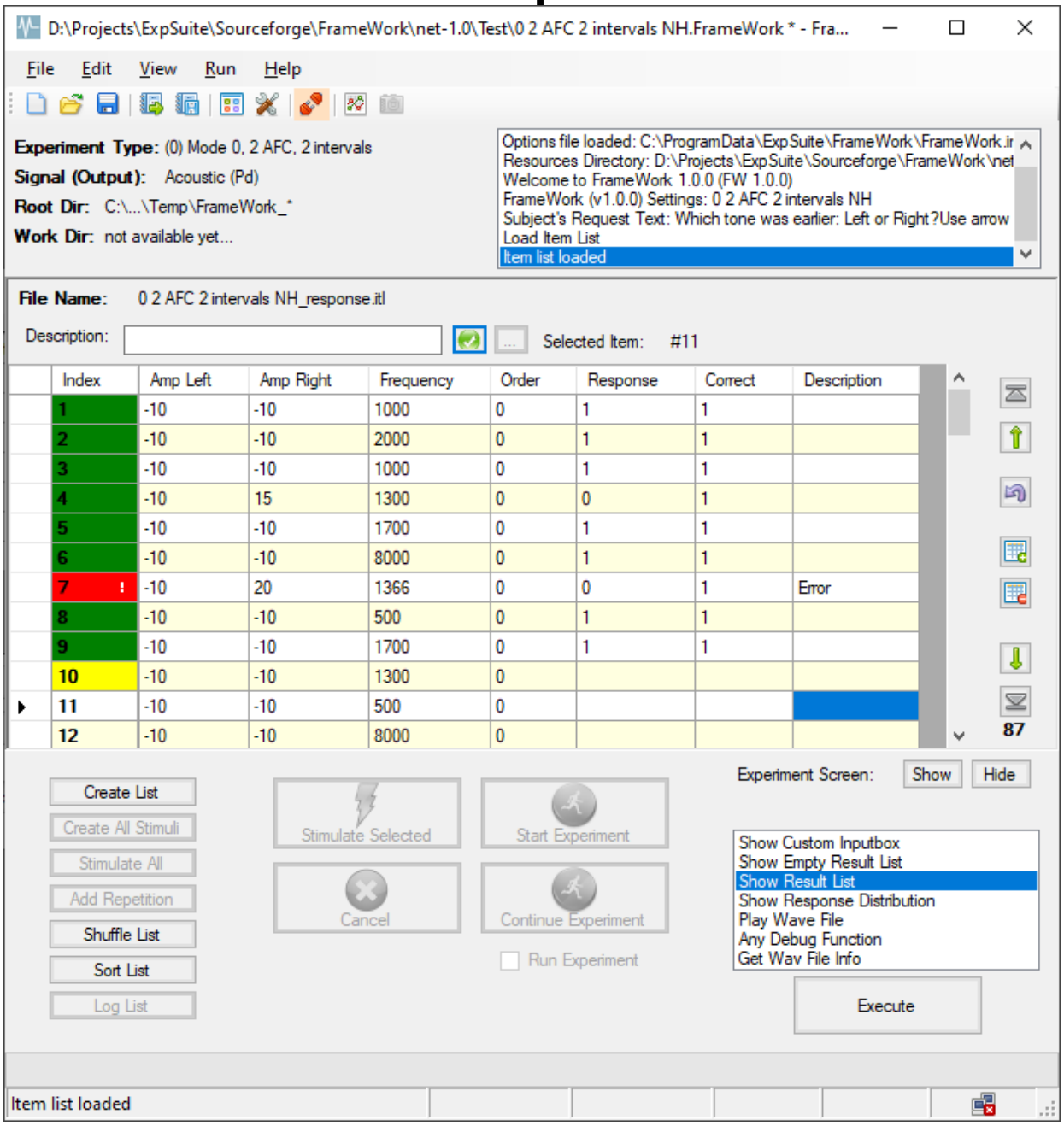
- 2010: version 0.01 as a draft
  - License: GPL version 3
  - Code repository: Sourceforge.net
  - Version control: Git; Software versioning: Incremental
  - Documentation: in-code (Mat2Doc), compiled to a website
- 2011: version 0.0.3: Switch to semantic versioning
- 2012: versions 0.1.x, 0.2.x, and 0.3.x
- 2013 to 2020: versions 0.9.1 to version 0.10.0
- 2021: version 1.0.0
  - Documentation website changed to amtoolbox.org
  - Multi-licensing introduced (two licenses)
- 2024: version 1.6.0 (three licenses: GPL, Apache, Ghent Univ.)

The ExpSuite

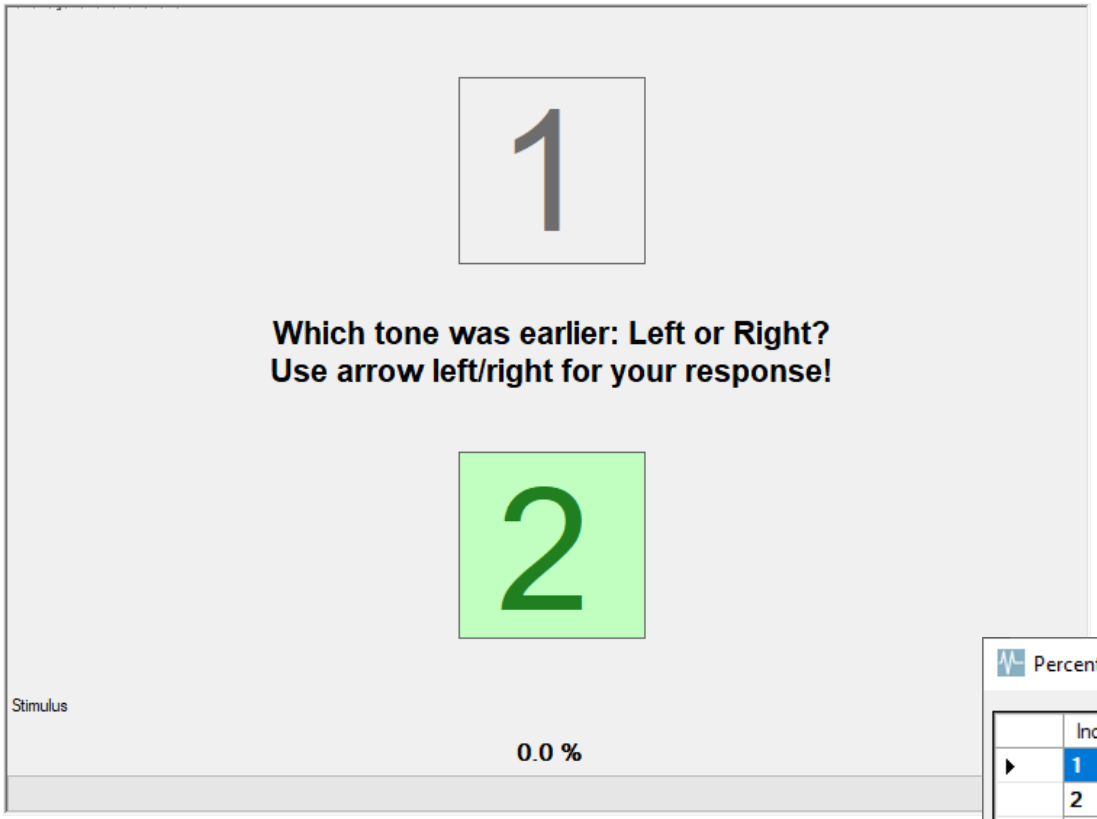
- Software **framework** for applications to perform psychoacoustical experiments and acoustic measurements
- Psychoacoustics: acoustic and electric stimulation for normal-hearing and cochlear-implant listeners
- Measurements: multichannel input/output, real-time
- Modular components:
  - GUI: .NET (in Visual Basic)
  - Offline processing: Matlab
  - Audio interface: pure data (via OSC)
  - VR support: Unity (via OSC)

ExpSuite

The ExpSuite



The ExpSuite



Percentage Correct

	Index	Amp Left	Amp Right	Frequency	# of Items	# of Correct	% Correct
▶	1	-10	-10	500	1	1	100
	2	-10	-10	1000	2	2	100
	3	-10	-10	1700	2	2	100
	4	-10	-10	2000	1	1	100
	5	-10	-10	8000	1	1	100
	6	-10	15	1300	1	1	100
	7	-10	20	1366	1	1	100

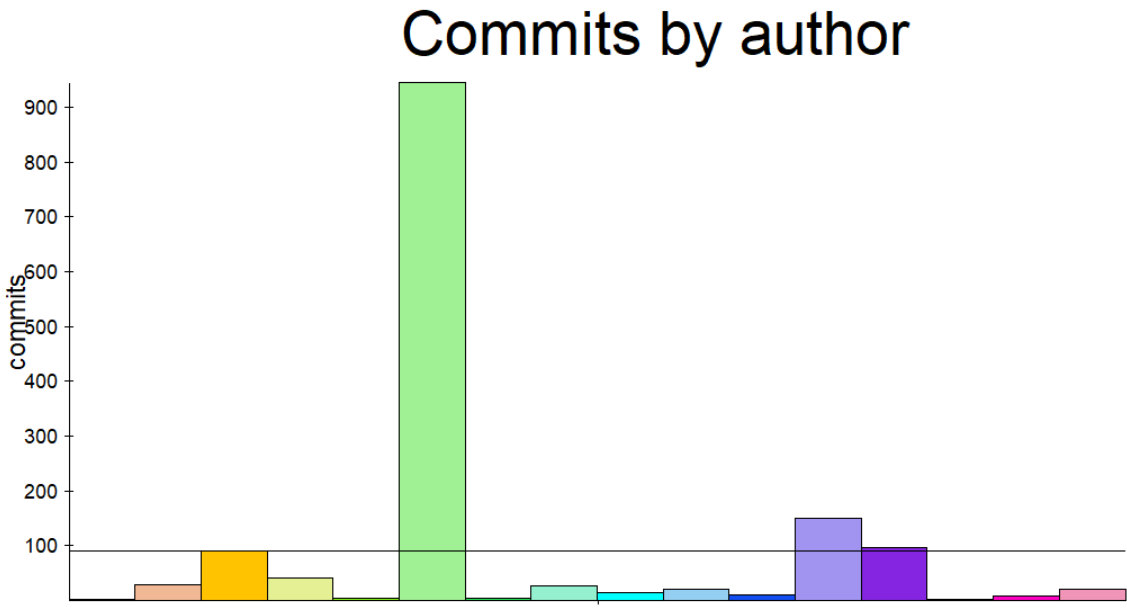
Print (Portrait) | Set opt.width | X axis | Set | Plot Results | Load list | Write to Log
Print (Landscape) | Renumber | Sort by: Index | Sort | Append list | Save As... | Close

The ExpSuite

- Applications:
  - **AMT@ARI**: HRTF Measurements, system identification
  - **Loca**: sound localization in VR
  - **Ironie**: irony detection
  - **BiPhase**: binaural phase detection
  - **RhySeg**: rhythm-based stream segregation
  - **SpExCue**: spectral cues for sound externalization
  - **PitchSIPI**: pitch sensitivity in electric hearing
  - **LoudSca**: loudness scaling
  - … 42 applications in total

The ExpSuite – Statistics

- Code: over 30 000 files, 42 applications
- Structure:
  - 16 contributors
  - 1451 commits
  - But: mostly by two contributors (project leaders)
- 5793 downloads
  - Used by our institute mostly



The ExpSuite – History

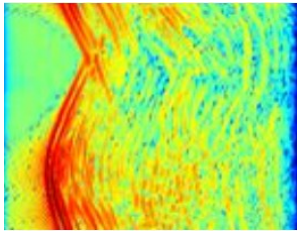
- 2003: development started as version 0.001 in VB6
  - License: No License
  - Code: No code repository (stored on an internal server)
  - Versioning: No version control; Incremental
  - No documentation
- 2003: version 0.0.17: Semantic versioning
- 2004: version 0.0.38 (first experiment, published 2006)

The ExpSuite – History

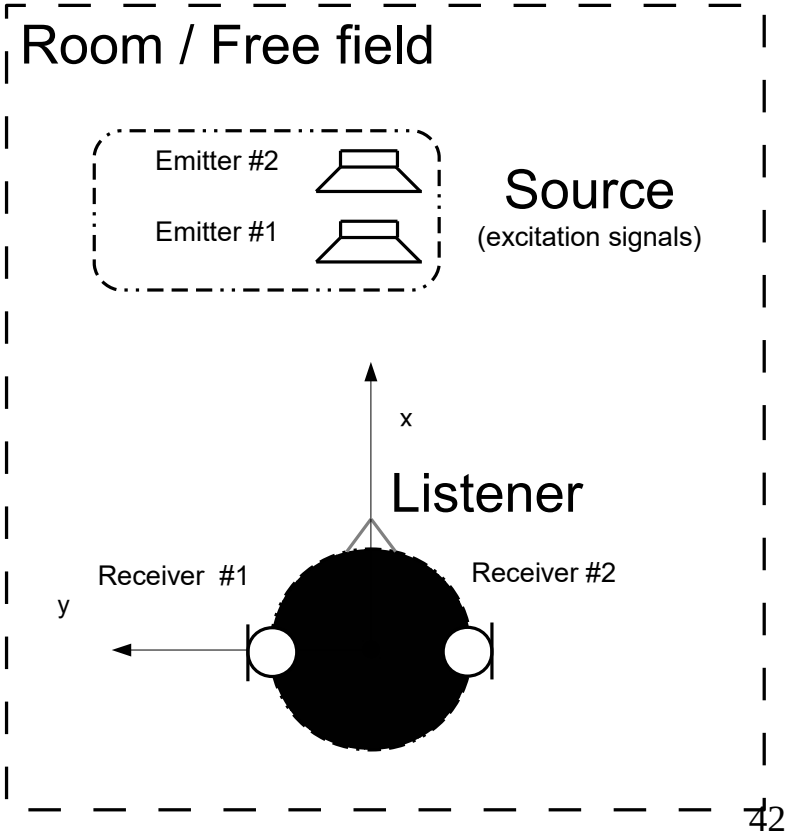
- 2009: version 0.8.0: Migration to .NET
  - License: **EUPL** (first EUPL project on Sourceforge!)
  - Code repository: Sourceforge
  - Versioning: Subversion, semantic versioning
  - Documentation: VBDOX
- 2013 to 2018: versions 0.9.0 to 0.10.0:
- 2019: versions 1.0 and 1.1: Audio player upgrade, Video player support, Unity support
- 2024: version 1.2.16

The SOFA Toolbox

- Spatially oriented format for acoustics (SOFA)
  - Standardized by the Audio Engineering Society (AES69-2015)
  - Widely used to store HRTFs
- Toolbox to read, modify, and write SOFA files
- Written in Matlab/Octave
- Role model for other SOFA-related software



SOFA Toolbox



The SOFA Toolbox – Statistics

- Code: Approximately 200 files
- 17 contributors
  - 700 commits
- Download: 17600 (since 2015)
- Triggered 18 forks
- Data and documentation: Wikimedia Website with
  - SOFA-related information
  - Over 800 GB of SOFA files

The SOFA Toolbox – History

- 2012: version 0.0.1 (**SOFA API MO**)
  - License: EUPL version 1.2
  - Code repository: Sourceforge.net
  - Versioning: Subversion; Semantic versioning
  - Documentation: in-code, Mat2Doc
- 2013 to 2015: versions 0.1.x to 0.4.x
- 2015: version 0.9 (release candidate) for the standard AES69-2015
  - Code repository: Github
  - Version control: git
  - Releases: still on Sourceforge
- 2015: version 1.0.0: Consistent with the standard AES69-2015

The SOFA Toolbox – History

- 2021: version 1.1.3: Approaching the revision of the standard (AES69-2020)
- 2022: version 2.0.0: Renamed to **SOFA Toolbox**, no release(!)
- 2022: version 2.1.0: Release, consistent with AES69-2022
- 2023: version 2.2.0: New features

Summary – Examples

- The SOFA Toolbox (2012-):
  - For developers, forked by many, much feedback, mainly developed by us
  - EUPL v1.2
- The AMT (2009-):
  - For developers, many contributors, integration of others' code
  - GPL v3 mainly, specific files with other licenses
- The ExpSuite (2003-):
  - For our users, no feedback, a few known external users
  - EUPL v1.2
- All this code is never finished...

Summary – The code

- Step 1: Choose a license!
- Step 2: Use a public code repository!
- Step 3: Go programming!
- Step 4: Documentation!
- Step 5: Version number!
- Step 6: Publish a package **… and iterate!**

- Code is never finished: Publish it, it's good enough!

Barnes, N. (2010). “Publish your computer code: it is good enough,” Nature, 467, 753.

Summary – Research data

- Step 1: Choose a license!
- Step 2: ~~Use~~ Choose a public ~~code~~ data repository!
- ~~Step 3: Go programming!~~
- Step 4: Documentation!
- Step 5: Version number!
- Step 6: Publish **persistently (with a DOI)**

- Any questions?

(add on)

- Myth #1: The bottom line will plummet
  - It’s unlikely if you have built a healthy ecosystem around your projects.
- Myth #2: Competitors will steal your features
  - Open-source code allows competitors to follow the logic of your features, but it doesn’t mean a smooth integration of these features into their work.
- Myth #3: Open-source code isn’t valuable
  - Even giants such as Windows, macOS, and Linux contain open-source components without any adverse effect.
- Myth #4: Open source = loss of control
  - Take it easy, you're just letting other people use it (= users), while the code is still yours (= copyright owner). Use a robust open-source license to reserve all the rights you want.