

# WorkflowMaker, a visual workflow software tool for the easy integration of self-developed console applications

28 January 2025

## Summary

WorkflowMaker is a set of tools aimed at the *visual design* and subsequent *execution* of workflows that integrate *console applications* developed by users themselves, whether in research (or even production) environments involving in-house software development.

In this context, *visual design* means *drawing* the workflow by including tasks (the applications) and connecting their inputs and outputs, thereby creating a *template* that specifies who does what (applications) and how information flows between them (data connections). See Figure 1. *Execution* means automatically generating the necessary scripts to process the data according to the designed workflow.

The goal of WorkflowMaker is to reduce the complexity of combining different console applications to create said workflows. It seeks to eliminate the need for manual execution of these applications or the writing of automation scripts, which require skills that not everyone possesses. This approach makes it possible for these applications to be used not only by their developers but also by any expert in the application field, even if they lack programming skills.

WorkflowMaker is available for Windows and Ubuntu-based Linux platforms.

## Statement of need

It is not surprising to find research (or even production) groups that create their own software. Often, however, despite being experts in their field of knowledge, only some members of these groups have programming skills.

Developers in these groups are able to create *console* applications with relative ease, as these do not include graphical interfaces. Combining these, it is possible

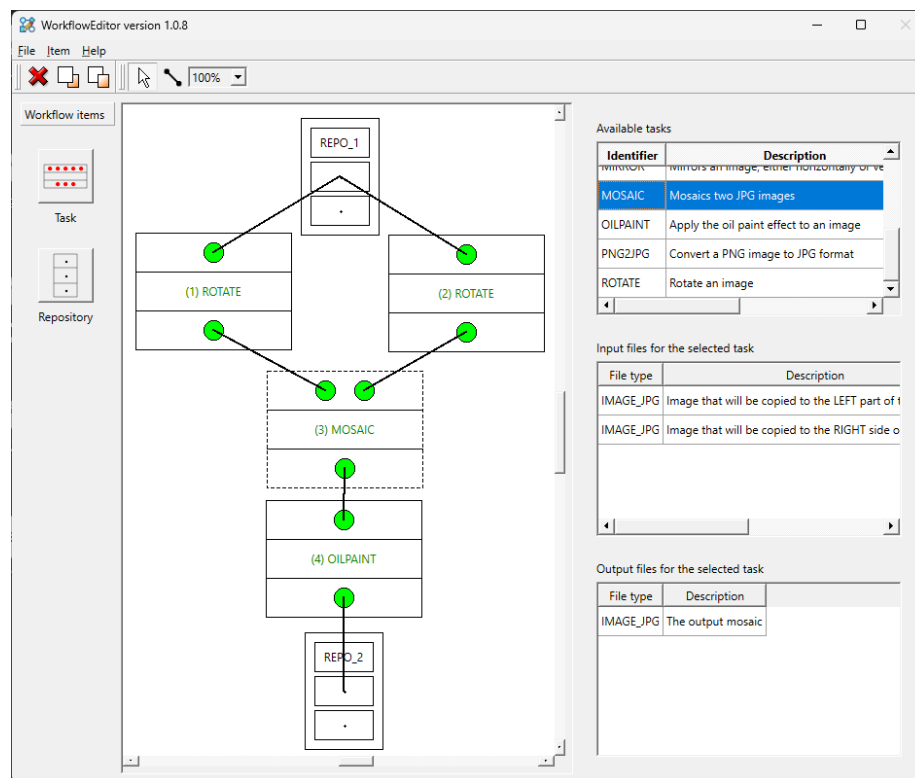


Figure 1: WorkflowEditor: visually designing a workflow.

to achieve more complex results. This requires either manually running the applications one by one or *writing scripts* to automate the process. This, which may be easy for developers, can be a stumbling block for those without programming skills, who need to rely on their more qualified colleagues to do it. This limits the flexibility of the group, creating dependence among its members—and bottlenecks.

The author believes it would be highly beneficial if those without programming skills could design workflows by combining the applications created by their colleagues using a tool that (1) would literally allow them to draw such workflows and (2) automatically generate the necessary scripts to execute them.

Although it is very difficult to prove the absolute non-existence of something, the author has not been able to find a visual tool to draw and execute workflows made of console applications.

Kuhail et al. (2021) carried out an exhaustive review of articles related to visual programming. The conclusions of this study on target fields of application are: 23.3 % of the tools are focused on the Internet Of Things (IOT) (Valtolina et al. 2019; Danado and Paternò 2014; Francese, Risi, and Tortora 2017; Johnsson and Magnusson 2020); another 23.3 % is oriented towards education (Kunimune et al. 2019; Abe, Fukawa, and Tanaka 2019; Broll et al. 2018; De Luca et al. 2018; Feng, Gardner, and Feng 2017; Rao, Bihani, and M. 2018); the remainder are spread across the fields of Business Process Management (BPM) (Weber, Paik, and Benatallah 2013), museum exhibits (Stratton, Bates, and Dearden 2017), reporting (Chen and Tu 2014), records management (Cabitzaiade and Gesso 2012), geographic information (Luong et al. 2012), robotics (Alexandrova, Tatlock, and Cakmak 2015), mechanical engineering (Sorce et al. 2019), information visualization (Mei et al. 2018), and psychiatry (Rough and Quigley 2015).

A less academic search for tools, whether open-source or proprietary, produced a long list. The best ones, according to (Pathak 2022; Boitnott 2023; Towns 2024) are listed here in alphabetical order: Airtable (Airtable 2024), Appy Pie (Appy Pie 2024), Betty Blocks (Betty Blocks 2024), Bubble (Bubble Group Inc. 2024), Carrd (Carrd Inc. 2024), Caspio (Caspio Inc. 2024), Jotform (Jotform Inc. 2024), Knack (Knack 2024), m-Power (mrc Ltd. 2024), Mailchimp (The Rocket Science Group 2024), Mendix (Mendix Technology BV 2024), Microsoft Power Apps (Microsoft 2024), Notion (Notion Labs Inc. 2024), OutSystems (OutSystems 2024), Quickbase (Quickbase 2024), Retool (Retool Inc. 2024), Shopify (Shopify.com 2024), SNAP (European Space Agency 2024), Visual LANSA (LANSA Inc. 2024), Webflow (Webflow Inc. 2024), Zapier (Zapier Inc. 2024) and Zoho Creator (Zoho Corporation Pvt. Ltd. 2024). For a very extensive list of visual programming / workflow / no-code / low-code tools please refer to (G2 2024).

Most of the tools above are targeted at BPM; they let users build workflows visually, connecting data and processes. Usually, they offer standard data stores (such as commercial or open-source databases or spreadsheets) and applications

available in the market to be used as the nodes in the workflow. Other applications are aimed at marketing / email distribution or website / e-shop builders.

Unfortunately, there does not seem to be any tool to build workflows from console applications.

## Description

WorkflowMaker consists of three applications.

*ToolkitEditor* — This tool allows developers to *formally characterize* their console applications by specifying (1) their keyboard parameters and (2) input and output files. In this context, *keyboard parameter* stands for these values that users provide using the keyboard when the console application is run.

To be integrable with WorkflowMaker applications must comply with very few constraints:

- They must accept a single command-line parameter: the name of a plain text file containing the values of the keyboard parameters as well as the names of the input / output files. The format of this file is standardized.
- They must return a value indicating whether the execution completed successfully or not so that the scripts automatically generated by WorkflowLauncher can manage the execution flow in case of errors.

Figure 2 shows the formal characterization of a console application.

*WorkflowEditor* — This is a visual editor that allows users to design workflows by combining the tasks defined with ToolkitEditor. It relies on three pillars: (1) repositories, which are locations where input and output files reside, (2) tasks, the formally defined applications, and (3) connections, which indicate how data flows between repositories and tasks. See Figure 1 above.

*WorkflowLauncher* — The workflows created with WorkflowEditor are just *templates* that outline how to perform some process, but say nothing about the data to be processed—that is, these can be reused for many data sets. WorkflowLauncher is used to set the names of the input files for those workflows, as well as the values of the required keyboard parameters. From this, an executable script is generated. Figure 3 depicts how the values of some keyboard parameters are set.

## A very simple example

Figure 4(a) shows a very simple image processing workflow drawn with WorkflowEditor; there, two images stored in some repository (REPO\_1) are rotated (tasks (1) ROTATE and (2) ROTATE). Then, their outputs are used by a third task ((3) MOSAIC) to mosaic these images producing a new one, which is used

Task data

General info

Identifier

OILPAINT

Description

Apply the oil paint effect to an image

Executable

oilpaint

Parameters

✕

+

Total: 1

Identifier	Type	Description
RADIUS	REAL	Radius (in pixels) for which an image processing f

Input files

✕

+

Total: 1

File type	Options file label	Description
IMAGE_JPG	INPUT_FILENAME	The input JPG file to which the oil pa

Output files

✕

+

Total: 1

File type	Options file label	Description
IMAGE_JPG	OUTPUT_FILENAME	The output JPG file, with the oil paint

Cancel

Save

Figure 2: ToolkitEditor: formally characterizing the OILPAINT console application.

WorkflowLauncher 1.0.8

Parameters Repositories Files

PARAMETER

Task identifier	(1) ROTATE
Description	Amount in degrees to apply by the image processing function
Type	REAL
Label	DEGREES
Value	90

PARAMETER

Task identifier	(2) ROTATE
Description	Amount in degrees to apply by the image processing function
Type	REAL
Label	DEGREES
Value	90

PARAMETER

Task identifier	(4) OILPAINT
Description	Radius (in pixels) for which an image processing function is applied
Type	REAL
Label	RADIUS
Value	10

Quit Load launcher New launcher Save launcher Export shell files

Figure 3: WorkflowLauncher: Setting the values of keyboard parameters.

by the last task ((4) OILPAINT) to apply an artistic image effect known as “oil painting”. The result of this process is stored in a new repository, (REPO\_2). In Figure 4(b) the two input images used to run this workflow are shown; 4(c) depicts the result.

The workflow in the example has been created using the *sample image processing toolkit* available in WorkflowMaker’s GitHub repository.

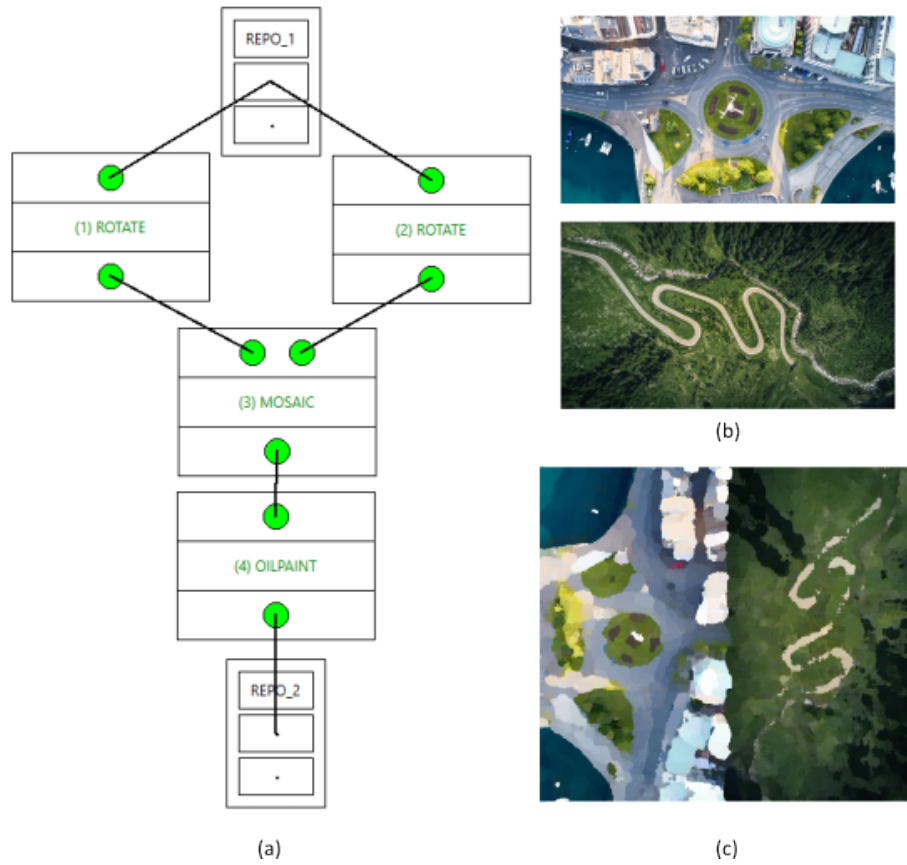


Figure 4: A simple image processing workflow (a), the two input images (b), and the result (c).

## References

- Abe, K., Y. Fukawa, and T. Tanaka. 2019. “Prototype of Visual Programming Environment for C Language Novice Programmer.” In *Proceedings of 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*, 140–45. <https://doi.org/https://doi.org/10.1109/IIAI-AAI.2019.00037>.

- Airtable. 2024. “Create Apps That Perfectly Fit Your Team’s Needs.” Available online: <https://www.airtable.com/>.
- Alexandrova, S., Z. Tatlock, and M. Cakmak. 2015. “RoboFlow: A Flow-Based Visual Programming Language for Mobile Manipulation Tasks.” In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 5537–5544. Seattle, WA, USA. <https://doi.org/https://doi.org/10.1109/ICRA.2015.7139973>.
- Appy Pie. 2024. “No-Code Platform. Workflow Automation Platform.” Available online: <https://www.appypie.com/>.
- Betty Blocks. 2024. “No-Code: Enterprise Application Development Platform (Betty Blocks).” Available online: <https://www.bettyblocks.com/>.
- Boitnott, J. 2023. “8 Best No-Code Tools Every Business Needs for 2023.” Available online: <https://www.jotform.com/blog/no-code-tools/>.
- Broll, B., Á. Lédeczi, H. Zare, D. N. Do, J. Sallai, P. Völgyesi, M. Maróti, L. Brown, and C. Vanags. 2018. “A Visual Programming Environment for Introducing Distributed Computing to Secondary Education.” *Journal of Parallel and Distributed Computing* 118: 189–200. <https://doi.org/https://doi.org/10.1016/j.jpdc.2018.02.021>.
- Bubble Group Inc. 2024. “The Best Way to Build Web Apps Without Code | BUbble.” Available online: <https://www.bubble.io/>.
- Cabitzaiade, F., and G. Gesso. 2012. “Rule-Based Programming as Easy as a Child’s Play. A User Study on Active Documents.” In *Proceedings of International Conference on Interfaces and Human Computer Interaction (IHCI IADIS)*. Lisbon, Portugal.
- Carrrd Inc. 2024. “Carrrd: Simple, Free, Fully Responsive One-Page Sites for Pretty Much Anything.” Available online: <https://carrrd.co/>.
- Caspio Inc. 2024. “Build Online Database Apps - Low-Code Platform | Caspio.” Available online: <https://www.caspio.com/>.
- Chen, W.-K., and P.-Y. Tu. 2014. “VisualTPL: A Visual Dataflow Language for Report Data Transformation.” *Journal of Visual Languages & Computing* 25 (3): 210–26. <https://doi.org/https://doi.org/10.1016/j.jvlc.2013.11.003>.
- Danado, J., and F. Paternò. 2014. “Puzzle: A Mobile Application Development Environment Using a Jigsaw Metaphor.” *Journal of Visual Languages & Computing* 25 (4): 297–315. <https://doi.org/https://doi.org/10.1016/j.jvlc.2014.03.005>.
- De Luca, G., Z. Li, S. Mian, and Y. Chen. 2018. “Visual Programming Language Environment for Different IoT and Robotics Platforms in Computer Science Education.” *CAAI Transactions on Intelligence Technology* 3 (2): 119–30. <https://doi.org/https://doi.org/10.1049/trit.2018.0016>.
- European Space Agency. 2024. “SNAP download - .” Available online: <https://step.esa.int/main/download/snap-download/>.
- Feng, A., M. Gardner, and W.-C. Feng. 2017. “Parallel Programming with Pictures Is a Snap!” *Journal of Parallel and Distributed Computing* 105: 150–62. <https://doi.org/https://doi.org/10.1016/j.jpdc.2017.01.018>.
- Francesse, R., M. Risi, and Tortora. 2017. “Iconic Languages: Towards End-User Programming of Mobile Applications.” *Journal of Visual Languages & Computing*



- Computing* 38: 1–8. <https://doi.org/https://doi.org/10.1016/j.jvlc.2016.10.009>.
- G2. 2024. “Best No-Code Development Platforms.” Available online: <https://www.g2.com/categories/no-code-development-platforms>.
- Johnsson, B. A., and B. Towards Magnusson. 2020. “Towards End-User Development of Graphical User Interfaces for Internet of Things.” *Future Generation Computer Systems* 107: 670–80. <https://doi.org/https://doi.org/10.1016/j.future.2017.09.068>.
- Jotform Inc. 2024. “Free Online Form Builder and Form Creator | Jotform.” Available online: <https://www.jotform.com/>.
- Knack. 2024. “Online Database Software: Use Knack for Custom Online Databases and Systems.” Available online: <https://www.knack.com/>.
- Kuhail, M. A., S. Farooq, R. Hammad, and M. Bahja. 2021. “Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review.” *IEEE Access* 9: 14181–202. <https://doi.org/https://doi.org/10.1109/ACCESS.2021.3051043>.
- Kunimune, H., S. Kamijima, T. Yamamoto, and M. Niimura. 2019. “Trial to Increase Motivation on Programming by Using Hardware Control Functions in the AT Visual Programming Environment.” In *Proceeding of 2nd International Conference on Education Technology Management (ICETM)*, 50–53. <https://doi.org/https://doi.org/10.1145/3375900.3375917>.
- LANSa Inc. 2024. “Professional Low Code for RAD and IBMi / AS400 Modernization | LANSa.” Available online: <https://lansa.com/>.
- Luong, T. N., P. Etcheverry, C. Marquesuzaa, and T. Nodenot. 2012. “A Visual Programming Language for Designing Interactions Embedded in Web-Based Geographic Applications.” In *Proceedings of ACM Conference on Intelligent User Interfaces*, 207–16. New York, NY, USA. <https://doi.org/https://doi.org/10.1145/2166966.2167003>.
- Mei, H., W. Chen, Y. Ma, H. Guan, and W. Hu. 2018. “VisComposer: A Visual Programmable Composition Environment for Information Visualization.” *Visual Informatics* 2 (1): 71–81. <https://doi.org/https://doi.org/10.1016/j.visinf.2018.04.008>.
- Mendix Technology BV. 2024. “Low-Cost Application Development Platform | Mendix.” Available online: <https://www.mendix.com/>.
- Microsoft. 2024. “Microsoft Power Apps - Build Apps with AI | Microsoft.” Available online: <https://powerapps.microsoft.com/en-us/>.
- mrc Ltd. 2024. “Low-Cost Without Limits: The m-Power Application Development Platform.” Available online: <https://www.mrc-productivity.com/products/m-power.html>.
- Notion Labs Inc. 2024. “Notion Desktop Apps for Mac & Windows.” Available online: <https://www.notion.so/desktop>.
- OutSystems. 2024. “High-Performance, Low-Code App Development | OutSystems.” Available online: <https://www.outsystems.com/>.
- Pathak, A. 2022. “15 Best Low-Code or No-Code Platforms to Build Amazing Products.” Available online: <https://geekflare.com/nocode-platforms-to-build-product/>.

- Quickbase. 2024. “Dynamic Work and Complex Project Management Platform | Quickbase.” Available online: <https://www.quickbase.com/>.
- Rao, A., A. Bihani, and Nair. M. 2018. “Milo. A Visual Programming Environment for Data Science Education.” In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 211–15. <https://doi.org/https://doi.org/10.1109/VLHCC.2018.8506504>.
- Retool Inc. 2024. “Retool | The Fastest Way to Develop Effective Software.” Available online: <https://retool.com/>.
- Rough, D., and A. Quigley. 2015. “Jeeves – a Visual Programming Environment for Mobile Experience Sampling.” In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 121–29. Atlanta, GA, USA. <https://doi.org/https://doi.org/10.1109/VLHCC.2015.7357206>.
- Shopify.com. 2024. “Start and Grow Your e-Commerce Business.” Available online: <https://www.shopify.com/>.
- Sorce, S., A. Malizia, V. Gentile, P. Jiang, M. Atherton, and D. Harrison. 2019. “Evaluation of a Visual Tool for Early Patent Infringement Detection During Design.” In *End-User Development: 7th International Symposium, IS-EUD 2019 (Lecture Notes in Computer Science, Vol. 11553)*, edited by A. Malizia, S. Valtolina, A. Morch, A. Serrano, and A. Stratton. Vol. 11553. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing. [https://doi.org/https://doi.org/10.1007/978-3-030-24781-2/\\_12](https://doi.org/https://doi.org/10.1007/978-3-030-24781-2/_12).
- Stratton, A., C. Bates, and A. Dearden. 2017. “Quando: Enabling Museum and Art Gallery Practitioners to Develop Interactive Digital Exhibits.” In *End-User Development. 6th International Symposium, IS-EUD 2017 (Lecture Notes in Computer Science, Vol 10303)*, edited by S. Barbosa, P. Markopoulos, F. Paternò, S. Stumpf, and S. Valtolina. Vol. 10303. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing. [https://doi.org/https://doi.org/10.1007/978-3-319-58735-6/\\_7](https://doi.org/https://doi.org/10.1007/978-3-319-58735-6/_7).
- The Rocket Science Group. 2024. “Marketing, Automation & Email Platform | Mailchimp.” Available online: <https://mailchimp.com/en>.
- Towns, A. 2024. “9 No-Code Platforms You Need to Know About.” Available online: <https://www.getclockwise.com/blog/no-code-platforms>.
- Valtolina, S., F. Hachem, B. R. Barricelli, E. G. Belay, S. Bonfitto, and M. Mesiti. 2019. “Facilitating the Development of IoT Applications in Smart City Platforms.” In *End-User Development: 7th International Symposium, IS-EUD 2019 (Lecture Notes in Computer Science, Vol. 11553)*, edited by A. Malizia, S. Valtolina, A. Morch, A. Serrano, and A. Stratton. Vol. 11553. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing. [https://doi.org/https://doi.org/10.1007/978-3-030-24781-2/\\_6](https://doi.org/https://doi.org/10.1007/978-3-030-24781-2/_6).
- Weber, I., H.-Y. Paik, and B. Benatallah. 2013. “Form-Based Web Service Composition for Domain Experts.” *ACM Transactions on the Web* 8 (1): 1–40. <https://doi.org/https://doi.org/10.1145/2542168>.
- Webflow Inc. 2024. “WebFlow: Create a Custom Website | Visual Website Builder.” Available online: <https://webflow.com/>.
- Zapier Inc. 2024. “Automation That Moves You Forward | Zapier.” Available

online: <https://zapier.com/>.  
Zoho Corporation Pvt. Ltd. 2024. “Zoho Creator: Best Low-Code Custom  
Application Development Platform.” Available online: [https://www.zoho.c  
om/creator/](https://www.zoho.com/creator/).