# Domain-Specific Forges in Architecture, Engineering, and Construction : Principles and Prototypes

Milovann Yanatchkov

▶ **To cite this version:**

Milovann Yanatchkov. Domain-Specific Forges in Architecture, Engineering, and Construction : Principles and Prototypes. ENSAN. 2025. ⟨hal-05289291v1⟩

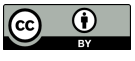
**HAL Id: hal-05289291**

**https://hal.science/hal-05289291v1**

Submitted on 29 Sep 2025 (v1), last revised 5 Oct 2025 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# Domain-Specific Forges in Architecture, Engineering, and Construction : Principles and Prototypes

Milovann Yanatchkov

September 2025

## Contents

## Abstract

This report explores the concept of the “Domain-Specific Forge (DSF).” “Forges” are collaborative platforms initiallly designed for software development, but they now find applications in domains beyond pure computing. We present three prototypes from our own experiments: a “Learning Forge,” a “Design Forge,” and an “AI Factory” in the AEC domain (Architecture, Engineering, and Construction).

We present a prototype of a “Learning Forge” designed as an interactive educational environment, combining the use of a “Development Forge” with a “Serious Construction Game” for teaching digital design in architecture. This system, based on the free and open-source forge “Forgejo,” offers a “narrative and immersive” learning environment through the co-construction of a 3D universe and the production of open educational resources (OER). This educational technology (EdTech) also provides an automated assessment system that facilitates tracking of student work.

The “Design Forge” prototype we describe stems from our research on “Educational Forges” and extends these principles as a platform for data generation and exchange for architecture, engineering, and construction (AEC). The prototype integrates version control and continuous integration systems proposed by “Development Forges” to adapt them for design uses and enhance collaborative “multi-user” design possibilities.

Finally, we introduce the concept of an “AI Factory” that combines “Forges” and “Large Language Models” (LLM), and we analyze how these semi-autonomous software production extensions open new perspectives in the context of the construction industry's transformation. These works, rooted in educational experiments and concrete prototypes, aim to illuminate the crucial role of “Forges” in contemporary digital fabrication and in the evolution of project and construction practices.

Document under Creative Commons *by* Milovann Yanatchkov

## Introduction

"Forges" are collaborative web applications primarily used by developers to work on IT projects, which are often complex and involving multiple stakeholders. Over the last decade, these online applications have become widely available, becoming an essential element of the global digital production chain. More recently, these "Forges" are also being used in fields not directly concerned with purely IT issues, such as the educational field. In the field of engineering and architecture, which is of particular interest to us, we also note the emergence of "Design Forges" which we describe in detail in this report. Each of these "Specialized Forges" has its own characteristics, but they can nevertheless be grouped whithin a common software family of a new type: "Domain-Specific Forges (DSF)." In this document, we describe more specifically the use of these Forges in the architecture, engineering, and construction (AEC) sector around three specific themes: "Learning Forges," "Design Forges," and "AI Factories."

The principles and results presented are derived from relatively recent experiments, notably linked to the creation of the French interministerial working group "Forgeons!" (*Let's Forge!*) created in November 2024. As this is a new topic, we also discuss how these "Forges" are expected to evolve in the future, in connection with issues of artificial intelligence and "AI Factories," for example. Finally, beyond the "common foundation" constituting these different "Domain-Specific Forges," we also detail the issues related to the field of construction and architecture, with elements from our previous research in this area. This include pedagogical experiments carried out French architecture schools (ENSA). These elements are the result of a long-term approach, with the first experiments conducted as early as 2009, as summarized in a historical overview available in the appendix.

## Forges: General Principles

### The "Forge": A Metaphorical Object

The use of metaphors is a traditional and widespread practice in the digital world, as it helps to illustrate and make more intelligible often abstract technical concepts. Architectural and urban metaphors, in particular, are numerous: in computing, we speak about "architects" and "urban planners" dealing, for example, with digital "heritage." But references to the natural world can also be found, as is the case with cloud computing or data lakes, for example. The metaphor of the "Forge" is somewhat distinctive, however,

because it does not fit into a broader metaphorical family, such as architectural or natural, but presents itself as a unique "object" in its category. In this case, the metaphor of the Forge is a way of expressing the "productive" nature of the technical object, as it is a kind of "factory" where digital artifacts are produced. As we will see in the rest of this presentation, this metaphorical element is not entirely isolated in its category, as similarities can be found with "software factories" in particular. To clearly differentiate the metaphorical digital object from the real object, we propose to refer to the digital object (the "software Forge") by using the term "Forge" with a capital letter (the same applies to digital Factories).

### Usage and Origin

A Forge is a "collaborative software development tool." This is how it is described in the glossary of the report published in 2023 by the Committee for Open Science (CoSo) entitled "Higher Education and Research Forges in France — Definition, uses, limitations encountered and needs analysis" [2], which serves as our reference in this matter. Since the concept was "invented" in 1999 by the site "sourceforge.org," as the report points out, "software forges have become an essential tool [for all computer developers]" (p. ii). The platform github.com, acquired 7 billion dollars in 2018 by Microsoft, symbolizes this central place in the contemporary digital factory, with over 100 million users [1].

From a purely formal point of view, a Forge is a web application accessible from a browser such as Firefox or Chrome. It is therefore a "platform," in the sense of a web-based software with a "client-server" architecture. On the "client" side of this architecture, we have a web "page" from which users interact with the Forge's features; this is the "user interface" (UI) or "frontend." On the "server" side, we find the "backend" and the Forge's "functional" tools such as the user database, and especially, the version control system that we detail further.

Forges are highly standardized development tools, and their use is now so widespread, and almost "universal," that it would be more accurate to speak of their role in terms of "industrialization of the software creation process" (Ibid. p. ii), but since our purpose is not purely computer science, we limit ourselves to describing their uses from the user side. We will return to these questions in more detail when we discuss the issue of AI Factories.

[1] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

### Version Control

First and foremost, a Forge is an application that allows developers to keep a "history" of their work, that is, to keep track of each modification made to a software in the writing of its source code (let us recall that a software is a "written" or "coded" artifact in a "language" close to human language, and that it is subsequently transformed into a "machine language" executable by a computer, we then speak of an "executable" or a "binary").

From a technical point of view, this history is achieved by a specialized "Version Control System" (VCS) software. These version control software have many features and come in different versions, but today Git is a software that, in particular, stands as a reference. Git is a software developed by Linus Torvalds, the "father" of the Linux operating system (with Richard Stallman, for the GNU part of the GNU/Linux system). Torvalds initially developed Git to aid in the development of Linux, but today it is "universally" used by other operating systems (Windows and Mac).

The main features of Git are as follows: writing a "trace" of software modification (the technical term is "`commit`"), then uploading this modification to the Git server, or downloading this modification from a Git server. The "trace," or "`commit`," is made by creating a file that stores the "differences" between an original state of the file and a modified state. In computing, we speak about a "`diff`" or a "`patch`": a standardized file that contains the "additions" and "deletions" made to a source file.

Keeping a "history" of development is very useful, as it allows, for example, to revert to a previous state if necessary — similar to "undo" features like "Control Z." But above all, it allows the implementation of collaborative work strategies; a fundamental aspect in the creation of complex systems involving many contributors, as we will discuss regarding "collaborative design" in the field of architecture and construction.

We can schematically describe the functioning of this strategy as follows: a software being composed of many files containing many features, we can assign each contributor of a development team a specific mission resulting in a set of specific modifications. To allow all contributors to work asynchronously and in parallel, we use the version control system to create as many "derivations" of the source code (`fork`), in the form of "branches" (`branch`), in which modifications can be made in isolation, and without risk of conflicts with other developers. At the end of the development process, each of the new features is "merged" into the main trunk (`trunk`) through a merge operation (`merge`).

### Collaborative Platform

As we have just seen, a Forge is therefore an application based on a version control software (the *Git* of GitHub) because it makes possible collaborative work potentially involving a large number of actors. More precisely, we can say that the concept of Forge appeared when a version control system (VCS) was first associated with a web interface, accessible from a browser. This was the "invention" SourceForge which therefore gave the current term "Forge".

Originally, SourceForge is as a software distribution platform, that tracked and updated the various versions of the software available for download. The innovative idea of this first Forge was to include a discussion system — a "forum" — allowing a community of users and developers to interact more easily (Ibid. pp. 6-7). Backed by a version control system, the principle of Forge will therefore gradually consolidate around the many collaboration tools both on the developer side and on the user side: discussion forum for bug tracking systems and interaction with the user community, but also documentation directly linked to the source code, and other communication and project tracking features such as issue management for example.

A Forge is therefore much more than just a technical tool, because in many respects, this type of platform is comparable to a "social network" (Ibid. p. ii). The collaborative aspects of this "productive platform" therefore make all its specificity, between productive technical tools on the one hand, and spaces of interaction between users, producers, and creators, on the other hand. In this sense, a Forge is a "socio-technical" object to use the terms of Benoit Combemale, research director at the Inria Diverse laboratory.

## Domain-Specific Forges

Historically, the invention of the Forge concept is linked to the SourceForge platform, but in reality, it is GitHub that has turned this simple idea into one of the pillars of the contemporary software industry. When Microsoft acquired GitHub in 2018, it was already a nerve center for the free software community and open-source production lines. This significant date marks, on one hand, the transition of Git—and the concept of Forge—to the status of a universal "pivot" in the world of software development, but it also represents a pivotal moment for the open-source movement, with a significant evolution of the Windows ecosystem towards the open-source paradigm [2].

[2] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

After fiercely opposing the free software community for more than two decades, Microsoft is ushering in a new era with its acquisition of GitHub, the historic home of free software. However, despite the many signs of openness from the Redmond company, this decision is not without consequences for the open-source world, which remains extremely wary of a multinational with often undemocratic practices. For the world of Forges, this turning point heralds the emergence of "alternative" Forges to GitHub, and with it, the advent of "Domain-Specific Forges."

### Open-source Forges

In this context, we can argue that 2018 also marks the emergence of Gitlab as a true competitor against GitHub's hegemonic position. GitLab is a Forge whose operation is very similar to GitHub—version control, communication and planning tools, automation chains, etc.—but it has one major difference: the platform itself is available as free software. For Linux developers, and the free software community more generally, this open-source platform represents an ideal alternative to GitHub now in the hands of Microsoft and the Windows world.

Being open-source, GitLab becomes the symbol of a significant shift in the Forge universe with, on one hand, the possibility of self-hosting Forges on dedicated servers (*on premise*), and on the other hand, the opportunity to modify and adapt these Forges to various contexts. With the possibility of transforming these platforms for specific use cases, the concept of "Domain-Specific Forge" (DSF) therefore emerges, in the same way that we speak of "Domain-Specific Languages" (DSL).

The emergence of Domain-Specific Forges is therefore correlated with the advent of free and open-source Forges. But in this context, and as will be discussed later, it is important to distinguish these Forges in terms of their "levels of openness." In this regard, the case of GitLab is symptomatic of a significant evolution of the notion of openness in the world of free software, as only the core of GitLab, essential to its operation, is available as a free software. This "variant" of free software is said to be "open core," in the sense that some "premium" services, offered outside the purely open-source base, are "proprietary" and available only in a commercial form.

Since GitLab is not entirely open, we are particularly interested in Gitea, a project developed over the past decade as a "clone" of GitHub. More precisely, we present "Forgejo" project, a free Forge derived from Gitea since October 2022 [3].

[3] The "T" in the acronym GPT for *Generative Pretrained Transformer*.

## Let's Forge!

As an Associate Professor in a national school of architecture (ENSA), we are interested in Domain-Specific Forges for at least two reasons: as a teaching tool and as a research instrument. In this context, notably, we developed the educational platform Sloyd, an Educational Forge in the form of a Domain-Specific Forge based on Forgejo, which we describe later in this document. As we will see later, with the creation of a Domain-Specific Forge, the issue of modifying its user interface (UI) arises. This is why we are particularly interested in Forgejo, because unlike GitLab, its interface is customizable as are the functionalities presented to the user.

In terms of adaptability, Forgejo and Gitea truly stand out as they use a template system that allows modifying or creating interfaces "on demand." This Forge being a fully-fledged free software, and moreover perfectly customizable, naturally interests an increasingly large community of users and developers, and particularly some public stakeholders in a context where "digital sovereignty" has become an important issue, at least in France. It is in this context that we initiated the creation of the French interministerial working group Forgeons! in November 2024.

"Forgeons!" (*Let's Forge!*) is a working group centered on Forgejo, bringing together stakeholders from the Domain-Specific Forges community in France, mainly public agents. This thematic group complements a number of pre-existing initiatives such as the "Forges" thematic group of the "Committee for Open Science" (CoSo) [4] led by Daniel Le Berre, a researcher at the University of Artois. In March 2025, we organized a first working day that brought together a wide variety of stakeholders on these themes. Here is the report that we wrote and published on the official website for Open Source within the French Interministerial Digital Directorate (DINUM).

### Citizen, Educational, and Creative Forges

"Forgeons!" Day on March 13, 2025, report published on code.gouv.fr

[4] See https://modelcontextprotocol.io/docs/getting-started/intro

**Figure 1:** "Forgeons!" Day at the Public Transformation Place in Paris

The inaugural session of the "Forgeons! (*Let's Forge!*)" working group was held in person on Thursday, March 13, 2025 in Paris. The agenda included "Educational, Creative, and Citizen Forges" in the first half of the day, and "Forges for Engineering and Architecture" in the afternoon. In terms of digital autonomy, free and open-source Forges are today essential elements to ensure the sustainability and resilience of development infrastructures; but they also represent an important issue outside the strictly speaking IT sphere, as was observed during this day.

"Forgeons!" is an interministerial working group focusing on Forgejo, the free and open-source forge associated with the Codeberg project, an alternative platform to GitHub and supported by a collective of citizens rooted in Europe. The group federates public agents with very diverse profiles: Education, Research and Higher Education, Culture, as well as CNRS, INRIA, CEA, and the National Assembly; it was created at the end of 2024 thanks to the BlueHats network led by the Free Software Mission (MLL) of the Interministerial Digital Directorate (DINUM).

At the opening of the workshop, the "Citizen Forge" project Tricoteuses.fr ("Law Under Git") demonstrated the numerous applications of "Domain Specific Forges", as in the leg-

islative field in this case. The prototype, also presented during the Rules-as-Code day, illustrates how a versioning system based on Git, associated with a continuous integration system, facilitates the monitoring of a constantly evolving body of legislation.

In the more established field of "Educational Forges," the Forge of Educational Digital Commons supported by the Ministry of National Education, continues to establish itself as a "forum" and "market place" enabling the federation of numerous educational projects such as Open educational resources (OER), source codes, and innovative learning applications.

It is in this context that serious games and Educational Forges have now become objects of study in its own right, as demonstrated by the GTNum Forges supported by the LIUM laboratory. How to adapt tools initially designed for computer engineers for a non-specialist audience? This is essentially the question posed in a thesis currently being prepared within the laboratory.

#### Forges and Digital Twins

Surprisingly, all the questions raised during the morning, in search of the specific "invariants" of the different applied Forges, resonated perfectly during the "Forges and Digital Twins" session in the afternoon. During this session, the presentation of two Forge prototypes intended for engineers and architects showed a strong convergence of the initial results from these researches.

As part of the work conducted by the Diverse/Inria laboratory, Forges are studied for their roles as "interfaces" between areas of expertise in engineering professions involving a large number of stakeholders, such as in aeronautics, for example. In the same vein, a prototype of an Educational Forge for Schools of Architecture served as a testing ground for the development of GitAec, a prototype of a "Design Forge."

For more information, see the resources accessible from the event page, as well as the BlueHats GitLab Day and the BlueHats Forgejo/SourceHut Workshop.

## Educational Forges

As we saw in the previous pages, the concept of the Forge "emerged" at the turn of the 2000s, alongside the Internet. Over the past twenty-five years, SourceForge and then GitHub have gradually become essential platforms for affirming the central role that free

and open source software plays today. Now occupying an essential place in the contemporary technological ecosystem, we can therefore assume that the Forges played a key role in the emergence of the digital world as we know it.

And yet, despite this central position, Forges are technical objects that are still little studied and, ultimately, quite poorly defined. Research is still necessary to better understand what their "invariants" are. The Domain-Specific Forges that we propose to describe in this report are therefore an opportunity to shed a little more light on this subject, even if in a "roundabout" way, by questioning the role and nature of these Domain-Specific Forges, "derived" from Development Forges, and of a new type.

## Educational Technologies

Domain-Specific Forges, such as the Educational Forges we describe here, have not yet been formalized because it is still a largely experimental subject, and examples are relatively rare to date. We will first provide an overview of this issue by distinguishing between two educational contexts: university and higher education on the one hand, and primary and secondary education on the other. But before doing so, we think it would be interesting to briefly introduce "educational technologies," as Educational Forges seem to belong to this family of software.

The term "educational technologies" (Edtech) refers to computer tools used for learning. In this field, there is a very active market of companies offering applications for primary and secondary education. In the university education sector, there are emblematic examples from major American universities, sometimes with technology transfers to the private sector [5]. These tools can be of various types, accessible either from traditional desktop computers or more oriented towards "mobile uses" [6]. We are not seeking here to delve into the subtleties of an emerging ecosystem, but we believe it is interesting to place this general context in order to help us better define the role of Educational Forges in relation to the rest of existing EdTech. In this regard, Moodle, which is very widespread in the academic world, is an element that may serve as a reference in this context.

## Academic Forges

With these preliminary remarks in mind, it seems natural to begin by discussing Educational Forges in the academic context, as we rely primarily on the CoSo "Forges" report

[5] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

[6] The "T" in the acronym GPT for *Generative Pretrained Transformer*.

[11] for the definition of principles. In this document, and after discussing the functioning of "Development Forges" in their primary software creation missions, the report refers to very common practices today, consisting of using Forges to generate websites from documents generally written in Markdown format [7]. More directly related to academic practices, the report also mentions "editorial" uses concerning the production of PDF documents from input written in Latex (Ibid. p. 6). In this regard, we could speak of "Editorial Forges" when it comes to producing and disseminating all types of informational contents, both educational and scientific.

The issue of Academic Forges is then addressed in a very specific context, because the authors of this report, computer scientists, are themselves software producers. It is in this context, in particular, that the issue of "research-based free software" [8] (Ibid. p. 7) is discussed because it raises the question of patents and copyright in sometimes complex environments involving public and private stakeholders. More generally, it is an issue that raises the problem of "digital sovereignty", and the degree of technological autonomy of a European States, in a context today dominated by China and the United States. A point raised by the "Let's Forge!" working group as we have seen previously. Therefore, and at a time when many challenges are arising around AI and large language models (LLM), we better understand the stakes for the world of research and education when it comes to discussing the crucial role that forges play today.

### Open Education

In a world increasingly dominated by technology, we understand how these "infrastructures" such as Forges and Large Language Models (LLM) are critical elements in a given geographical and cultural context. This is all the more true when it comes to educational technologies. In this context, "open models" derived from free and open-source software are now fully part of the digital strategies implemented in the academic world. In this regard, and among the 26 measures established in 2023 by the French "Digital Committee for Student Success and Institutional Agility" (COREALE) of the Ministry of Higher Education and Research (MESR), the principle of "open education" and the need to "strengthen access to and the promotion of open educational resources" [12] (p. 8.) are established.

In a report published by this committee in February 2025 [4] the principle of "open education" is defined according to the terms given by the European Commission in 2016:

[7] https://codeberg.org/rvba/osai
[8] https://www.primeintellect.ai/

> "Open education is an educational practice, often involving the use of digital technologies. It aims to facilitate access to education by removing barriers and offering accessible, rich, and customizable learning to as many people as possible. It relies on multiple ways of teaching, learning, building, and sharing knowledge." [9]

It is then specified that "open educational resources (OER) are one of the tools enabling the opening of education" (Ibid. p. 8).

### Open Educational Resources (OER)

In this same report, the principle of "Open Educational Resources" (OER) is stated based on the definition given by UNESCO (Ibid. p.8):

> Open Educational Resources are teaching, learning, and research materials that, thanks to appropriate tools like open licenses, can be freely reused, constantly improved, and adapted by third parties for educational purposes [10]

We understand quite evidently that the principle of OER is directly derived from the culture of free software and Creative Commons, but applied here to the specific field of education. This proximity therefore implies issues that are close to those of open-source software in general. This is why the report discusses the role of Forges regarding the "life cycle" of OER:

> On the other hand, it also allows thinking about setting up collaborative spaces (like the forges [a] used for software, and facilitating collaboration for communities wishing to reuse the resources stored on a platform. Thus, thinking in terms of life cycle allows not to forget any stage of a resource's life, from its creation to its permanent archiving, or even its deletion. [4] p.18.
>
> [a] https://fordj.org

When making recommendations, the report mentions Forges in terms of "collaboration space" and "versioning" as we mentioned earlier in terms of history:

> Provide collaboration spaces to promote the reuse and modification of resources, on the principle of forges used in free software and which also allow managing the versioning of resources and access to their native format (source code if applicable) to modify it (Ibid. p.20).

Thanks to this COREALE report, and more generally to the positions of principle taken by the European Commission and UNESCO, free software and open educational resources are now fully recognized as an important lever in the educational world.

### Forges of Educational Digital Commons

To now introduce the question of Educational Forges in the world of primary and secondary education, we can cite, still in this same report, the creation of the "Forge of Educational Digital Commons" ("ForgeEdu") in the fall of 2023 [5] (Ibid. p.17). Here is how the project is presented in the glossary annexed to the report:

> **Forge of Educational Digital Commons**: an online space designed to promote the creation, collaboration, and sharing of open educational resources in primary and secondary education. (Ibid. p.24)

The "Forge of Educational Digital Commons" is a project carried by a collective of teachers [9] within the Ministry of National Education (MEN) since 2022 [10]. Originally, it is a GitLab "instance" [11] allowing the pooling of IT resources among teachers on a common server. Today, the project brings together, beyond the Forge, an entire community of teachers in the creation of free educational software and open educational resources (OER).

The "digital commons" hosted on this Forge are, in the COREALE glossary, defined as follows:

> **Digital Commons**: a term chosen by national education to designate free educational resources, that is to say, a set of digital resources produced and managed by a community. They are therefore shared and collective. (Ibid. p.23)

In October 2024, a Forgeathon [12] organized by the "ForgeEdu" community allowed for the exchange of initial feedback between stakeholders of Educational Forges, mainly teachers, and researchers from Inria LearningLab. These exchanges notably allowed discussing the need to evolve the user interfaces (UI) of the Forges to adapt them to the educational context. On this particular point, it was recognized that Forgejo offered greater opportunities compared to GitLab because its interface is not customizable [13].

[9] For technical reasons, the Go sources are temporarily hosted on gitaec.org presented later.
[10] See https://codeberg.org/forgejo-contrib/delightful-forgejo#forks
[11] See https://gitaec.org/rvba/docs/wiki/Zurich-hackathon
[12] See Industry Foundation Classes
[13] https://github.com/brunopostle/ifcmerge

### Thematic Research Group

The question of user interfaces is one of the subjects addressed by a thesis in preparation within the framework of the "GTNum Forges" with the LIUM laboratory of Le Mans [Forest, Marne, and Marfisi-Schottman [6]]. In these preliminary works, the interfaces of Educational Forges are put to the test of "learning management systems" (LMS) that have been specifically developed for pedagogical uses:

> More generally, Zagalsky *et al*. argue that traditional software forges were not designed to share educational resources and do not have features adapted to the needs of teachers, like those offered by learning management systems (LMS) such as Moodle [9]. Moreover, Glassey [19], in his study on the use of GitHub by teachers, joins these conclusions. [Forest, Marne, and Marfisi-Schottman [7]] (Ibid. p.8)

Beyond these purely ergonomic considerations, the authors also refer to the questions raised by the use of version control systems (VCS) in a non-IT context, while mentioning the particular case of computer science studies:

> The adoption of version control systems by teachers is studied by two authors within the *corpus*. According to Glassey, there is a burgeoning literature that reports on the benefits, challenges, and experiences of adopting version control systems within a learning environment for teaching computer science [19].

In the case of architectural studies, which are more directly relevant to our study, we note here the specificity of certain training courses involving "advanced" digital content, such as generative design, in which connections with these specialized digital tools are rather sought after because they prepare for "hybrid" profiles of "user-developers" as we described in an article published in 2022 [23].

### Educational Forges: Principle Definition

At the end of this initial overview of Domain-Specific Forges in the educational and academic fields, we offer some lexical clarifications. As mentioned in the glossary appended to this document, we propose a description of Forges in a "taxonomic" form in order to systematically describe all new technologies and uses related to Forges. At the "root" of this classification are "Development Forges", which are the "original" Forges from which derivatives are created that we call "Domain-Specific Forges."

Based on this common core, we propose to refer to "Educational Forges" for the field of education, primarily primary and secondary education. For higher education, it might

be more appropriate to use the term “Academic Forge” or “Research Forge,” if we stick to the specificity of these Forges, as is the case for the work carried out by Daniel Le Berre and the introduction of an ORCID profile (*Open Researcher and Contributor ID*) for GitLab, for example [14].

Finally, as we mention later, we propose the term “Learning Forge” to designate Forges with an educational and/or academic purpose, specifically intended for students. We propose a specific term in this regard because these Forges are modified to serve as a learning environment, unlike Educational Forges or Academic Forges, which are “limited” to the hosting, production, and sharing of Open Educational Resources (OER) and research software.

## Learning Forges

We now describe a prototype of an “Learning Forge” developed as part of our teaching in architecture schools (ENSA). As mentioned previously, Educational Forges are generally intended for teachers and researchers to produce educational digital commons or open-source research software. The particularity of the prototype we present now is that it is aimed more specifically at students. As we will describe later, this project aims for a number of specific educational objectives reflecting a series of observations from our teaching practice and responding to emerging educational objectives corresponding to new practices specific to the field of architecture and engineering (AEC) [15], notably “generative design.”

We propose to describe this type of Educational Forge as a “Learning Forge” because, as we demonstrate, the Forge we have developed is based on a serious game and an interaction system that gives it characteristics similar to those of “educational software” (*EdTech*). In this sense, it is much more than a simple Educational Forge intended as a tool for sharing digital commons. Before describing its operation in detail, we provide some information about the architectural professions, the academic context, and the educational choices that led us to develop this prototype.

### Generative Design

The Learning Forge prototype we propose to describe is an innovative concept, adaptable to various educational contexts. However, it was developed in a very specific context,

[14] See https://gitaec.org/rvba/docs/wiki/Zurich-hackathon

[15] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

with specific educational objectives, particularly related to architectural design. Furthermore, since it is directly related to the issue of "Design Forges," which we will discuss later, we feel it is important to present this context now.

This specialized Forge was developed to support the creation of an "Introduction to Digital Design" course in the field of Mathematical and Computer Tools for first-year undergraduate students. This course aims to teach the fundamentals of digital technology for architecture and prepare students for future careers and specialties in this field. "Digital design" in architecture is an emerging field and a highly specialized professional practice, yet central in a context of environmental and technological change. Given the relative breadth of the topic, we will limit ourselves for now to presenting the issue of "generative design," which is at the heart of this approach.

"Generative design" is a design practice that makes advanced use of digital technologies by leveraging the "generative effects" of programming. In short, it involves the designer describing their design intentions in terms of "generative functions," that is, "parametric objects" that, like a "smart mold" in a foundry, would be capable of producing complex objects on demand, based on input parameters.

At the heart of this practice is knowledge of digital mechanisms, particularly programming, and more specifically, learning the `Python` language, which is now taught in secondary schools in France. To appreciate the importance of this new content in science education in schools, it should be noted that a one-off event such as "La Nuit du Code" (*Code Night*) brought together more than 14,000 students in 2025 [16]. In the field of architectural studies, Python is now a key language for generative design and also relates to the issue of AI factories, discussed later in this document. For a more general introduction to the issue of generative design, consult the educational resources we have developed (in French) on the site codeatlas.cc.

### A "Screenless" Digital Pedagogy

As we will see later in this presentation, the main objective of this Learning Forge is to create an "online workshop" for students, using a "flipped classroom" approach [17] and geared toward the production of Open Educational Resources (OER) by the learners themselves. In developing this platform, we aimed to create a "dissociated" pedagogical framework, between moments shared with students and teachers, in person and without a computer, and moments of independent teaching, remotely, with a computer.

[16] The "T" in the acronym GPT for *Generative Pretrained Transformer*.
[17] See https://modelcontextprotocol.io/docs/getting-started/intro

Somewhat provocatively, we proposed describing this “dissociated” digital pedagogy as a “screenless digital pedagogy,” at least between teachers and learners.

We started from a radical observation: classrooms equipped with computers may be useful as “IT self-service,” but they are not, in our opinion, suited to learning activities [18]: the computer screen and the multiple contents accessible online are so attractive that they capture the learner's attention that it is impossible to teach in a traditional, “analog” manner in a confined room with so many “screens” interposed between brains.

The prototype we developed is a new kind of software, in the sense that it corresponds neither to a classic Educational Forge nor to a traditional MOOC (Massive Open Online Course), even if it were “small” [19]. Using this platform, based on open standards and freely composable software components, we seek to explore a new type of learning space, more interactive than MOOCs, which are generally based on static content (mainly audio-visual), accessible in often “closed” educational spaces.

From a learning methods perspective, with this prototype, we sought to establish an “immersive” and interactive space that functions as an “open” space and promotes “mutual teaching” learning methods such as the “learning by teaching” method [20] formulated by Jean-Pol Martin [13] [1]

### A Serious Construction Game

Sloyd, the Learning Forge prototype we are now describing, is the result of a series of elements developed since 2009 in various educational and research contexts. The platform combines an Educational Forge and a serious game based on an interactive system that we will describe later. It is a “serious construction game” whose origins date back to an initial discovery workshop for Blender, a free 3D modeling software [20]. As part of the “Make Art” festival, an event dedicated to creative free software, we organized a learning workshop in 2009 on the theme of “open source architecture” [19]. This is where the idea of discovering 3D modeling by interacting in a “common space” was born. The principle is very simple: each learner has a “buildable square,” like a plot of land, on which they can build within a larger virtual environment. These plots are regularly assembled to form a sort of common urban space in the form of a large rectangular grid.

A few years later, in 2016, we adopted this principle in our teaching at the ENSA La Villette school of architecture in Paris, under the name “Alphaville” [21]. It was in this context

[18] https://codeberg.org/rvba/osai
[19] https://www.primeintellect.ai/
[20] https://fordj.org
[21] For technical reasons, the Go sources are temporarily hosted on gitaec.org presented later.

that we began developing a system capable of automatically assembling learners' plots, a task previously performed “manually.” As we will detail, this system will gradually be enriched with numerous features until it becomes a sort of “game engine” comparable to the mechanisms encountered in the design of a “video game,” although its complexity is not comparable. During this gradual evolution, the serious game will be enriched to become a prototype of a Learning Forge and will serve as a basis for research on Design Forges and architectural generative modeling.

### Alphaville

Alphaville is the name of the “serious construction game” and its “game engine” programmed in Python. Technically, this game engine is in many ways a prototype, continuously developed before and during the teaching sessions. It can be described as a “testing lab” or “proof of concept” that is still partially developed, and whose avenues have not yet been fully tested. As we will see later with Design Forges, this experiment has gradually opened up questions that are increasingly removed from a purely pedagogical context and serve as a basis for research more directly focused on the generative designing of architecture. The prototype was developed in two distinct phases: a first version was created at ENSA La Villette from 2016 to 2020, and a second, more advanced version, developed starting in 2020 as part of our teaching at ENSA Nancy, to support a more specific course, “Introduction to Digital Design”.

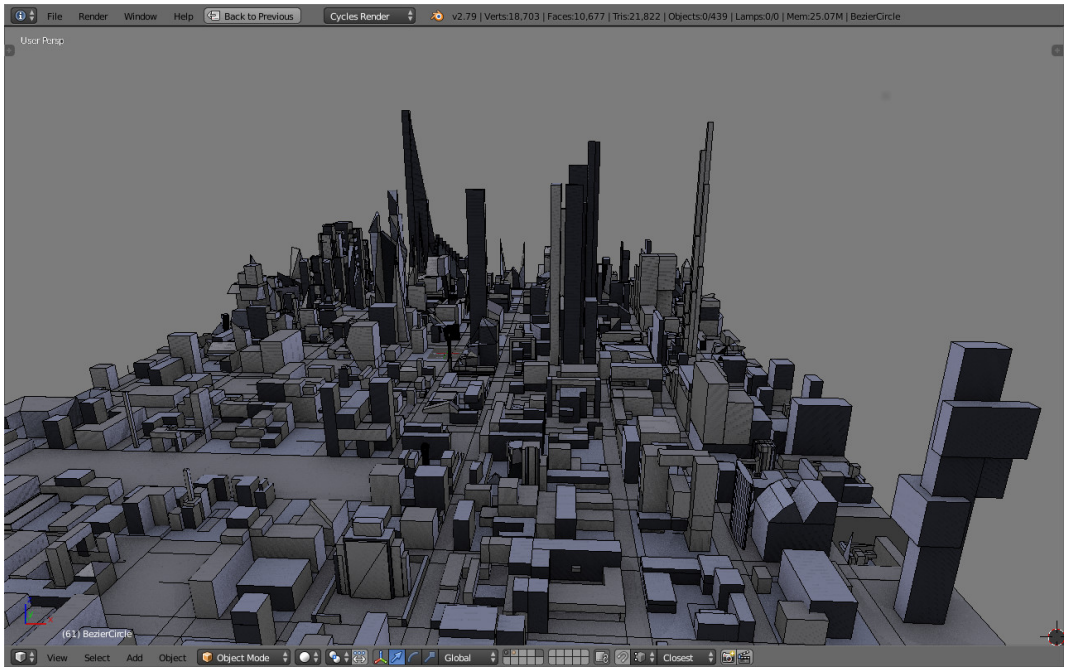



**Figure 2:** Alphaville in 2017 at ENSA La Villette

From a purely pedagogical perspective, the notable difference between the “La Villette” and “Nancy” versions concerns narrative. By offering a course in digital design for architecture that is not limited to a simple technical manual, but rather constitutes an introduction to the digital and technical culture specific to architectural design, we chose to transform the initial serious construction game into a culturally richer environment, particularly through narrative.

Several reasons motivated this choice. While it may be necessary to simply explain to readers who might be confused by “the story of Alphaville” and the place of narrative within it, it is important to keep in mind this essential point: developing an architectural project, as surprising as it may seem to a reader without prior knowledge of architecture, consists of “telling a story.” Places are open books, and architectures are chapters in a narrative that encompasses them.

Finally, by teaching “generative design,” the narrative principles of Alphaville, as strange as they may sometimes seem, aim to introduce the algorithmic, constructive, and also regulatory principles that generally govern architectural and urban design procedures. In short, the narrative proposed by this serious game seeks to highlight, even in an offbeat and humorous tone, the preponderant role that the code will be called upon to play, as we will explain later.

## Pygram

The second version of the prototype developed at ENSA Nancy contains an introductory module for 3D programming with Python. Python is a simple and universal "scripting language." Present in most digital environments and design software, its importance is now such that it was introduced into the French National Education curriculum in 2018. In 2022, we offered the first Python teaching modules, before integrating them more specifically into the Learning Forge game engine in 2023.

Based on the algorithms developed for the game engine used to assemble Alphaville and assess student work, as described below, we also introduced a Python-based "micro-language" in 2023, designed to teach the fundamentals of coding, particularly 3D programming. Pygram is a small language designed to teach students how to manipulate mathematical objects such as vectors and perform simple geometric operations to familiarize themselves with the principles of analytic geometry, that is, with objects located in space using "x", "y", and "z" coordinates.

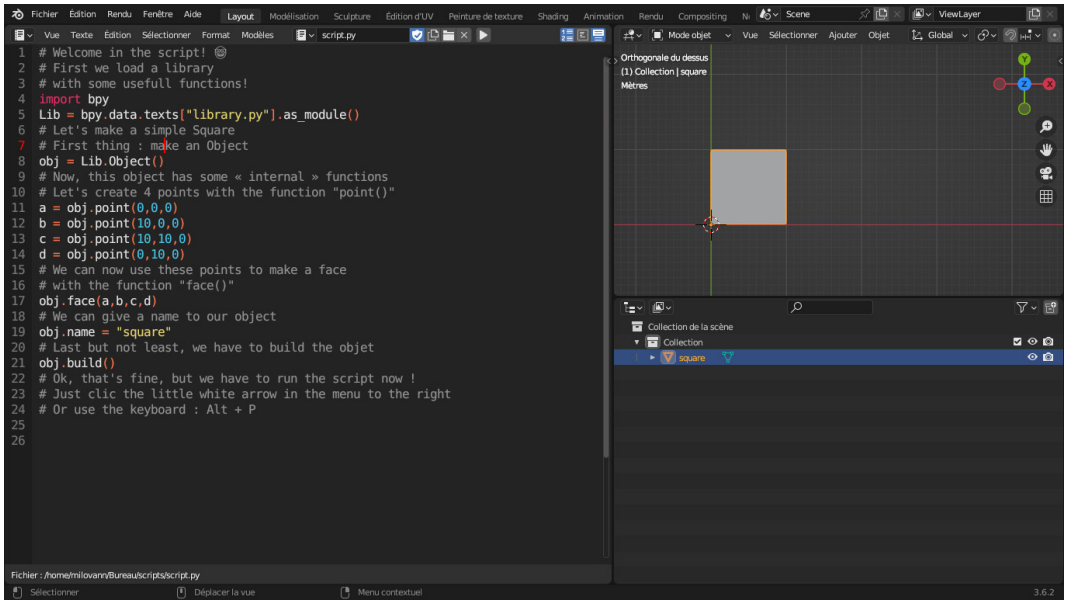



**Figure 3:** Pygram: screenshot of the 3D Blender software

The term "micro-language" is somewhat misleading here, as Pygram has no syntax of its own and is neither compiled [22] nor interpreted [23]. Rather, it is a "class" [24] developed to simplify the creation of 3D objects in Blender. However, we have chosen to refer to

[22] See https://codeberg.org/forgejo-contrib/delightful-forgejo#forks
[23] See https://gitaec.org/rvba/docs/wiki/Zurich-hackathon
[24] See Industry Foundation Classes

Pygram as a “language” because it is intended to emphasize computer languages in general, as is the case with learning the Markdown language, which is an integral part of the teaching.

### Sloyd

It was precisely as part of the practical work for the new Introductory Digital Design (ICN) course in 2022-2023 that we significantly upgraded the initial version of Alphaville [25] to form the basis of a serious game interactivity engine, including a system for evaluating student work. This system, called *Alphengine*, operates on the basis of Blender's Python API and primarily involves collecting files produced by students, opening them, and exploring their content. The first features of *Alphengine* [26] included automated checks regarding, for example, compliance with certain nomenclature, the presence of 3D objects, and their dimensions.

It was during the summer of 2023 that we combined the *Alphengine* interactivity system with the free and open source Forge Forgejo to create the Learning Forge as we describe it today [Yanatchkov [21]]. Before being a web platform, the first version of the software, in 2022-2023, was a resumption of the script initiated in 2017 at ENSA La Villette and ran on the school's local network. In addition, the game engine did not yet work autonomously, but had to be updated manually. With the creation of Sloyd and this first Learning Forge, our idea was to transform the educational experience into a real interactive online game, running continuously and accessible from outside the school. We named this experimental platform “Sloyd” in reference to a pedagogical practice in the Nordic countries [27].

Technically, the platform works through a system of webhooks attached to a Forgejo instance and triggering *Alphengine* events. Specifically, each time a student submits a 3D file or modifies a text file, such as Markdown, directly from the Forge interface, a call to *Alphengine* is triggered by the webhooks. With each new event, the city evolves based on the content of the 3D files produced by the students. The constraints imposed by the serious game are then evaluated, and points are awarded to the players based on their responses.

It is precisely within the framework of the practical work (TD) of the new course on introduction to digital design (ICN) in 2022-2023, that we significantly evolved the initial version of Alphaville [28] to make it the basis of an interactivity engine of the serious game

[25] https://github.com/brunopostle/ifcmerge
[26] See https://www.inriastartupstudio.fr/portfolio/
[27] Recall that Forgejo is a “hard fork” of Gitea
[28] https://github.com/brunopostle/ifcmerge

including notably a system for evaluating the work done by the students. The system, named *Alphengine* [29], operates on the basis of Blender's Python API, and mainly consists of collecting the files produced by the students, opening them, and exploring their content. The first features of *Alphengine*, included automated checks concerning, for example, compliance with certain nomenclatures, the presence of 3D objects, and their dimensions.

It was during the summer of 2023 that we combined the *Alphengine* interactivity system with the free and open-source Forge Forgejo to make it a Learning Forge as we describe it today [21]. Before being a web platform, the first version of the software, in 2022-2023, was a resumption of the script initiated in 2017 at ENSA La Villette and operated on the school's local network (LAN). Moreover, the game engine did not yet operate autonomously but had to be updated by a "manual" intervention. With the creation of Sloyd and this first Learning Forge, our idea was to transform the educational experience into a real interactive online game, operating continuously, and accessible from outside the school. We named this experimental platform "Sloyd" in reference to a pedagogical practice from Nordic countries [30].

Technically, the platform operates on the basis of a "webhooks" system attached to a Forgejo instance and triggering *Alphengine* "events." Concretely, each time a student submits a 3D file or modifies a text file, such as Markdown, directly from the Forge interface, a call to the *Alphengine* is triggered by the webhooks. With each new event, the "city" evolves based on the content of the 3D files produced by the students. The constraints posed by the serious game are then evaluated and based on the responses, points are awarded to the players.

## Design Forges

We now introduce the principle of the "Design Forge." This new type of Domain-Specific Forge, and the prototype GitAec that we propose to describe here, are a continuation of the experiments carried out during the development of the Learning Forge Sloyd described previously. Like the Learning Forges, this is still an emerging, unformalized topic. We use the term "design" in a very general sense, even though our main field of study is architecture and urban planning. As we will see later, this is the term we have used to describe next-generation "design platforms." In English, *design* carries a double meaning: form and intention, process and result. It is with this open conception of *design*,

[29] See https://gitaec.org/rvba/docs/wiki/Zurich-hackathon
[30] See https://www.inriastartupstudio.fr/portfolio/

also including engineering as we will see, that we propose to explore this new type of Domain-Specific Forge.

### Design Platforms

As we mentioned in the section on Learning Forges, the prototypes presented in this study are part of an exploratory approach to generative digital design for architecture. In this context, two elements are crucial: first, digital languages such as Python, and second, "documentation" languages such as Markdown. Finally, network technologies [31], the Web, and online applications constitute the framework within which these new practices specifically emerge.

In a previous study [23], we showed how second-generation design tools (BIM, *Building Information Modeling*), whose use became widespread in the 2010s, were evolving towards "networked" uses, directly accessible from the Web. We have proposed the term "design platform" to describe this third generation of design tools, which is a defining characteristic of the 2020s [32]. Unlike second-generation tools (BIM 1.0), which are "desktop software" usable by a single user, design platforms (BIM 2.0) are online applications designed to be used by multiple users.

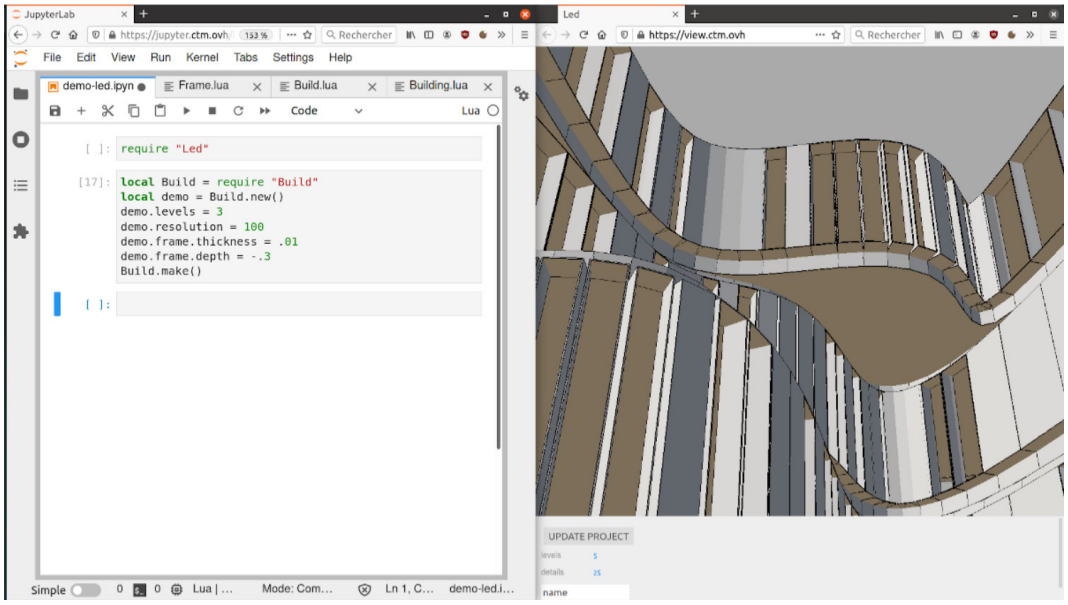



**Figure 4:** Led: Design platform prototype created with Jupyter

[31] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

[32] The "T" in the acronym GPT for *Generative Pretrained Transformer*.

The main innovation of this new type of application lies in the ability to design in a "network." Like many professional practices that have already benefited from the networking of data, and sometimes work tools, via online platforms, third-generation design tools offer new possibilities that meet the needs and imperatives of collaborative work. It is in this context that Forges appears to us to be an ideal candidate to serve as an infrastructure for these new types of design tools.

As we mentioned in the introduction, architectural metaphors are numerous in the field of computer science. Although the architectural metaphor is debatable in some respects [33], it seems to us that in this case, and through a "mirror effect," the software engineering practices associated with Forges could apply and advance the design methods of architecture and construction engineering (AEC).

## Architecture as Code

It is within the framework of our research into the use of "code" in architecture—which we refer to as "écriture numérique de l'architecture (*digital writing of architecture*)" in French, or in English for *Architecture as Code (AaC)*)—and more specifically in the radically new dimension of architectural design through programming, instead of drawing (or "direct modeling" of forms in a BIM 1.0-like space), that we began to focus on the Forges as a "place" for the production of this "writing."

We hypothesize that the Design Forges, combined with forms of digital writing adapted to architectural design, could be a key element in the evolution of third-generation design tools (design platforms), and that this could potentially disrupt, once again, the traditional "order" of design, as already observed during the transition from the first to the second generation. Just as second-generation tools (BIM 1.0) have effectively "reversed" the order of representation by favoring the production of a virtual 3D model from which 2D documents (plans, sections, elevations) are automatically generated, third-generation tools (BIM 2.0), combined with forms of digital writing, could potentially "reverse" the order of modeling by favoring the production of "intelligent" parametric objects from which 3D models (BIM models) are generated.

We hypothesize that the digitization of design tools should lead to a transformation of the order of modeling and that, in this context, "Architecture as Code" should be "primordial" to "models," being the "source" of their generation. Following this paradigm, we wish here to highlight the paradigm of "generativity" in the sense attributed to the "generative grammars" described by Chomsky. [34].

[33] See https://modelcontextprotocol.io/docs/getting-started/intro
[34] https://codeberg.org/rvba/osai

### Fordj

After the creation of the Sloyd Learning Forge prototype in 2023, and its "testing" by students [35], we decided to group together under the banner of the Fordj [36] project the various software components that made it possible to create this platform, with a view to developing prototypes of Learning Forges and Design Forges on this basis.

Fordj brings together the various modules enabling the development of Domain-Specific Forges based on Forgejo, including the sources for *Alphengine* (coded in `Python`) and the sources enabling the development of Domain-Specific Forges based on Forgejo [37] (coded in `Go`).

Fordj is therefore a derivation of Forgejo (a *fork*), "officially" referenced as such by the developer community [38]. The source code for this fork is developed using the principle of a "soft fork" of Forgejo. In Git jargon, a "soft fork" is a fork of source code intended to provide additional functionality to the underlying project. This "soft fork" development method allows for the development of a fork while continuously monitoring the evolution of the base software over time. This method differs from a "hard fork," which decides to deviate from the reference project. This is particularly the case for Forgejo with respect to Gitea.

Forgejo also contains a prototype rewrite of the *Alphengine* engine from Python to Go in order to future-proof the project and better integrate it with the Forgejo code, developed in the latter language. This reimplementation project under the name "Playground" has a dual purpose: to serve as an engine for serious construction games, and at the same time, to serve as a testing ground for the development of a "generative engine" for Design Forge, in combination with "continuous integration services" as we now describe it within the GitAec project.

### GitAec

GitAec is a Design Forge prototype that uses and extends the results obtained during the development of Learning Forge Sloyd. As mentioned previously, there is a convergence here between the development of a game engine for Learning Forge and that of a "generative engine" for Design Forge. In this case, this convergence mainly concerns the development of a 3D file visualization tool (a *viewer*) and the development of a generative engine based on a continuous integration system that we detail later.

[35] https://www.primeintellect.ai/
[36] https://fordj.org
[37] For technical reasons, the Go sources are temporarily hosted on gitaec.org presented later.
[38] See https://codeberg.org/forgejo-contrib/delightful-forgejo#forks

Since Sloyd is based on the serious game *Alphaville*, we sought to integrate a 3D file viewer to easily visualize the evolution of the city directly from the web platform. We developed an initial prototype based on Three.js to display files in `Obj` format based on conversions made from native files modeled with Blender. Subsequently, during an AEC hackathon in Zurich in February 2024 [39], we revisited this initial draft by choosing to integrate the open-source viewer now developed under the name ThatOpenCompany [40]. A demo file on the platform demonstrates that the Forge is capable of displaying BIM files in `IFC` format [41] as it natively does with PDF documents, for example.

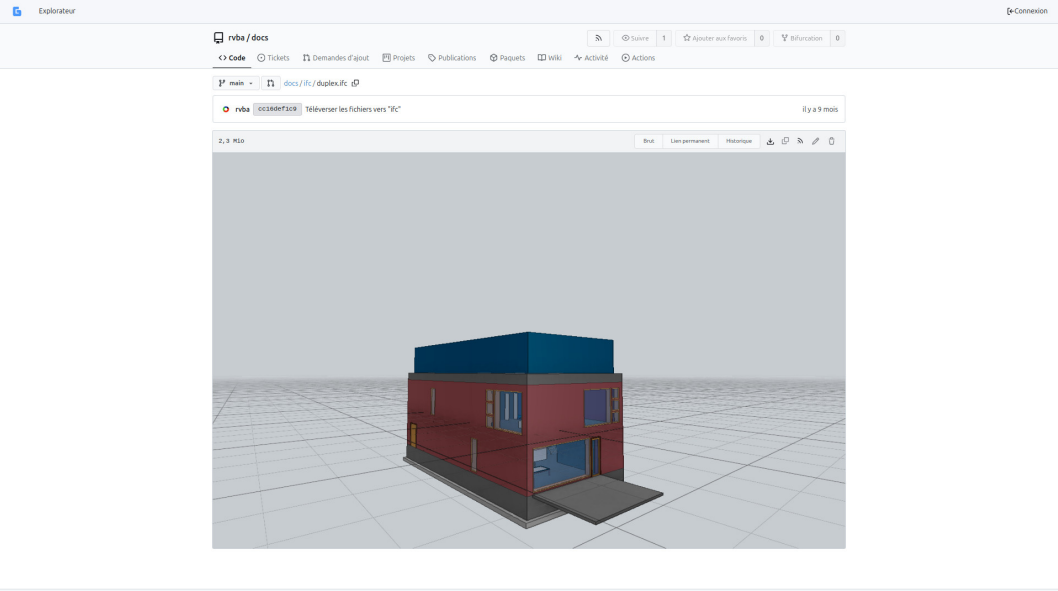



**Figure 5:** GitAec: the browser of IFC files

During the same hackathon, we also presented a simple generative system based on Forgejo's continuous integration system. In computing, a continuous integration (CI/CD) system refers to a task automation system integrated with Forges that responds to events such as the uploading of a change (`commit`) to the platform (`push`). Typically, this type of CI/CD system is used to generate software versions from source code (compilation) or static websites from Markdown sources.

In the field of AEC, we have demonstrated that such a system can be used to extract data from a project (example) or serve as a generator of parametric objects (example). As a Design Forge, GitAec is both an online storage space and a platform for data generation and

[39] See https://gitaec.org/rvba/docs/wiki/Zurich-hackathon
[40] See Industry Foundation Classes
[41] https://github.com/brunopostle/ifcmerge

exchange. These characteristics are therefore similar to those of a CDE (*Common Data Environment*), here "reinterpreted" in the specific context of Git-related technologies. This experimentation around a CDE "derived" from Git also shows promising results in terms of project version management, as demonstrated by the `IFcMerge` project [42]. Thanks to the 3D viewer, it is possible to easily visualize the different "phases" of a project [43].

### Myrmix

The Design Forge, as we tested it with GitAec, aims to bring together two environments that are currently similar but nevertheless quite distinct: CDEs and document management systems on the one hand, and Digital Twins on the other. In a "minimal" sense, a Digital Twin is a 3D model that faithfully represents a constructed building. The relationship between the BIM model, the CDE, and the Digital Twin can be described in a "temporal" manner, in the sense that the three models represent the three stages of a building's life, from design to construction and operation. During the operational phase, certain types of Digital Twins also integrate data captured in real time, as is the case for technical buildings or Territorial Digital Twins, for example.

As a tool for developing complex software systems, Forges, and more specifically Design Forges, seem to us to be an interesting candidate, capable of combining the issues common to BIM models, CDEs platforms, and Digital Twins. This convergence could benefit the ecological imperatives that require us to evolve our current, rather linear construction systems towards more circular models, as in the context of the "circular economy." More generally, and as we will see later regarding AI factories, we propose exploring the ability of Design Forges to address the challenges of "industrializing construction," inspired in particular by production rationalization methods from the automotive and aerospace industries.

[42] https://github.com/brunopostle/ifcmerge
[43] See https://gitaec.org/rvba/docs/wiki/Zurich-hackathon

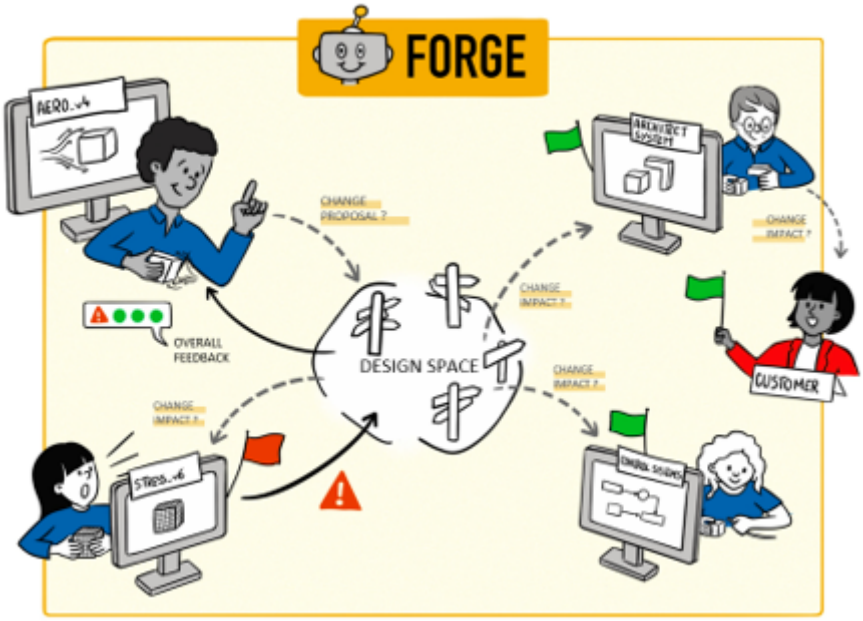



**Figure 6:** Illustration of the principle of the Engineering Forge. (c) Myrmix

We found that these hypotheses and our initial research results in this area converged perfectly with the results obtained by Myrmix, an "Engineering Forge" prototype developed by Inria researchers [44]. During the "Forgeons!" workshop dedicated to Digital Twins, we were able to compare our results and verify that there were very similar points of convergence between our Design Forge prototype, developed using Forgejo, and the Engineering Forge prototype developed by Myrmix using Gitea [45].

A blog post published on the Myrmix website [46] with the evocative title: "Why does systems engineering need its own Github?" [18], perfectly echoes the challenges of the AEC sector: complex systems are multidisciplinary "puzzles". It is from this same observation that BIM was born, with the idea that a plurality of domain-specific models can converge towards a central model during the design phase. In reality, the challenges are obviously numerous, particularly in terms of data collection and synchronization. It is currently the role assigned to CDEs to be exchange platforms. When it comes to converging towards a complex and single model heterogeneous data from multiple stakeholders, the task is far from trivial, and initiatives like Speckle propose to respond to these problems thanks to a "3D server" that acts much more than a simple model. Faced with this observation, we can see how the specificities of a technology like `Git`, and more generally those of Forges, potentially represent a solid response to this complex problem.

[44] See https://www.inriastartupstudio.fr/portfolio/
[45] Recall that Forgejo is a "hard fork" of Gitea
[46] https://www.myrmix.tech

## AI Factory

Before concluding this report, we will address the theme of artificial intelligence (AI) in the context of Domain-Specific Forges, with the notion of "AI Factory". This concept, and the numerous innovations that have emerged since the launch of ChatGPT in November 2022, could usher in a new chapter in the evolution of design tools. Despite persistent doubts about the viability of the new economic model driven by this wave of innovations, and the reality of the very principle of Artificial General Intelligence (AGI), Large Language Models (LLMs) have already proven their worth; their uses in the field of AEC could herald the emergence of a fourth generation of digital design tools.

As detailed in this section, the very principle of the AI Factory is polysemous, even ambiguous, or open to debate. The concept is likely to evolve with the rapid developments characterizing this new technological phase, and the term may be reevaluated in the future. Despite these numerous questions regarding the sustainability or validity of the terms used, we have chosen to retain the term "Factory" because it directly refers to the underlying principle of a "software factory," corresponding to the principles we seek to describe. In computer science, a "software factory" refers, as detailed below, to systems for automating software creation.

We begin by providing some definitions of principles as published on the CodeAtlas website [47].

### Definition

The **AI Factories** are next-generation software production systems based on generative AI mechanisms. As a term referring to an emerging technology, it is not entirely univocal and today takes on two distinct meanings:

- (1) AI Factory of foundation models (LLM)
- (2) Application AI Factories

In a general sense, the term AI Factory (1) generally refers to "supercomputer" type infrastructures that allow the design and manufacture of the latest generation foundational models (LLM) as is the case with the European AI Factories program, and GENCI in France.

[47] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

In a “derived” sense, the term AI Factory (2) rather refers to “superstructures” of software production using foundational models (LLM); this is, for example, what Wikipedia describes at the entry **AI Factory**.

### Uses

Although these two types of “Factories” are correlated, we refer here to “application” AI Factories (2), specifically in their articulation with the disruptive nature of **Vibecoding**. The IT development professions are currently experiencing a wave of radical innovation with the advent of **intelligent agents** capable of producing software in a “semi-autonomous” manner. Even if the subject raises many questions (notably on security) and sparks heated debates, many companies today announce staff reductions in connection with this “revolution.”

This wave of innovation will therefore impact all digital sectors and represents as many challenges and opportunities for the world of architecture. In an environment today dominated by big American players, there is an important issue for the digital future of construction, particularly in the strategic field of **off-site construction**.

### Forges and Factories

It is in this context of semi-automatic software generation (vibecoding) that AI Factories are called upon to act as platforms for the development of next-generation architectural design tools. To fully understand the role and operation of these new Factories, it is important to understand their relationship with the development Forges.

Forges are essential and central development platforms in the design, manufacturing, and delivery processes of software and digital infrastructures. The AI Factory, designed as an application manufacturing hub, is an extension of the Forges, onto which it is grafted, in a way, to extend them. This relationship could be summarized as follows: “Forges for humans, Factories for agents.”

### Prototype

AI factories and open source AI models are therefore set to revolutionize software production methods, as discussed in the presentation “OpenR1 and the Future of AI Factories in April 2025.

Based on these elements, we made a prototype of AI Factory for architecture (Anything-Button) during a hackathon in April 2025, awarded with an award (**Most Impactful Hack**) by a jury composed of professionals such as: Zaha Hadid, Foster+Partner (London), and Thornton Tomasetti (New York).

### Flow-based modeling

At the heart of this wave of innovations triggered by the emergence of generative AI and Large Language Models (LLMs) based on the "Transformer" architecture [48], one element in particular seems to be profoundly redefining the way we interact with our computing tools: "agents." The nature of these "intelligent agents," or more generally, "agentic AI," remains a matter of debate, particularly regarding the distinction between a "simple" automation chain based on an LLM (workflows) and a "clean" agent. In this area, a compromise seems to have been reached regarding the "non-linear" nature of an agent that makes "decisions" on the tasks assigned to it.

In this context, the MCP (*Model Context Protocol*) unifies the central issue of "tool calling" by LLMs, whether performed by agents or workflows. Thanks to this new protocol, any application can be "interfaced" to be usable by an LLM. Ultimately, whether agents or workflows, the innovation brought by this protocol lies in the networking of applications. Popular application interfaces such as n8n, based on "visual programming," are directly reminiscent of the interfaces used in generative design and in many contemporary 3D applications.

In a research paper entitled "Flow-based modeling" [22], we described the principles at the heart of this paradigm, which works, similar to APIs and "microservices" architectures, by connecting output points to input points to form data flows from a set of applications or software libraries accessible from the network. During a GitAec presentation in Zurich in 2025 [15], we demonstrated how the "hexagonal" architecture (of the *ports and adapters* type) was a common paradigm for platforms such as Speckle, BHoM, and Compas.

[48] The "T" in the acronym GPT for *Generative Pretrained Transformer*.

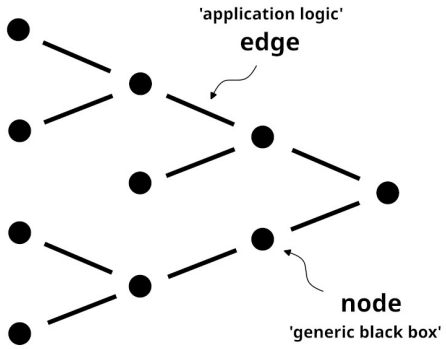



**Figure 7:** Flow-based Modeling Principle

During this event, we also demonstrated how the architecture of a Forge operates according to a pattern that meets the expectations of the stream modeling problem. Forges function as platforms (*hubs*) where different types of streams converge and are processed. More generally, in the context of generative design, we demonstrated how the very principles of continuous integration (CI/CD) offered an interesting solution for managing and steering generative workflows, with upstream data and functions stored and versioned on a Forge. We emphasized the importance of the direction of this flow, which determines the "*generating => generated*" relationship.

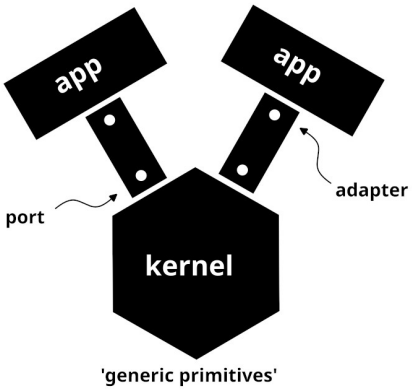



**Figure 8:** Principle of Hexagonal Architecture

### Decentralized AI

With MCP (*Model Context Protocol*) [49] and "agentic AI," the networking of digital tools seems to be part of a trend that, in many ways, recalls the early days of the internet. Long confined to university labs and large digital groups, AI has until now remained a specialist domain, reserved for a privileged few. But since the public launch of ChatGPT in November 2022, everything seems to indicate that we have entered a new phase of rapid technological evolution, and the current excitement in the financial markets recalls somehow the "euphoria" of the 2000s, which ended with the bursting of the "dotcom bubble."

If we look for points of comparison, we observe similarities in two areas in particular: "open source AI" and "decentralized AI." In terms of disruption in this field, January 20, 2025, marks a milestone with the release of the DeepSeek R1 model and this Chinese "Sputnik moment." As we document on Codeberg [50], this date marks the "explosion" of open source AI models and the central role played by China in this field. On this subject, two reports published in quick succession by the White House [17] and Beijing [8] in July 2025 confirm the crucial role that open source will play in this "hegemonic" battle for generative AI.

In this context marked by geopolitical tensions, whether it be an embargo on next-generation chips on the American side or the control of information in the case of LLMs produced under the Chinese flag, the question of the very strong "centralization" of these new technologies arises. In a white paper titled "A Perspective on Decentralizing AI" [16] recently published by MIT, the issue of decentralizing "LLM infrastructures" is raised, perhaps opening a new chapter in the history of large language models, with initiatives like PrimeIntellect [51]. An initiative that could herald a new era of AI in which open-source and decentralized protocols would ensure that the technology remains in the hands of individuals, serving the public interest.

## Conclusion

In conclusion to this report, which provides an initial mapping of the various Domain-Specific Forges and their potential uses in the fields of architecture and construction (AEC), we propose to synthesize these issues as an introduction to a broader context related to the challenges of climate change. First, let's briefly review the role that Forges could play in the current digital design ecosystem.

[49] See https://modelcontextprotocol.io/docs/getting-started/intro
[50] https://codeberg.org/rvba/osai
[51] https://www.primeintellect.ai/

In a context marked by rapid developments related to artificial intelligence and the networking of design tools, we have shown how Domain-Specific Forges could play a stabilizing role by contributing to the dissemination of best practices from the IT world and serve as a benchmark for the creation of next-generation infrastructures. As we have illustrated, Git, as an international standard, could serve as a basis for the development of collaborative applications in the AEC field, just as the USD (Universal Scene Description) standard is currently under consideration to serve as the basis for the future version of IFC (IFC5).

In a similar vein, recent research [3] also refers to Web3 standards and technologies, such as DAO (Decentralized Autonomous Organizations). Here, the links between industry technologies and specific issues in architecture and construction converge towards the principle of "Industry 4.0." In this context of increasing complexity in architectural design and construction, and particularly the circular economy, Forges could serve as a basis for the development of future platforms similar to management tools such as PML (Product Lifecycle Management) from highly standardized industries.

This necessary complexity of construction therefore calls for new types of tools, as the construction sector represents a significant part of the climate change issue and must therefore evolve to make our industrial societies more sustainable. Off-site construction, and more generally the "industrialization" of construction (prefabrication), are currently important themes capable of providing concrete solutions to the challenges of climate change. However, for the construction world, this represents numerous challenges in an industrial context marked by a significant fragmentation of the sectors and stakeholders involved in the process of building, from design to construction.

If a paradigm shift is necessary in the field of construction, we must therefore question new paradigms in the field of design. As we have shown with the Design Forges, Architecture as Code (AaC) could play a role in this context. Today, our vision of the architectural project has evolved, as we no longer consider buildings in a "static" manner, but as "open" works, with a more "dynamic" or even "cyclical" design point of view. With a more circular vision of the art of building, it is perhaps time to consider the architectural project as an open work and as a open process [52]. Architecture as Code could be the support allowing this new vision, and with it, the Forges as a place of exchange, co-construction, even sharing of architectural know-how.

[52] See https://www.tudelft.nl/en/architecture-and-the-built-environment/research/research-at-bk-bouwkunde/the-new-open

## History

### 2009

- First version of a “Serious Construction Game” developed for the Blender workshop at the “Make Art” festival on the theme of Open Source Architecture

### 2017

- First version of the Alphaville script for a bachelor's course at the ENSA La Villette.
- First experimentation with GitHub for a master's course (ENSAXP) at the ENSA Val de Seine

### 2018

- Publication of “Flow-based Modeling” [Yanatchkov [22]]

### 2022

- Publication of “Design Platforms” [Yanatchkov [23]]
- First experimentation of a Educational Forge in a master's program at ENSA Nancy: Parametric Palace

### 2023

- Development of the Learning Forge prototype Sloyd

### 2024

- Development of the Design Forge prototype GitAEC
- Presentation of Educational Forges during the OW2/Apereo conference [*Open Source in Education, Science and Research - OW2con* [14]]

### 2025

- Creation of the working group “Forgeons !” (*Let's Forge!*) with the support of DINUM

## Glossary

- **Forge**: A web application for collaborative software development based on Git.
- **Development Forge:** The "original" Forge serving as the basis for creating "Domain-Specific Forges."
- **Domain-Specific Forge:** A derivation of a Development Forge in a specific domain.
- **Educational Forge:** A Domain-Specific Forge for the field of education, primarily for primary and secondary education, for the production and sharing of Open Educational Resources (OER).
- **Learning Forge:** A specific type of Educational Forge, intended particularly for pupils and students, serving as an interactive learning environment.
- **Design Forge:** A Domain-Specific Forge in the field of architecture, engineering, and construction (AEC).
- **Design Platform:** A new generation design application accessible online from a browser, differing from "native" desktop applications.
- **AI Factory:** A next-generation software production system based on generative AI mechanisms. The term has two distinct meanings:

    1. **Foundation Model AI Factory (LLM):** Supercomputer-type infrastructures for designing and manufacturing foundational language models.
    2. **Application AI Factory:** Software production superstructures using foundational models (LLM) for specific applications, particularly in semi-autonomous software development.

## References


[1] Safiye Aslan. "Is Learning by Teaching Effective in Gaining 21st Century Skills? The Views of Pre-Service Science Teachers". In: *Educational Sciences: Theory and Practice* 15.6 (Dec. 2015), pp. 1441–1457. ISSN: 1303-0485. URL: https://eric.ed.gov/?id=EJ1101263 (visited on 08/30/2025).

[2] Daniel Le Berre et al. "Higher Education and Research Forges in France - Definition, Uses, Limitations Encountered and Needs Analysis". report. Comité pour la science ouverte, Sept. 29, 2024. DOI: 10.52949/37. URL: https://hal-lara.archives-ouvertes.fr/hal-04208924 (visited on 09/15/2025).

[3] David F. Bucher et al. “From BIM to Web3: A Critical Interpretive Synthesis of Present and Emerging Data Management Approaches in Construction Informatics”. In: *Advanced Engineering Informatics* 62 (Oct. 1, 2024), p. 102884. ISSN: 1474-0346. DOI: 10.1016/j.aei.2024.102884. URL: https://www.sciencedirect.com/science/article/pii/S1474034624005329 (visited on 09/10/2025).

[4] COREALE. *Numérique Pour l’enseignement Supérieur et La Recherche - Mesure-IX*. Ministère chargé de l’enseignement supérieur et de la recherche, Feb. 2025. URL: https://media.licdn.com/dms/document/media/v2/D4E1FAQFL5Nr60F7gkQ/feedshare-document-pdf-analyzed/B4EZZs.iMHHYAc-/0/1745585066824?e=1756944000&v=beta&t=Dy6UO7mqbcAwy8-HE-fexHyGao4Z8qWzoBPO4IbQFFM.

[5] *Feuilles de routes : Stratégie du numérique pour l’éducation 2023-2027*. Ministère de l’Education Nationale, de l’Enseignement supérieur et de la Recherche. URL: https://www.education.gouv.fr/feuilles-de-route-450426 (visited on 08/26/2025).

[6] Thierry Forest, Bertrand Marne, and Iza Marfisi-Schottman. *Compte Rendu d’atelier : Recueil de Pratiques d’enseignantes En Formation*. Le mans Université, July 2025. URL: https://hal.science/hal-05158143 (visited on 08/28/2025).

[7] Thierry Forest, Bertrand Marne, and Iza Marfisi-Schottman. “Revue systématique de littérature sur les fonctionnalités des forges logicielles exploitables dans un milieu éducatif”. report. Le mans Université, July 1, 2025. URL: https://hal.science/hal-05158111 (visited on 08/28/2025).

[8] *Global AI Governance Action Plan_Ministry of Foreign Affairs of the People’s Republic of China*. URL: https://www.mfa.gov.cn/eng/xw/zyxw/202507/t20250729_11679232.html (visited on 09/11/2025).

[9] Andreia Inamorato dos Santos, Yves Punie, and Jonatan Castaño Muñoz. *EUR 27938 - Opening up Education: Support Framework for Higher Education Institutions*. Publications Office of the European Union, 2016. ISBN: 978-92-79-58537-1. URL: https://data.europa.eu/doi/10.2791/293408 (visited on 08/27/2025).

[10] *La Recommendation 2019 de l’UNESCO Sur Les Ressources Éducatives Libres (REL)*. URL: https://unesdoc.unesco.org/ark:/48223/pf0000383205_fre.

[11] Daniel Le Berre et al. *Forges de l’Enseignement Supérieur et de La Recherche - Définition, Usages, Limitations Rencontrées et Analyse Des Besoins*. Ministère de l’enseignement supérieur et de la recherche, 2023. DOI: 10 . 52949 / 34. URL: https://hal-lara.archives-ouvertes.fr/hal-04098702 (visited on 09/09/2025).

[12] *Le comité numérique pour la réussite étudiante et l'agilité des établissements (COREALE)*. enseignementsup-recherche.gouv.fr. URL: https://www.enseignementsup-recherche.gouv.fr/fr/le-comite-numerique-pour-la-reussite-etudiante-et-l-agilite-des-etablissements-coreale-98527 (visited on 08/27/2025).

[13] Jean-Pol Martin. "« Lernen durch Lehren » : quand les apprenants font la classe". In: *Les cahiers de l'APLIUT. Pédagogie et Recherche* (Vol. XXIII N° 1 Feb. 15, 2004), pp. 45–56. ISSN: 2257-5405. DOI: 10.4000/apliut.3439. URL: https://journals.openedition.org/cahiersapliut/3439 (visited on 08/30/2025).

[14] *Open Source in Education, Science and Research - OW2con*. URL: https://www.ow2con.org/view/2024/Breakout_Sessions/Open_Source_Education_Science_Research (visited on 09/10/2025).

[15] opensource.construction, director. *GitAec – Making IFC Versioning Accessible for Everyone, Milovann Yanatchkov, ENSA Nancy*. Mar. 3, 2025. URL: https://www.youtube.com/watch?v=vqkKc3TCWdg (visited on 09/11/2025).

[16] Ramesh Raskar. *A Perspective on Decentralizing AI*. MIT Media Lab. URL: https://www.media.mit.edu/publications/decai-perspective/ (visited on 09/10/2025).

[17] *White House Unveils America's AI Action Plan*. The White House. July 23, 2025. URL: https://www.whitehouse.gov/articles/2025/07/white-house-unveils-americas-ai-action-plan/ (visited on 09/11/2025).

[18] *Why Systems Engineering Needs Its Own GitHub*. myrmix. May 19, 2025. URL: https://www.myrmix.tech/fr/blog/articles-2/why-systems-engineering-needs-its-own-github-1 (visited on 09/10/2025).

[19] Yanatchkov. *Villes et Architectures : Vers Des Œuvres Ouvertes ?* Dec. 8, 2009. URL: https://archive.bleu255.com/makeart/2009/?page=milovann_yanatchkov (visited on 09/11/2025).

[20] Milovann Yanatchkov. *"Fork Une Maison !" Architecture Open Source. Atelier Make Art 2009*. Dec. 8, 2009. URL: https://archive.bleu255.com/makeart/2009/?page=workshop (visited on 09/11/2025).

[21] Milovann Yanatchkov. *A New Kind of Forge for AEC*. LinkedIn. Feb. 9, 24. URL: https://www.linkedin.com/pulse/new-kind-forge-aec-milovann-yanatchkov-fmyce/ (visited on 09/10/2025).

[22] Milovann Yanatchkov. "Modélisation par le flux". In: *SHS Web of Conferences* 47 (2018), p. 01006. ISSN: 2261-2424. DOI: 10.1051/shsconf/20184701006. URL: https://www.shs-conferences.org/articles/shsconf/abs/2018/08/shsconf_scan18_01006/shsconf_scan18_01006.html (visited on 09/10/2025).

[23] Milovann Yanatchkov. “Plateformes de conception”. In: *SHS Web of Conferences* 147 (2022), p. 01001. ISSN: 2261-2424. DOI: 10.1051/shsconf/202214701001. URL: https://www.shs-conferences.org/articles/shsconf/abs/2022/17/shsconf_scan22_01001/shsconf_scan22_01001.html (visited on 09/10/2025).