D5.3 Generative Design Platform as Social Community

Siemens AG (SAG)

June 2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 870037.

DELIVERABLE INFORMATION	
Author(s)/ Organization(s)	A. Votintseva / SAG, M. Kramer / SAG, R. Johnson / SAG
Document type	Report
Document code	D5.3
Document name	D5.3 Generative Design Platform as Social Community I
Status	EU
Work Package / Task	WP5, T5.2
Delivery Date (DoA)	June 2021
Actual Delivery Date	29 June 2021
Abstract	This deliverable reports about activities performed in the task T5.2, Generative Design Platform, during the period of M7-M18. The document gives an overview of the technologies of Generative Design, specifies the overall architecture of the platform under development, and describes the current state of the implementation of all components constituting the Generative Design Platform.

DELIVERABLE HISTORY			
Date	**Version**	**Author/ Contributor/ Reviewer**	**Summary of main changes**
01/03/2021	V0.1	A.Votintseva / SAG	Initial version of TOC
18/03/2021	V0.6	M. Kramer / SAG	Architecture Section 3, maintained list of abbreviations
21/05/2021	V0.7	A.Votintseva / SAG, R. Johnson / SAG	Major Sections update (besides Architecture)
28/05/2021	V0.9	M.Kramer / SAG	Finalization of Section 3 (GDP Architecture) & subsection 4.3
31/05/2021	V1.0	A.Votintseva / SAG	Finalization of the document, maintained references
8/06/2021	V1.1	D.Paul / FIT	Peer Review
28/06/2021	V1.2	M. Kramer / SAG, A.Votintseva / SAG	Update of all sections according to the review

DISSEMINATION LEVEL		
PU	**Public**	**X**
PP	Restricted to other program participants (including the EC services)	
RE	Restricted to a group specified by the consortium (including the EC services)	
CO	Confidential, only for the members of the consortium (including the EC)	

DISCLAIMER

iPRODUCE ● Grant Agreement: 870037 ● Innovation Action ● 2020 – 2022 | Duration: 36 months

Topic: DT-FOF-05-2019: Open Innovation for collaborative production engineering (IA)

List of Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
AR / VR	Augmented Reality / Virtual Reality
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
CNC	Computer Numerical Control
CAD	Computer-Aided Design
cMDF	collaborative Manufacturing Demonstration Facilities
DIY	Do It Yourself, maker activities
GD / GDP	Generative Design / Generative Design Platform
GUI	Graphical User Interface
HW	Hardware
KPI	Key Performance Indicator
ML	Machine Learning
NFR	Non-Functional Requirements
NLP	Natural Language Processing
OpIS	Open Innovation Space
OTS	Off-The-Shelf
PCB	Printed Circuit Board
SAG	Siemens AG
SME	Small and medium-sized enterprises
SDK	Software Development Kit
SMF	Social Manufacturing Framework
SW	Software
UC	Use Case
UCD	User-Centered Design
UI	User Interface
UX	User Experience
WP	Work Package

Executive Summary

This report is a deliverable of the iPRODUCE project, funded by the European Commission's Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation program (H2020). The report describes the results of the activities carried out between M7 and M18 with the context of WP5 (Customer-Driven Production and Co-Creation Enabling Tools), particularly in the Task 5.2 "Generative Design Platform" (GDP).

The objectives of the task 5.2 are:

- To implement a SW application that exploits generative design methods and algorithms
- To provide an intuitive and user-friendly access to co-creation and co-design methodology for a number of user groups with different target uses:
    - Engineers and designers in FabLabs/MakerSpaces to generate (parametric) 3D geometries covering different design constraints (and possibly defining freedom spaces for consumers)
    - Consumers to personalize their products with respect to the constraints identified for actual product line
    - Administrators of production line to define requirements, constraints and priorities
- To run the implemented GDP application in a cloud infrastructure to offer a web-based frontend
- To address social community aspects within the GDP enabling the mentioned users to communicate their decisions, share results and influence one another's proposals
- To create a positive User Experience (UX) to promote the usage of generative design by end users

This Deliverable D5.2 "Generative Design Platform as Social Community" describes the first prototype of the GDP application, including:

- Initial settings needed to start implementation: technology evaluations, use case research, mock-up trials
- Overall GDP architecture
- Detailed description of the functionalities and current development state of the three main components of the GDP: Mathematical Modeling, 3D Product Configurator, and 2D Layout Planner
- Plans for the development in the next period

Because the Generative Design Platform is currently only in a prototype state, there are a number of limitations in all its constituting components. The major limitation is that all components are not yet integrated with each other and will be first provided for the early evaluation as individual tools. Their functionality is also limited, in particular the corresponding User Interfaces (UI) are not yet fully implemented and tested. Besides, only few iPRODUCE use cases are addressed by the current prototype, which also will be extended in the future.

The GDP application will be further developed in the next period of the iPRODUCE project, when the current limitations will be addressed, as well as the aspects of the GDP integration into the OpIS platform of iPRODUCE.

Table of contents

List of Figures

List of Tables

1. Introduction

The results of the first version of the prototypical implementation of individual components for Generative Design SW is reported in this deliverable, D5.3: “Generative Design Platform as Social Community”.

The complete development of the tool will be accomplished by month 36. The completed platform will promote the Generative Design techniques through the implementation of parametric modelling and application of genetic algorithms for both “makers” (engineers, designers, manufacturers) and consumers and advance suitable User Interfaces to do so. Generative Design will be used to engineer the 3D models in the FabLabs using the typical additive manufacturing constraints such as material consumption and stability, but also offered to the consumers as a means to personalize their product, by giving the system the parameters of the manufacturing product line as constraints, but letting the user influence the unconstrained solution space.

The completed Generative Design Platform (GDP) will contribute to the open innovation process being a part of co-design methodologies and tools that help to bring together a wide range of relevant stakeholders. The specific UI of the GDP will shape the collective co-design of products for diverse areas of consumer products. It will be evaluated in different use cases of iPRODUCE.

1.1. Purpose and scope of this deliverable

The purpose of this deliverable is to provide a description of the current state of the tool development of the GDP starting from month 7 (July 2020) until month 18 (June 2021) in the project.

In the scope of this deliverable are:

- Activities performed during the first period of the tool development
- GDP Architecture & decisions on the technologies to be used for the GDP & feature prioritization
- Decision about the use cases to be addressed by the GDP and to be used for the GDP validation
- Description of the web-based frontend for co-creation with the focus on the UX to be researched and improved further
- Focus on promoting generative design methods for different stakeholder groups very early, already in the prototypical state
- Groups of users:
    - Amateur engineers & designers in FabLabs / MakerSpaces
    - Consumers wishing to personalize their product
    - Administrators of product lines who intend to specify production constraints in design and who want to collect and view consumer wishes

Out of scope are:

- Currently at this early development phase, integration of the individual components is not addressed. It will be covered in the next deliverable.
- Social community aspects will be developed further and described in more detail in the successive report.

- Authorization & authentication features are not considered in the GDP because it will be covered by another tool from an iPRODUCE partner (probably by the Marketplace from WP4, T4.3, which is still under discussion).
- Currently only few use cases are addressed in this GDP prototype. Further use cases will be covered after M18.

1.2. Generative Design Overview

Generative Design is a form of engineering/designing solutions to a given problem with an emphasis on simply outlining the problem ("need objects for human to sit upon") together with a set of rules ("material: wood", "carrying capacity: > 100kg") or constraints ("do not use more than 20kg of wood") as well as digitized success criteria ("result should enable human to rest back"), having the system to generate all possible solutions to fit the described problem. State-of-the-art engineering software (Siemens NX [6], Solid Edge [7], Autodesk Fusion 360 [8], Rhinoceros 3D [4]) offer Generative Design add-ins for engineers to explore the solution space to 3D modeling with constraints such as to respect certain dimensions, material waste or weight. Currently, the User Interfaces (UI) for viewing possible solutions as well as the means for the user to interfere during the generation and to alter the afore entered parameters are weak. Current engineering education also includes Generative Design only as an optional side topic.

> "*Generative design mimics nature's evolutionary approach to design. Designers or engineers input design goals into generative design software, along with parameters such as materials, manufacturing methods, and cost constraints*." – Cadabra Studio [9]

Generative Design is often understood as an algorithmically-supported designing process that can be applied nowadays in different domains, including 3D modeling, 2D layout, animations, audio/video content creation and many other domains. Algorithmic design is not only used to create objects and contents. Special kinds of algorithms are implemented to overcome different limitations of traditional CAD & 3D modeling systems.

Parametric modeling plays important role for the application of generative design to 3D or 2D construction & optimizations. Parametric models define how an object can be created, and their parameters define what the spaces of freedom are, i.e. how the generated object can be modified. The definition of the parameters may contain further constraints on the possible values. Alternatively, constraints can be defined during the definition of the generative study, as it is typically performed within CAD systems like Solid Edge or Fusion 360. Other than parametric models, direct modeling does not define the freedom space so explicitly. One can imagine a direct model like a single model for a product (which still can be modified directly in the CAD application) and a parametric model contains a wide range of different models that are generated by one algorithm. This allows easy application of genetic and other evolutionary algorithms upon parametric models for optimization to reach specific design objectives. Even multiple optimization objectives can be specified for genetic algorithms, although this impacts performance quite heavily.

On the one hand, direct modeling is essentially easier and more intuitive for designers within 3D tools, but on the other hand parametric modeling is open to so much more different optimization techniques and involving customer requests.

Currently, the algorithmic design is available in several CAD/3D modeling tools: Customizer on Thingiverse.com [10], Algorithmic Modeling in Siemens NX, generative studies in Solid Edge or Fusion 360, Grasshopper [5] with evolutionary and other kinds of solvers in Rhinoceros 3D. All algorithmic designs are specific for these tools, and they require solid CAD/3D modeling skills to create parametric models or perform generative studies. While the public consumers primarily define their product wishes verbally with human language, this can sometimes be ambiguous for the manufacturer if product wishes are too specific. The collaboration with the consumer is limited and may induce many iterations of communication.

Nowadays, generative design grows in popularity, in particular when a skillful developer prepares a routine that can be used by many stakeholders with no or little skills in CAD or 3D modeling. So far, such design solutions are focused on particular user groups: Thingiverse for makers, CAD tools (with or without algorithmic modeling and generative studies) for engineers, 3D modeling tools for designers.

The GDP would like to provide a platform where participants with different skills and abilities can contribute to the shared product design combining different technologies to support consumers and connect makers to them: speech recognition in a special chat-bot to retrieve exact parameters of wished designs and perform preliminary generation of 3D/2D layouts, parametric design for 3D & 2D objects to be used with genetic algorithm for the search of the most optimal product configuration within given constraints, web-based frontend that provides remote access to co-design with the User Interface appropriate for consumers.

> "*Genetic Algorithms are developed primarily for problem-solving and optimization in situations where it is possible to state clearly both the problems and the criteria to be fulfilled for their successful solution*." (Prof. John Frazer – Cambridge) [11]

Generative Design is achieved using Genetic Algorithms.

In the time you can create one idea….a computer can generate thousands, along with the data to prove which designs perform best. (Autodesk)

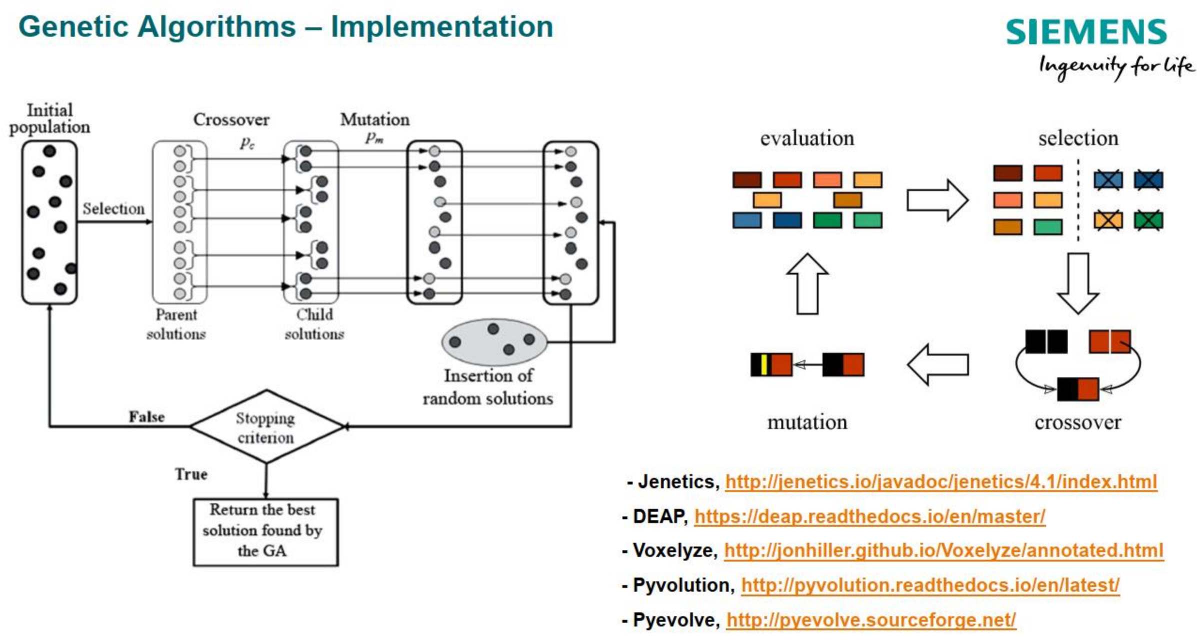

Figure 1: Genetic Algorithms Overview

In the process of Genetic Algorithm, there are several basic elements – population of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring. Selection, Crossover and Mutation are the three essential procedures. (Wikipedia 2018) They are core of the iterative process in the implementation of any genetic algorithm:

- **Step One**: Generate the initial population of individuals randomly. (First generation)
- **Step Two**: Evaluate the fitness of each individual in that population (time limit, sufficient fitness achieved, etc.)
- **Step Three**: Repeat the following regenerational steps until termination:
  - *Select* the best-fit individuals for reproduction. (Parents)
  - Breed new individuals through *crossover* and *mutation* operations to give birth to offspring.
  - Evaluate the individual fitness of new individuals.
  - Replace least-fit population with new individuals.

Figure 2: Simple Example for Generative Design

The Generative Design Platform of SAG will be used to explore a solution space that conforms to a set of constraints using genetic algorithms to explore all possible solutions. The Generative Design Platform is intended to be used in many applications. iPRODUCE introduces a new space exploration process to enable the engineer/designer and user to innovate creatively and engagingly "breed" the final, personalized solution together. This procedure is "borrowed" from social networks (like, dislike), biology (dominant, recessive gene selection and breeding) and gamification (competitions and real time simulations).

1.3. Addressing Project Objectives and KPIs

The GDP aims to address two objectives of the iPRODUCE project:

Objective 3: To develop and deploy a digital platform that will facilitate the activities of the local cMDFs and allow them to operate independently (at local level) under a loosely federated organizational structure (at European level).

The GDP is planned to be integrated into the overall iPRODUCE platform OpIS. The GDP itself integrates several technologies within one web-based tool: parametric 3D & 2D design, application of genetic algorithms to layout and product generation under specified constraints, speech recognition for natural and verbal human communication for layout & design construction; all this with the focus on a

User Interface appropriate for user groups with different backgrounds, in particular for the general public without CAD skills. The GDP will be developed modularly, such that experienced makers can contribute continuously by providing more parametric models and transformations to the platform. Besides, the platform will provide a UI such that consumers can participate in co-design and optimization of their product.

Objective 4: To develop and deploy a set of digital tools that will stimulate co-creation and open innovation in the consumer goods sectors (including the development of e-training sessions, recommendations for green production engineering, recycling/repairing and circular economy approaches).

The GDP will contribute to the iPRODUCE platform by providing means for co-design in one location which will lead to a shorter design and validation cycle. This will be achieved via a special UI that will allow consumers to specify their product wishes in an effective and visual way with no need for CAD & 3D modeling skills from them. On the other hand, skilled makers can propose product ideas with a narrow or wide range of freedom in form of parametric design

The GDP will also contain a set of parametric models and transformations to be used for the creation of new or adaption of existing ideas for consumer wishes. All ideas captured in the GDP can be taken both by experienced and new makers as teaching examples and further extended. The GDP will be designed itself being extendable by new parametric models and transformations.

The GDP will contribute to the following KPIs of the iPRODUCE project:

KPI-1. *Social manufacturing framework = 1*. The GDP will be part of the overall iPRODUCE framework, containing a web-based tool integrating several technologies, a set of parametric models & transformations that can be also used for the training of makers, and guidelines and tutorials for technologies addressed by the GDP. The final implementation of the GDP can be used for training and promotion of generative design technologies to the consumers.

KPI-5. *Number of engaged makers and consumers in the collaborative manufacturing processes of the MMC communities > 1200*. The GDP will provide a UI, which is appropriate and enjoyable for consumers to participate in co-design and thus motivate them to join the design process and potentially connect more participants to FabLabs and MakerSpaces.

KPI-9. *Digital iPRODUCE platform (Open Innovation Space – OpIS) = 1*. The GDP will be part of the overall iPRODUCE platform. It is planned to be integrated with other tools.

KPI-10. *Consumer good sectors addressed = 5*. The GDP is planned to cover continuously more use cases from iPRODUCE and outside of iPRODUCE (extending coverage after the reporting period M18) to evaluate the application of generative design technologies in different domains.

KPI-12. *Customer-driven products manufactured in cMDFs > 12*. The GDP will develop features useful for consumer participants and improve their quality and UX, thus motivating consumer to contribute actively to the product design and effectively request production of their designed product.

KPI-15. *Improvement in makers' and consumers' perceived readiness to participate in collaborative manufacturing > 20% (measured by the evaluation framework)*. The GDP will provide a UI appropriate for consumers to participate in co-design, thus promoting the generative design technologies in understandable way.

2. Activities Performed in Task 5.2

Task 5.2 Generative Design Platform (GDP) has started in M7, July 2020. In this task we planned to develop a platform containing three main components – 3D Product Configurator, 2D Layout Planner, and Mathematical Modeling – to show different approaches of generative design techniques on different domains. 3D Product Configurator and Mathematical Modeling are being developed completely from scratch, and 2D Layout Planner adapts some generative design technologies already evaluated at Siemens for web-based usage in iPRODUCE use cases. The new components and the prototyped one will be integrated into one platform to show different aspects of generative design and different kinds of UI. The implementation of new components started with the initial evaluation of the existing technologies that can support their development to suit to the overall collaborative design from iPRODUCE.

iPRODUCE aims to address specific needs of FabLab/MakerSpace SMEs and users and to explore collaborative methods. Collaborative design methods are getting more acceptance in service and product designs in the industrial sector to address the increasing demand of end user involvement. Solutions are often designed collaboratively with end-users and experts from various domains by applying UX and User-Centered Design (UCD) dedicated methods. Another aim of iPRODUCE is to evaluate and investigate on collaborative creation and creative DIY methods feasible for product development and enhancement involving FabLab personnel as well as employees from the industrial sector and their end-users. The entry threshold for SMEs in prototyping and end user-involving development will be lowered through iPRODUCE's collaborative methods, which will imply a boost in the innovation practice and economic advantage.

The aim of the GDP is to contribute to the collaborative design methods of iPRODUCE. This was the focus of the existing technology evaluation.

During the analysis phase, we also analyzed the use cases that are being developed in iPRODUCE. We have selected several of them, more suitable for the evaluation and demonstration on our GDP components. As a result of this analysis, mockups were developed and demonstrated at the iPRODUCE plenary meetings, at the end of 2020.

Next, the architecture for the GDP was developed covering the main functionality shown in Section 3. All features have been prioritized and the most crucial ones have been selected for the MVP development in M18. The architecture was developed to be aligned with the overall iPRODUCE architecture for the OpIS.

During the elaboration on the GDP architecture, all three components constituting the GDP platform are implemented with the focus on the main features selected for MVP. The development state until M18 (June 2021) is described in Section 4 with current functionality and limitations.

2.1. Technologies Evaluation

At the first stage of the platform development, we have evaluated tools and technologies, that can be used during the development of the GDP to address iPRODUCE's collaborative design objectives. The tools shall address the following contemporary paradigms that we consider as crucial for the support of the collaborative design process (for improved remote collaboration, accessibility, easy operation, effective visualization, fast learning of requested skills, quick prototyping, interaction of multiple stakeholders):

Paradigm/Method	Main feature
Visual Scripting	Product authoring for parametric 3D & 2D models or for requirements visualization with fast learning experience. Due to its graphical interface, better understandable for humans.
Parametric modelling	Allow to "program" designs with multiple parameters that can be adjusted dynamically, so containing many design options for 3D & 2D models of products in only one "small script".
3D model visualization	User-friendly rendering of 3D & 2D models of products for easy end user interactions (including remote interaction).
Cloud applications	Easy access for different stakeholders to allow their remote collaboration over shared product models.
Gamification	Currently the most effective way to learn new skills and to collaborate.
Accessibility	Different kinds of user interaction covering possibilities for disabled people to participate in collaboration, e.g., via haptic, physical, sensor, audio, user interfaces.
Rapid prototyping	Fast product evaluation at earlier design stages.
User Experience (UX)	It is a general notion and covers all the paradigms above. Among others UX includes how a user interacts with and experiences a particular product, system or service. In our prototype we will focus only on different techniques targeting for better user acceptance, simplified usage, and learnability. This becomes particularly important when different kinds of users collaborate using one tool or one system.

Table 1: Paradigms crucial for collaborative design

The following tools were evaluated for their coverage of the methods and paradigms important for the collaborative design mentioned above:

Tool	Pros	Cons
Rhino [4] with Grasshopper [5] plugin	Support for Visual Scripting, Parametric 3D & 2D design, accepts a large number of input & output formats for 3D & 2D. Parametric models (scripts) can be visualized externally with 3rd party cloud service. Large numbers of free plugins for more design, analysis & optimization	License for Rhino is necessary for the development of visual scripts for transformations
Solid Edge [7] with generative Design	Mature CAD system, ready support for generative studies inside the tool	Need for CAD skills, generative design tool is limited
Fusion 360 [8] with generative design	Mature CAD system, ready support for generative studies inside the tool, usage within iPRODUCE project	No API available for generative design component = Tool License needed. Need for CAD skills

AWS (Amazon Web Services [13])	Possibility to deploy our application on cloud for remote control and access for all iPRODUCE partners	Cost issues. A tool for SW developers, not for the end users
ShapeDiver [14]	Cloud service running 3D visualization of parametric scripts from Rhino/Grasshopper in web browsers (thus accessible on mobile devices), mature tool with API, continuously improved	Cost issues, limitations for design parametrization, limits for calculation time.
Compute 3D [15]	Cloud service, support for all features of Rhino 7 and Grasshopper, full control over the computations	Not yet mature enough, poor usability for both end users and developers, cost issues
3D printing: Ultimaker [16]	Rapid prototyping for design evaluation. Used for online training preparation	
Arduino [17]	Prototyping for electronic functionality evaluation, physical inputs into design parameters (from sensors) and output (to motors). Open source HW	May not be applicable for all kinds of end users. But can be used in MakerSpaces and FabLabs
Figma [18]	Rapid prototyping of all kinds of user interfaces; high fidelity clickable prototypes, remotely accessible, rich design library	Only design skills needed
MQTT [19] (diverse applications)	Remote user interaction with design prototypes for early evaluation of design functionality and possible user interactions	Integration into the design tool questionable
GPT-3 [20] from OpenAI	Language Generator for speech processing and generation of machine-readable formats for user requirements	License costs
Google BERT [27]	A transformer-based ML technique for natural language processing (NLP) pre-training developed by Google. Free Apache license	
ThreeJS Editor [21]	Editor for effective UI creation, freely available	
3rd party genetic algorithms (Ex.: Genetics.js [22], GeneticSharp [28])	Genetics.js – an evolutionary algorithm used for web-based development, open source. GeneticSharp – fast, extensible, multi-platform and multithreading C# Genetic Algorithm library	Genetics.js: problems with performance for online applications

Table 2: Technologies evaluated for Generative Design

For the integration of 2D Layout Planner into the GDP a set of UI Editors has been evaluated:

Name	OS	X-platfor	Web	comment	suitabl	rating
Sweet Home 3D	x	x	-	installable, not browser based	-	
Seemless3d	x	x	-	installable, not browser based	-	
HomeStyler	-				-	
FloorPlanCreator	-				-	
GoJS FloorPlanner		x	x	good, but commercial, expensive, no OSS	-	
DrawerJS	x	x	x	more drawing tools, possible, to implement based upon although a couple of controls are bad in UX,	x	3
Draw2d.org	x	x	x	'e basic than GoJS, but good, ähnlich wie DrawerJS... b	x	3
node red	x	x	x	not really a candidate	-	
KineticJS				no longer maintained	-	
ConcreteJS	x	x	x	bad documentation, lacking examples, unclear how & if job can be done using this framework	-	4
Konvajs	x	x	x	Looks promising, appears to be based on KinticJS	x	2
fabricjs	x	x	x		x	2
blueprint-js	x	x	x	looks promising, even integrates VR walk through, according to screenshots inventory ins integrated? https://github.com/aalavandhaann/blueprint-js Nevertheless, did not find out how to do the internal room/floor planning, only building planning	x	2
JointJS	x	x	x	Paper attributes demo has a set of the required features, but I am lacking polylines and rotate	x	2
react planner	x	x	x	Best candidate amongst tested: https://github.com/cvdlab/react-planner	x	1
ShadowEditor	x	x	x	more 3D than 2D	-	
PixiJS	x	x	x	levelopment, more collection of tools, than usable ec	-	
FloorSpaceJS	x	x	x	Good candidate from first sight. Nevertheless I was unable to find functionality for Interiour planning --	-	
ThreeJS Editor				Good editor candidate for mathematical modelling, less capable for the 2D use case requirements.	-	

Figure 3 Evaluated UI editors (final candidate selected is highlighted as green)

Some methodologies of the technology evaluation are described in the iPRODUCE Deliverable D5.1 [1], Section 4.

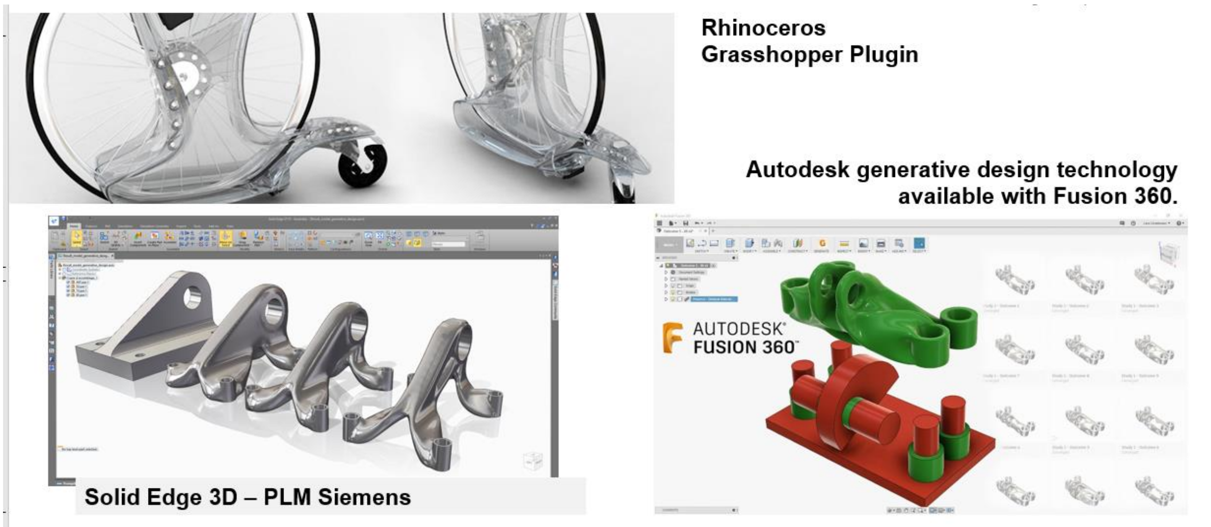

Figure 4: CAD tools evaluated for Generative Design

After different evaluations we have chosen the following tools to be used for the development of Generative Design Platform:

- Rhino 7 with Grasshopper and further plugins for 3D & 2D authoring of parametric models:
    - Small file size for design
    - Easy parametrization definition
    - User-friendly interface (visual scripting)
    - Easy to learn and use
    - Rich library of plugins for different kinds of simulation, optimization and fast design operations, active community
    - Scripts for parametric models (including transformation models) can be used externally for visualization and configuration by end users without Rhino or other CAD programs.
- ShapeDiver for online visualization of parametric models from Grasshopper scripts integrated into the GDP via API:
    - Cloud-based service, available online
    - User-friendly interface, effective for the development of the GDP
    - Less performance issues than in other similar services
    - Available and well documented API with a rich set of features.
- AWS for cloud deployment of the GDP:
    - Provisioning of an easy access to the platform for different stakeholders
    - Diverse additional services that can be added on demand.
- Figma for rapid prototyping and visualization of future functionalities:
    - Online accessible by different stakeholders
    - Easy development of a high-fidelity prototypes for GUI
    - Suitable for early documentation.
- Arduino (different versions) with diverse sensors and actuators:
    - Open-source HW available for makers: rich SW libraries, active community
    - Fast physical prototyping and early evaluation of electronic functionality
    - To be used for physical input interfaces within accessibility topics to attract more participants into collaborative design, including wearable technologies.
- Raspberry Pi will be further evaluated and used for specific iPRODUCE use cases (not covered yet).
- GeneticSharp to be used for optimization within generative design in online 3D Product Configurator & 2D Layout Planner.
    - Fast and extensible
    - Support for multi-platform and multithreading
    - Simplifies the development of applications with Genetic Algorithms
- Google BERT will be used for natural language understanding in the chatbot UI of the GDP (a.k.a. Companion):
    - Free availability
    - Ease user interaction with the platform by automatic extraction of customer wishes
- React planner, an open-source UI editor, will be used for the development of UI of the 2D Layout Planner:
    - Covers the required features of the tool
    - Best integrated with the backend of the GDP.

2.2. Analysis of the iPRODUCE Use Cases

After the analysis of the use cases available in iPRODUCE's initial phase with their preliminary specifications, we have selected several use cases from [2] to be demonstrated with individual components of the GDP, as shown in Table 3, where Prio 1 indicates that the use case will be covered first; Prio 2 stands for the next implementation; Prio 3 use cases could be considered later depending on the resources.

Use Case	Main Feature	GDP component	REQ	Prio for GDP
Spain UC1	Smart bed headboard with electric & electronics components	3D Product Configurator (2D Layout Planner)	Specification of the electrical equip-ment, examples of design	2
Spain UC2	Smart adjustable gamer chair, health supporting with sensors & feedback	3D Product Configurator (+ Math Modeling)	Actual 3D model of a chair needed + electronics constraints	1
Denmark UC3	Sustainable parametric designs for festivals & similar venues	2D Layout Planner (+ Math Modeling)	Catalog of 3D models of assets	2
Denmark UC2 & UC1	Distributed Design of customized furniture	3D Product Configurator + Math Modeling	Definition of chair parameters	1
Greece UC1	Orthopedic back braces for children with IoT, gamifying for higher adherence & acceptance	3D Product Configurator (+ Math Modeling)	Some simple models of products, parametrically designed + electronics constraints	1
Greece UC2 & UC3	Lightweight & skin-friendly splints for children or pets, designed by users (patients)	3D Product Configurator (2D Layout Planner)	Some simple models of products, parametrically designed	3
German UC2	Microelectronics development (PCB)	2D Layout Planner, Math Modeling	Constraints for component positioning on PCB	3
Italian UC1	Movements in a mechatronic device (Robotics)	3D Product Configurator	Design Example, preferences for interaction interface	2
French UC1	Training, including product modifications	3D Product Configurator, Math Modeling		Potential usage
French UC2	Mobility eco system	2D Layout Planer	Examples of eco systems for urban mobility	3

Table 3: Use Cases mapping to Generative Design components

These use cases shall be utilized for the early evaluation and development of the GDP components. We also identified which inputs we need from these use cases for better alignment of the GDP development on the iPRODUCE's targets. The needed inputs were presented during a plenary meeting and are listed below:

- Examples of 3D & 2D designs of the currently considered products or layouts:
    - Including individual components as building blocks
    - To be parametric defining the space of modification freedom = ways for diversity
    - Combination with electric/electronics
- Information about the technologies to be considered in the use cases (and how they are expected to be covered in the GDP):
    - 3D printing
    - Laser cutting
    - CNC routing
    - Painting techniques
- Parameters of the product design (what can be possible to modify):
    - Production constraints
    - Allocation rules
    - What is needed for the price calculation or other output calculation?
    - What are other parameters that can influence the product design?
- Information about possible optimization targets:
    - Price minimization (together with the information how to calculate price)
    - Complexity minimization (together with the information how to calculate complexity)
    - Weight/material reduction
    - Stability increase
    - What are other numeric outputs that should be calculated and optimized?

2.3. Initial Mockup & First Prototype

The Generative Design Platform will contain three main components: Mathematical Modelling (called Companion in the web-based GUI), 3D Product Configurator, and 2D Layout Planer, as shown in Figure 5 below. For the initial prototype, the components will be delivered as individual tools, but will be integrated together and into the iPRODUCE OpIS platform later during the project.

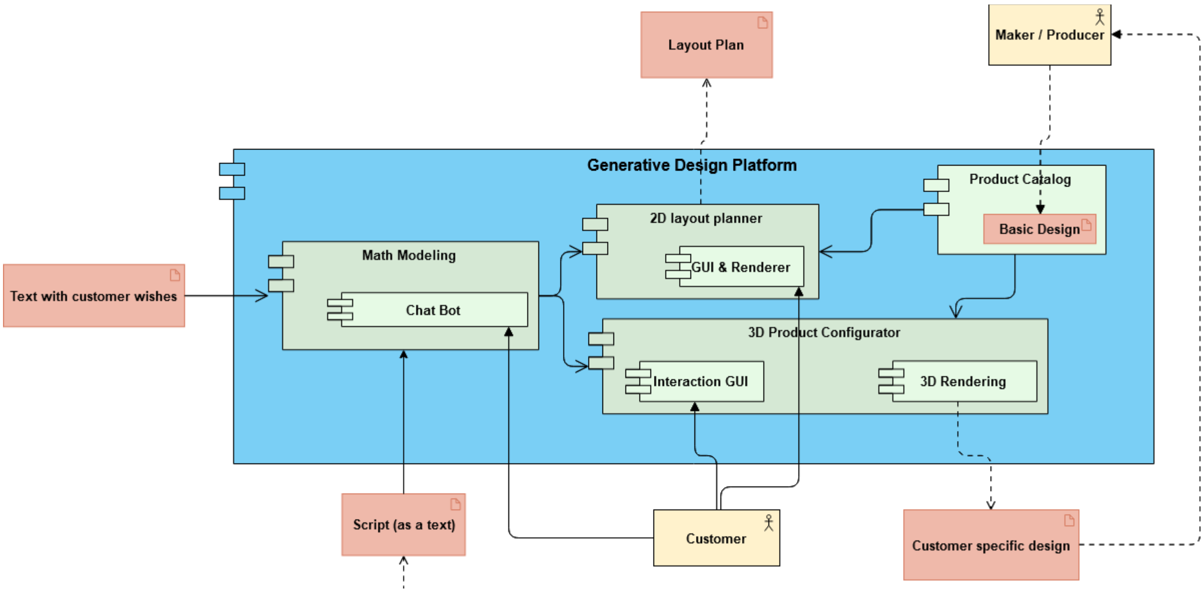

Figure 5: Generative Design Platform - Sub-Components

For the 3D Product Configurator mock-up, the first examples of selected use cases were developed in Rhinoceros 3D with Grasshopper plugin for parametric design models to demonstrate the main features of the future platform. Several parametric 3D models were prepared in form of Grasshopper scripts, i.e., visual scripting programs that generate multiple designs for a defined product depending on the parameter inputs from the customers. Also, different possible user interfaces were tested to operate the parameter settings: via GUI, through interaction with an electronic prototype with sensors, available HW like MIDI controller, available application on mobile devices.

The developed scripts, each as an individual file, are used after some adaptation to the selected technology within the further development of the platform. The platform shall provide the remote access to the visualization and interaction with such scripts over a web interface without any need for further tools on the customer side.

Figure 6: Mockup of 3D Product Configurator demonstrated at a plenary meeting

The mockups for the math modelling integrated with 3D configurator were proposed for advanced development during the project illustrated in the figures below:

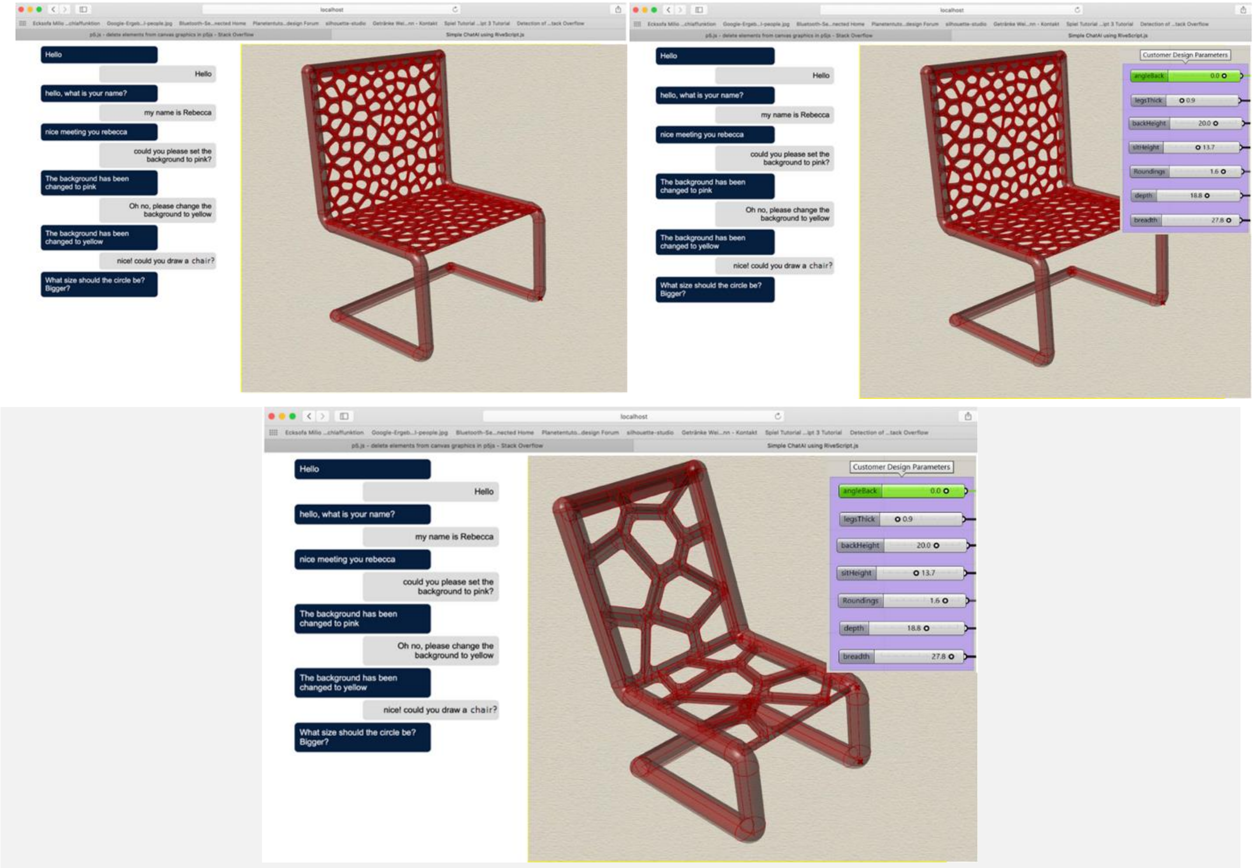

Figure 7: Mockup with Math Modelling (Chat-bot on the left) & 3D configurator (on the right)

At the first step, the chatbot of the Math Modelling component collects wishes from the user and transforms them into a form that can be used by other algorithms. This step is currently being developed as described in Section 4 with current tool status and progress. Next, the 3D Product Configurator will provide a proposal for initially pre-generated design corresponding to the customer requirements obtained from Math Modelling. Math Modelling shall also extract wishes for the modification of the available or selected product and propose a corresponding user interface, allowing to perform the desired modification. This is supposed to keep the GUI as simple as possible, while still having the whole space of the modification as rich as possible.

The sub-component "2D Layout Planner" was developed by SAG as a desktop tool for generating Interior Design plans. In this version, a genetic algorithm follows digitized Siemens restrictions and priorities to find the optimal location of the office furniture within different rooms. Exported results include PNG images and JSON–files with layouts. It is planned to adapt 2D Layout Planner for further 2D layouts design within iPRODUCE use cases such as: 2D model(s) of headboard with electronics allocation, PCB design, festival facilities planning. The Mock-up on Figure 8 below shows different versions of a furniture location within a defined room as three selected results of the optimization with the genetic algorithm.

Figure 8: 2D Layout Planner Mock-Up

We plan the initial light-weight integration of the three components within the Generative Design Platform when defining products linked to the corresponding configurator(s) and layouts and collecting results of the components within product definitions.

A user journey starting from the entry point of the GDP – product selection or definition – is sketched on the Figure 9 below.

Figure 9: User Journey for initial prototype: start with Products

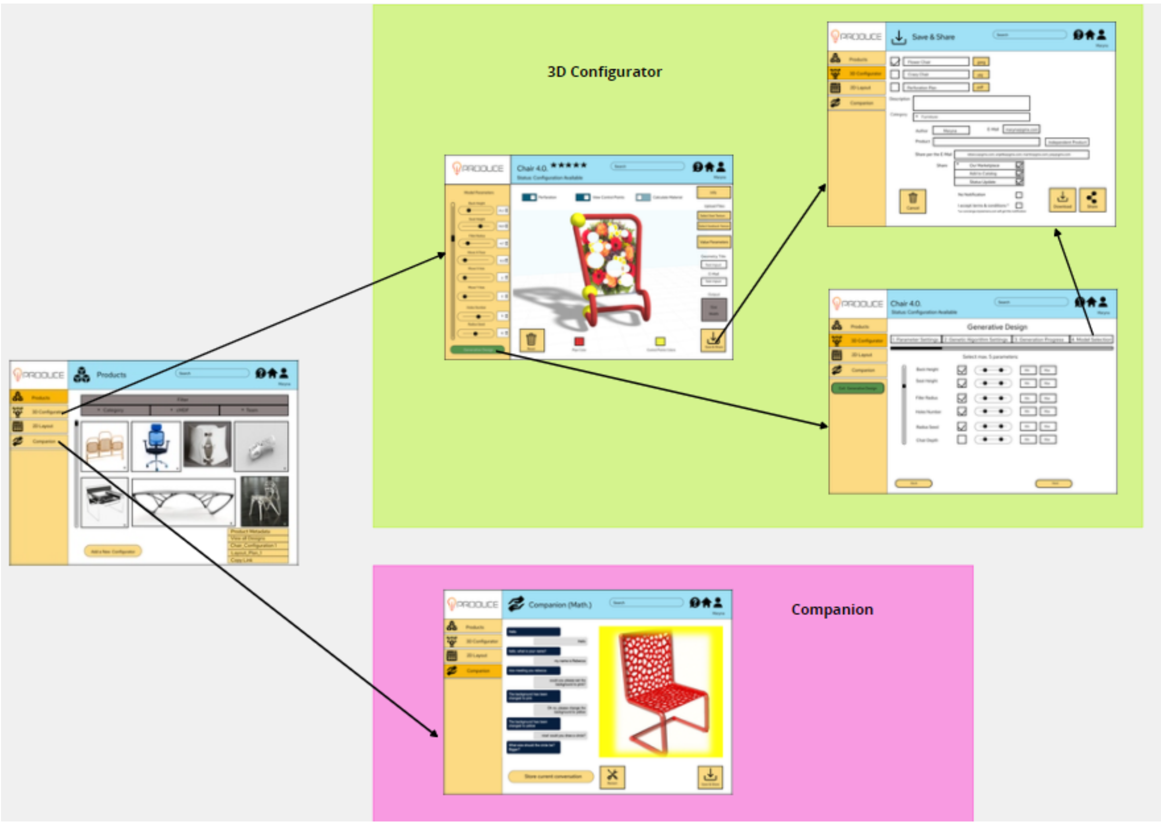

Figure 10: User Journey for initial prototype: 3D Configurator & Companion (Math Modelling)

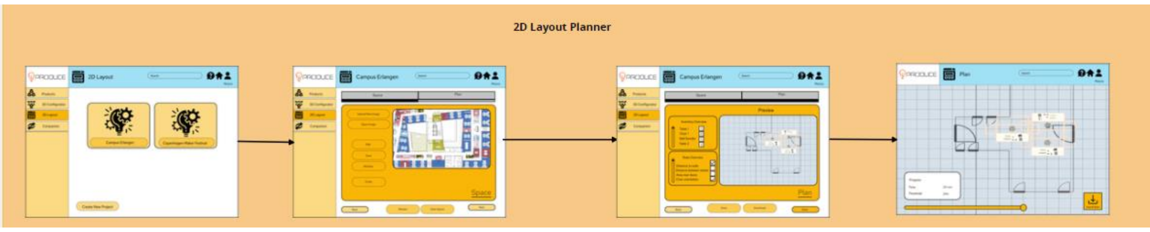

Figure 11: User Journey for initial prototype: 2D Layout

3. Generative Design Platform Architecture

3.1. Context View

3.1.1. System Context

The responsibility of the Generative Design Platform is to provide the user with appropriate means to explore the solution space in 2D and 3D. To accomplish this, the Generative Design Platform integrates three independent tools, which will be formed into a partially integrated solution in the course of the iPRODUCE project. The architecture for the GDP described in this section was developed in alignment with the general architecture for the iPRODUCE OpIS platform [3].

Figure 12 shows the most relevant neighbor systems – which is not a complete list of available software components in iPRODUCE.

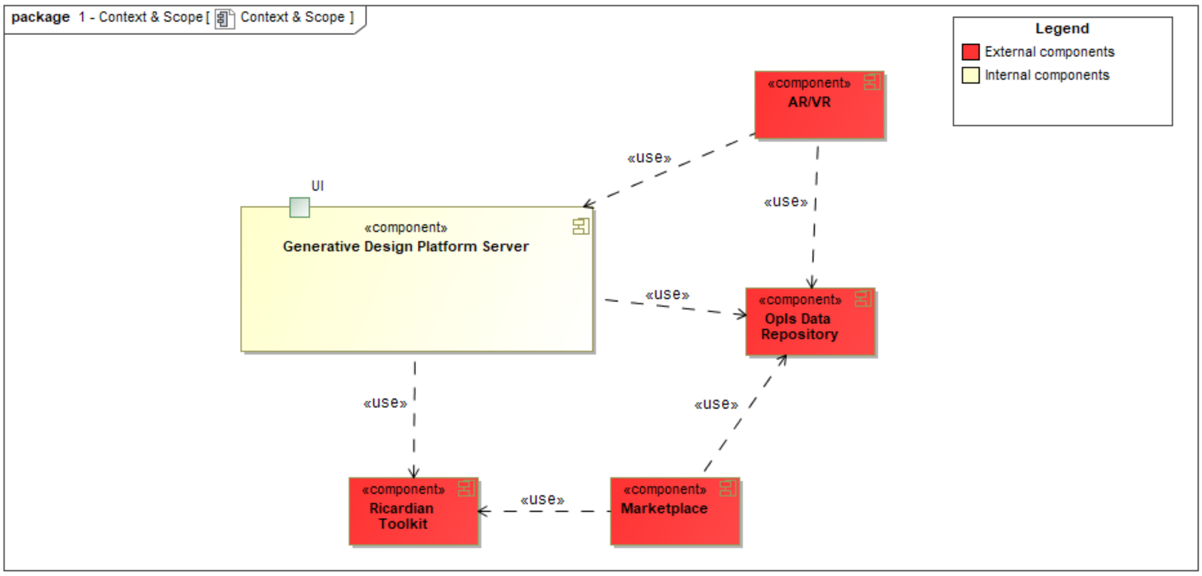

Figure 12 System Context of Generative Design Platform

Internally, the Generative Design Platform is divided into the following main functional components: Mathematical Modeling, 2D Layout Planner, and 3D Product Configurator. This is shown in Figure 13, which depicts, in addition to the main components, also the Artifact Model containing descriptions of all artifacts, and these are persisted in the Artifact Repository. The Optimizer component uses the Artifact Model to generate new artifacts based on optimization criteria. Corresponding parameterizations are persisted in the Optimizer Repository.

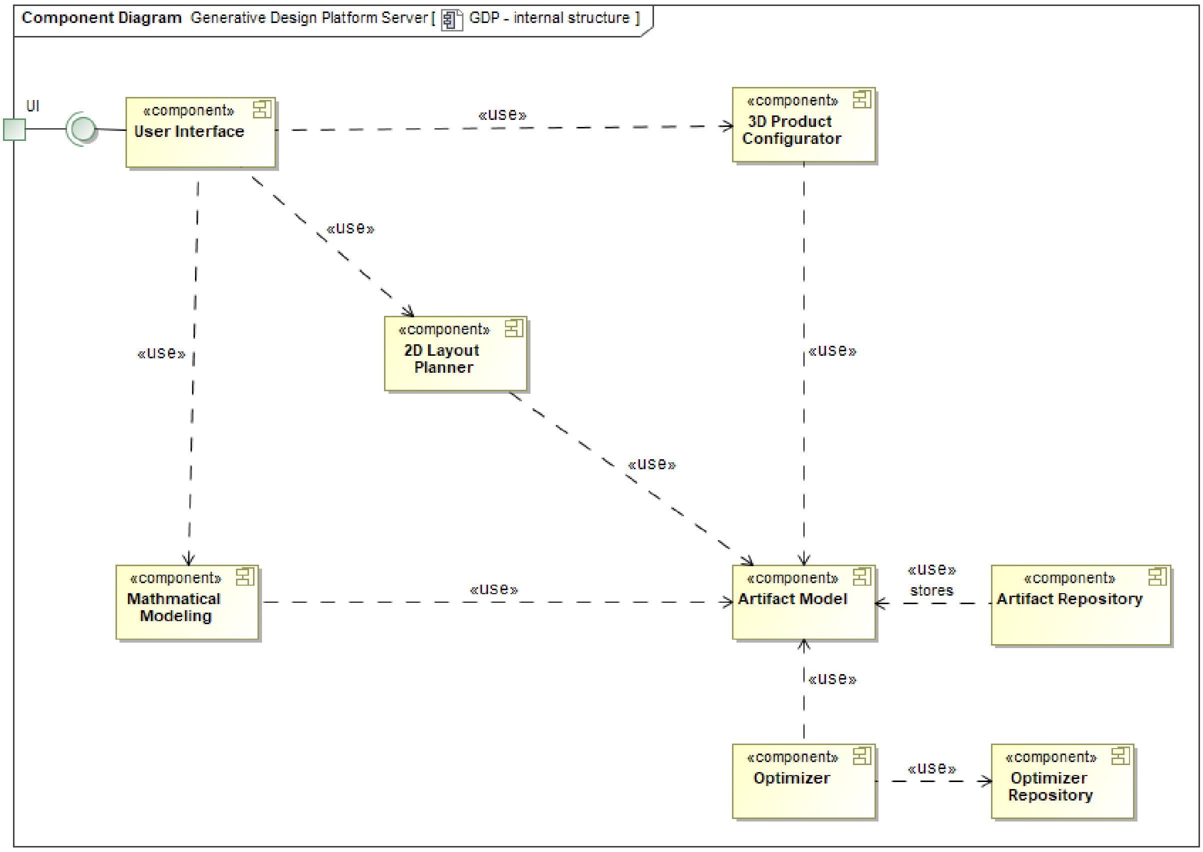

Figure 13 Internal structure of Generative Design Platform

3.1.2. Domain Model

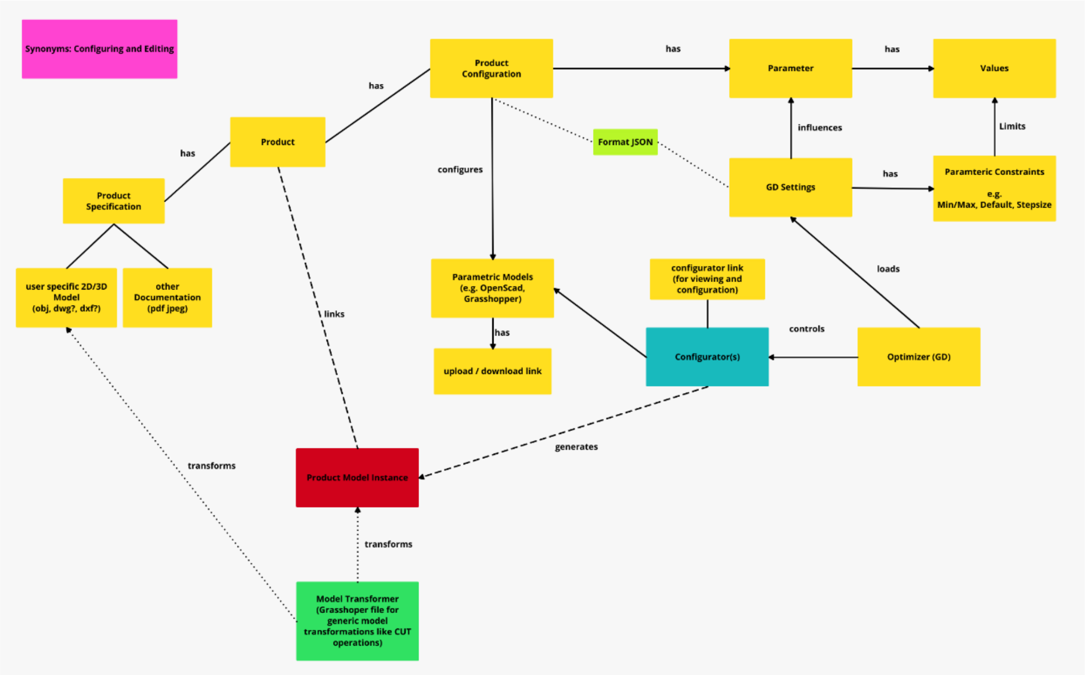

Figure 14 Domain Model for the GDP

The Domain Model in Figure 14 describes a rough sketch of the concepts considered relevant in the technical domain of Generative Design Platform.

Parametric Models (e.g., OpenScad [12], Grasshopper [5])

Parametric models are scripted CAD files that have a predefined flexibility, i.e., a user can set the degrees of freedom of the model (parameters such as length, width, height, density), leading to a customized result.

Product Model Instance

This is a concretization of a product that was created from an abstract product specification, for example, with the help of a configurator.

Product

In the context of the GDP, a product or project is an abstract entity that holds all the data for a piece of work together.

Product Specification

A product specification contains machine and human readable documents that contain solutions as well as requirements.

Parameter

A parameter is a quantity that can be specified by a value such as seat height, length, width, weight, color, name. Only numeric parameters can participate in the optimization process.

Values

These are concrete values that can be set for parameters such as 20 cm.

Parametric Constraints (e.g., Min/Max, Default, Step size)

These are constraints on the degrees of freedom for parametric models so that the generative design does not use the entire available solution space. For example, if a user wants to generate a chair, (s)he may restrict the height of the seat in a range between 40 and 80 cm, which is less than the possible maximal total chair height if a user wants to keep the rest space for the chair back.

GD (Generative Design) Settings

GD Settings are hyperparameters that are used for generative design. They are typically options to control evolutionary algorithms.

These can include inheritance settings, termination conditions for an optimization, population size, initialization settings, etc.

Configurator(s)

A Configurator is a software component into which products can be loaded to be configured or modified by a user.

upload/download link

Via an upload or download link, a user is able to add files to a product or to download the product in whole or in part. For this purpose, the user will select on a corresponding UI, which product artefacts (s)he would like to consider for a download.

configurator link (for viewing and configuration)

This is a URL that can be inserted into the browser to navigate directly to the model in the configurator.

user-specific 2D/3D Model

These are 2D or 3D models, designed by users, that are necessary for further processing in a product. Possible file formats for this are: obj, dxf.

other Documentation (pdf, jpeg, txt)

This includes any product descriptions that are not directly machine-readable/interpretable, such as images or text-only information used for communication among team members.

Model Transformer

A Model Transformer can apply special transformations to a given 3D model input file. For the GDP, these transformers are implemented in Grasshopper [5] and allow users without CAD skills to perform modifications on their 3D geometries.

Product Configuration

This is a collection of parameter values, set by an end customer when configuring a parametric model. This is serialized in JSON format.

Optimizer (GD)

An GD Optimizer implements an evolutionary algorithm and checks candidates generated by random mutations for fitness. If a candidate is fitter than other candidates, the probability that its genes will be passed on to the next generation increases. The detailed process of evolutionary algorithms is introduced in Section 1.2.

3.1.3. Interfaces (to external systems)

Depending on the type and expertise of the user, the Generative Design Platform offers different levels of interfaces.

In the simplest case it is a graphical or textual web-based UI. In case the user is a programmer, API endpoints are also offered.

Each of the following components provides web-based UI (currently not all of them are integrated with each other):

- GDP as a kind of management system, which binds together the different tools
- Mathematical Modelling
- 3D Product Configurator
- 2D Layout Planner

The respective tools offer a set of API endpoints which are mostly for internal use. Data import and export interfaces are planned to be released to iPRODUCE project partners who have legitimate interest. If run in development mode, a swagger endpoint is generated [26].

Disclaimer: API endpoints and data schemas are in an early prototyping state and thus subject to change.

The Generative Design Platform offers two API interfaces:

- /api/Management/ for management of product and project entries
- /api/SearchAndBrowse/ to search through the generated projects and products on the platform.

Management:

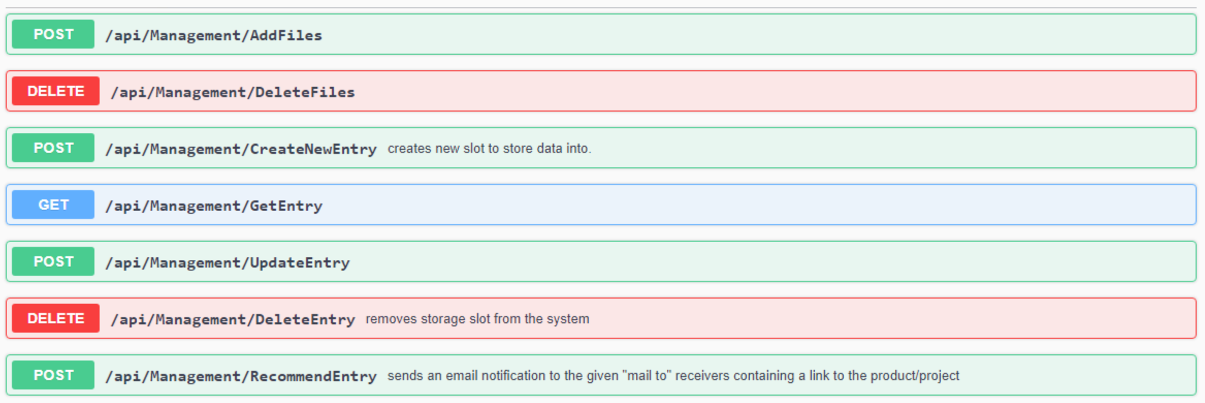

Figure 15 Current management API

To view and edit products/projects, API of the corresponding editors/viewers will be used.

SearchAndBrowse:

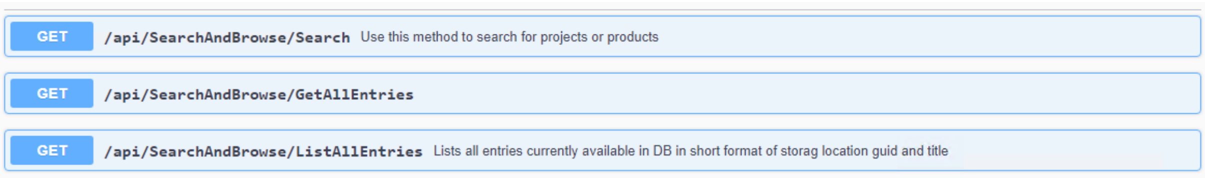

Figure 16 Current SearchAndBrowse API

Data schemas currently used in Management and SearchAndBrowse API:

- BucketLocation
- Category
- ItemType
- User
- Configuration
- Alias
- Comment
- Image
- FileAttachment
- Rating

- Product
- Make

Current limitations in API "Management":

- Import from other OpIS applications/tools from iPRODUCE partners
- Export to other OpIS applications/tools from iPRODUCE partners
- Direct upload from a user to product/project
- Direct download from product/project to the user

3.2. Functional Decomposition

3.2.1. Internal and External Use Cases

Figure 17 depicts planned main functionalities of the first prototype, to be delivered in M18.

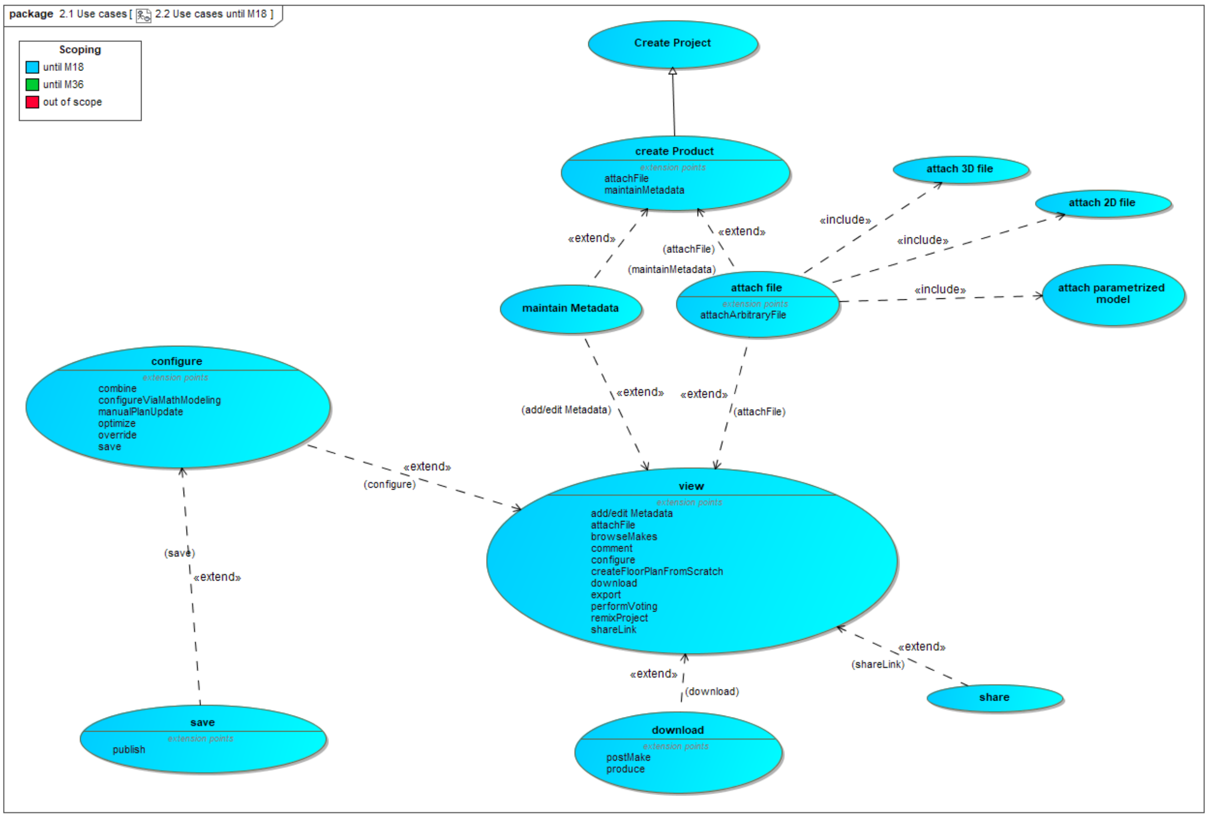

Figure 17 GDP scope until M18

Table 4 below lists the main features with their descriptions.

Most significant use cases in this implementation phase are highlighted in yellow color.

Use Case name	Description and Notes
attach 2D file	Graphics used for layouts, textures, logos, pre-views, sketches of preliminary ideas, etc. Possible formats: dxf, jpeg, bmp, png.
attach 3D file	Geometries defining 3D design of a product. Possible format: obj (a format accepted by all OpIS partners)

attach file	This use case is a general placeholder to attach a file to a product or project. The real implementation is outlined in included or extending use cases (for scoping purposes).
attach parametrized model	Within this use case, an experienced user can upload his/her Grasshopper files. The GDP will process these files and create configurator sites.
configure	The predefined variability of a product/project will be resolved when a user selects parameter values defining the 3D/2D forms. Structural modifications of products/projects are included in use case "combine". Creating a new floorplan is covered by "create floor geometry".
create product	In this use case, the user can create a product. This should be possible by starting the workflow corresponding to his/her role on a web page. The minimum data required to create a product or project is the title and the associated user as user ID, email address or the like.
create project	This use case is a special extension for "create Product" because the component 2D Layout Planner is about projects. As a logical consequence, the stereotype "Product" is also a specialization of "Project". The other use cases for Project are analogous to those for Product - which means, they are valid for both products and projects. Overall functionality is defined in Section 4.3
download	In this use case, product data are offered to the user for download in the browser. Data are not pushed to the iPRODUCE Repository, but stored locally. This activity will be logged: time of the download, affected files, user (if possible to identify), and IP address.
maintain metadata	The user is able to create, read, update and delete (CRUD) product/project meta data, which have a simple textual or numeric representation.
save	In this use case, the user can save his/her modifications to the product or project. The user may choose to publish his/her work. The user is asked to provide a meaningful way to name his/her configuration – a default is proposed, i.e.: <product or project name>_<timestamp> The user is free to accept or overwrite this name.
share	In this use case, users are given the possibility to recommend a product/project entry of the platform to another user. The recommendation will be send via email and can also be targeted towards users who are new to the platform.
view	In this use case, a product or project is just exposed to the user and not modifiable. The user can transit into the configure step, which actually changes the model and immediately updates the visualization.

	Within the corresponding UI, the user will also be able to select and execute further connected use cases: “comment”, “rate”, “share”, “download”, “publish”.

Table 4: Primary features of the GDP (developed by M18)

Figure 18 depicts further planned use cases to be implemented in the final version of the GDP prior to M36.

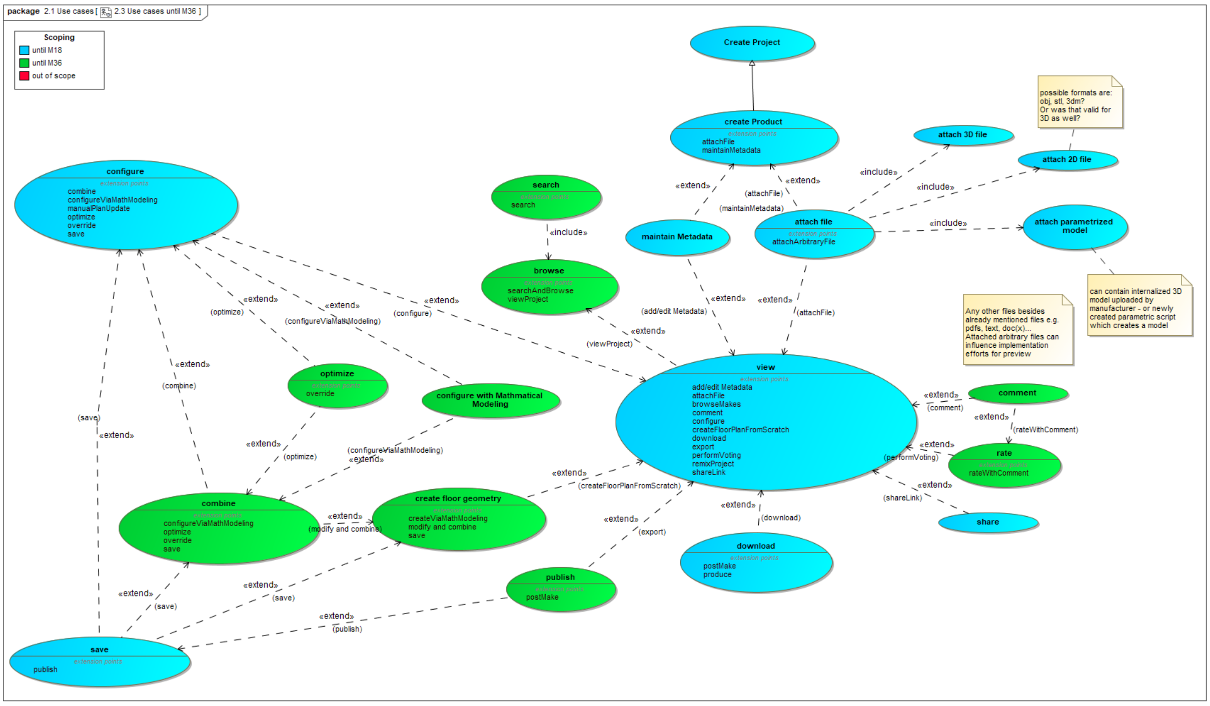

Figure 18 GDP scope until M36

The table below lists further features that are planned for the implementation after M18.

Most significant use cases in this implementation phase are highlighted in yellow color.

Use Case Name	Description
browse	The Browse use case can be carried out via categorical browsing, 'advertised browsing' or by browsing through the results of a search. Advertised browsing can mean: - new projects - projects actively discussed - projects actively rated (no. of rates) - projects ranked good - projects ranked bad (can be used to search who needs help) - projects made by awarded people (Platform effect!)
combine	In this use case, new elements can be added to a model. Two different scenarios are considered: 1. A floor plan – “combine” means placing furniture and adding constraining rules and their importance among each other. 2. 3D model – “combine” means adding other 3D model parts in a joint

	coordinate space as well as constraints/connections/dependencies between them.
comment	This use case gives users the possibility to add comments to existing products and projects.
configure with Mathematical Modelling	In this use case, the user shall express his/her wishes (changes to the status quo he/she is currently seeing) in the format of written natural language. The interaction must be possible in English language; other languages are optional. The language needs to be told explicitly (if more than one) at the beginning of the 'conversation'. The user shall have a clear guidance of what he is able to do interacting with the system.
create floor geometry	This use case allows a user to freely create a floor plan from scratch. This implies outer boundaries (geometry) of the room(s)/floor(s). Filling the defined geometries with objects is not part of this use case.
optimize	This use case extends others and tries to find optimal solution towards a (set of) optimization goal(s) using generative design. There are two different optimizer scenarios, like already outlined in the "combine" use case. Optimizing 2D layout implies generating different solution proposals, storing them and rating them according to the fulfilment of the optimization rules. Optimizing 3D layout (as an extension of the use case “configure”) means to generate solution proposals based on the levels of freedom specified in the corresponding parametric model, targeting towards the selected optimization goals. Example: If a chair is configurable in height, the user is free to decide if (s)he wants to lock (constrain) this parameter for optimization by assigning a fixed value or to allow adjustments of this parameter within a selected boundary – e.g., setting the range from 75cm to 85cm. Optimizing combined layouts (as an extension for the use case "combine") focuses not on the individual 3D objects, but their relation towards each other based on given constraints. Example: If we add lamps, power supply, and a controller to the joint coordinate space of a bed, then we might want to optimize for less cabling (shorter, less material). This use case can be extended by manual override – this means, a user can select a generated "nearly perfect" candidate and update it further manually.
publish	In this use case, the user can choose to publish his/her work saved previously. The system will ask the user which parts/files should be

	included when publishing – the user has the ability to select. In case nothing is selected, the publishing action does not make sense and thus, is disabled. Chosen files plus Product/Project meta-data gets serialized in the file system, compressed as zip and uploaded to the attached iPRODUCE Data Repository. The export procedure will be logged: time of the export, affected files, user (if possible) and IP address. A user may select only specific configuration(s) out of the whole product specification (as attachments) to be published. The user can also invoke the use case "publish" for a currently created configuration within a "configure" use case.
rate	Within this use case, the user can vote for a specific product or project. The vote will be captured and displayed (aggregated) at the rated item. It could be possible to even add a textual comment to the rating e.g., as justification for the chosen rating.
search	In this use case, the user can use simple single word search terms to query the list of products and projects and thus create a filtered search result list, which he can browse as next step. Search is an optional use case as the user can directly start browsing categorized lists, as defined in use case "browse".

Table 5: Secondary features of the GDP (developed after M18)

3.2.2. Architectural significant requirements

ISO/IEC 25010 [29] and former ISO/IEC 9126 [30] describe non-functional requirements (NFR) which are also referred to as "system qualities" which are clustered in a tree-like structure.

Figure 19 Model of system qualities according to ISO/IEC 25010 (not a complete list)

In the context of the GDP, the most significant architectural drivers are:

Performance Efficiency:

“Degree to which the amounts and types of resources used by a software product, when performing its functions, meet requirements.” (Source: [29])

The fact that these developments are prototypes and not real products weakens this requirement a bit. Nevertheless, evolutionary algorithms, 3D rendering/calculations, and language processing with Mathematical Modelling are resource extensive tasks which can be long running tasks.

To achieve a good UX the compute resources and parametrization of algorithms need to be chosen wisely to result in a good trade-off between cost/duration and quality of the generated result.

Wherever possible parallelization or even hardware acceleration of computation needs to be considered to make use.

Availability:

Availability is a sub characteristic of Reliability and is defined as follows:

“Degree to which a software product or component is operational and accessible when required for use.

NOTE: Externally, availability can be assessed by the proportion of total time during which the software product or component is in an up state. Availability is therefore a combination of maturity (which governs the frequency of failure), fault tolerance and recoverability (which governs the length of down time following each failure)." (Source: [29])

This development is considered as prototype development which smoothens the requirement by a considerable amount. Nevertheless, we decide for IT hosting based on AWS for a good availability of the overall infrastructure. Our own containerized infrastructure allows restart on failure and can return to a working condition. Scalability itself in no considered under this NFR.

Interoperability:

“Degree to which two or more software products or components can exchange and (commonly) use information. (based on [29])”

For the GDP we understand interoperability of the possibility to exchange data between different SW components developed by our partners in the project – for this purpose we try our best to define stable versioned and non-changing interfaces as well as stable (add only) data definitions.

3.3. Logical View

3.3.1. Static Aspects

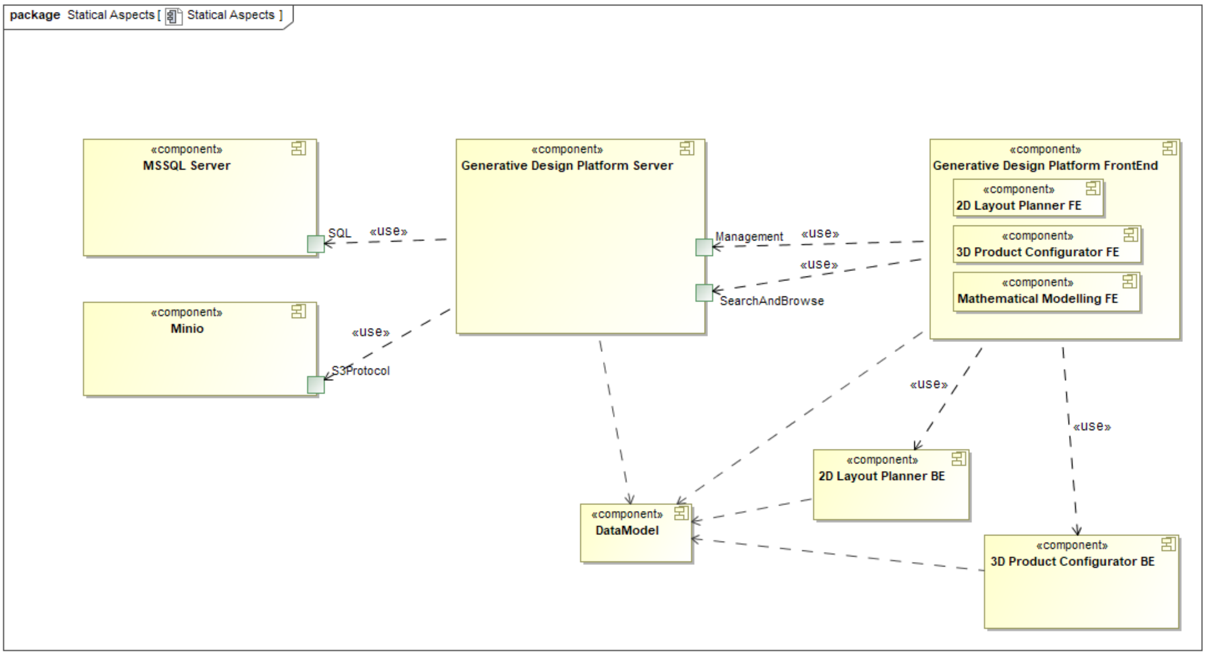

Figure 20 The major components of the GDP

MSSQL Server:

For storing most of the data of relational nature, we use the latest Docker version of Microsoft SQL Server, from Microsoft's Container Registry [24]. Without further initialization by a product key, as in our case, it is the Developer Version 2019 for Linux, a version for development and testing which is not released for productive use.

Minio Server:

Since storing files in a relational database schema is inconvenient, we store all such files in an object storage database, which is fully compatible with AWS S3 [23]. As with AWS S3, MinIO [25] has the concept of buckets – these are named as GUIDs and referred to by name in the relational database.

Generative Design Platform Server:

This component is the management system that hides the concrete persistence from the user. Currently, it provides two endpoints to the outside. The component is implemented in C# and is based on DotNet Core Framework version 5.0. To access the object storage, the application uses Amazon's AWS S3 SDK, which is available via NuGet. To access the relational database, the Entity Framework Core object-relational mapper is used.

Datamodel:

The data model is implemented in C# as a library and is used by the Generative Design Platform Server and made available to callers via REST endpoint. "Manual" serializations, if required, were done with Newtonsoft Json (source: NuGet).

Frontends:

All frontends are implemented in HTML, JS, CSS and one in React. Where available, they communicate with their respective backends via REST interface. For a subset of the frontends, an OTS platform called ShapeDiver [14] is used.

3.3.2. Dynamic Aspects

Start-up behavior of the GDP

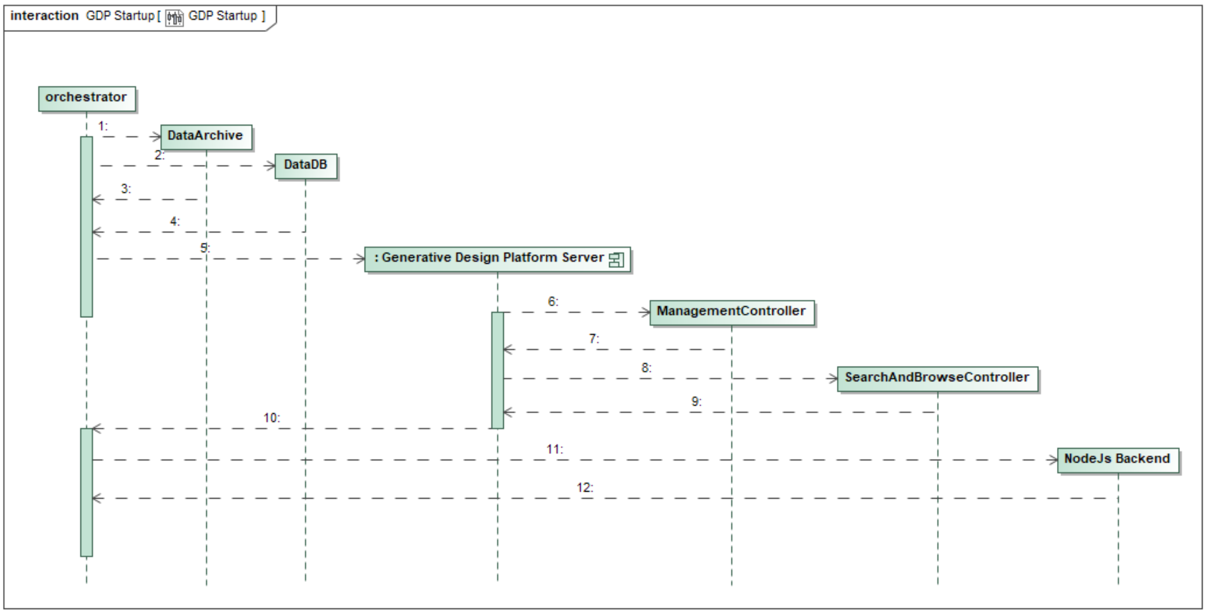

Figure 21 Sequence of the GDP start-up

Project creation

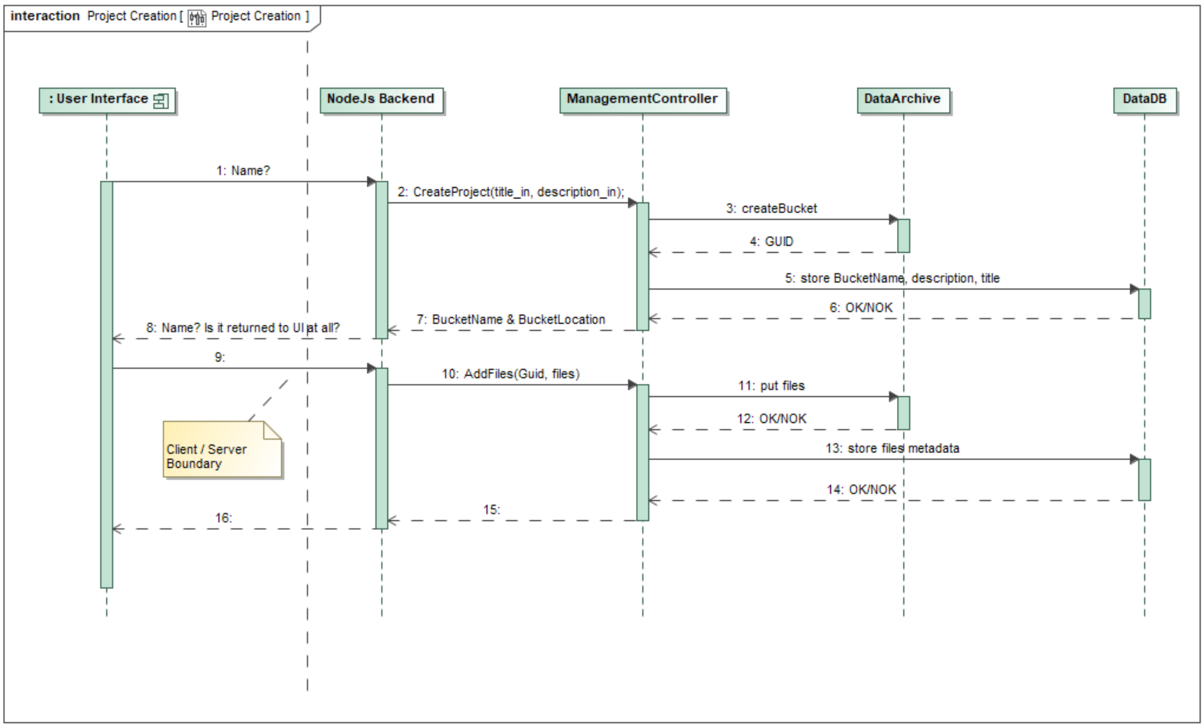

Figure 22 Sequence for creating a product/project

3.4. Deployment View

Currently, all components are deployed in distinct docker containers, orchestrated via docker compose.

Deployment will be into a single node server – size depends on performance requirements.

Scale-up possible if situation requires. Scale-out possible as well – but likely to get expensive due to clustering.

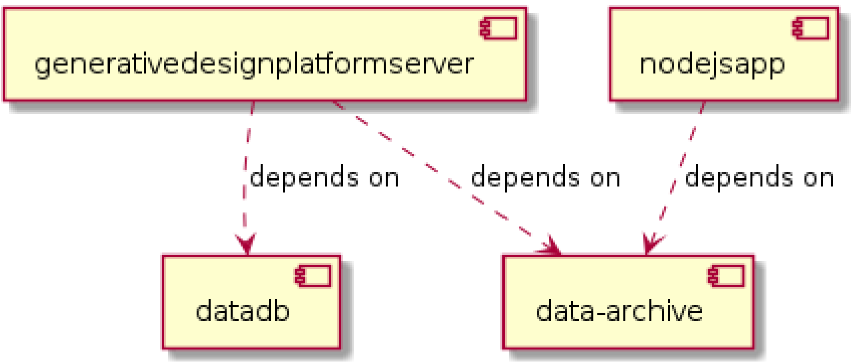

Figure 23 Current docker deployment (not all containers are integrated yet)

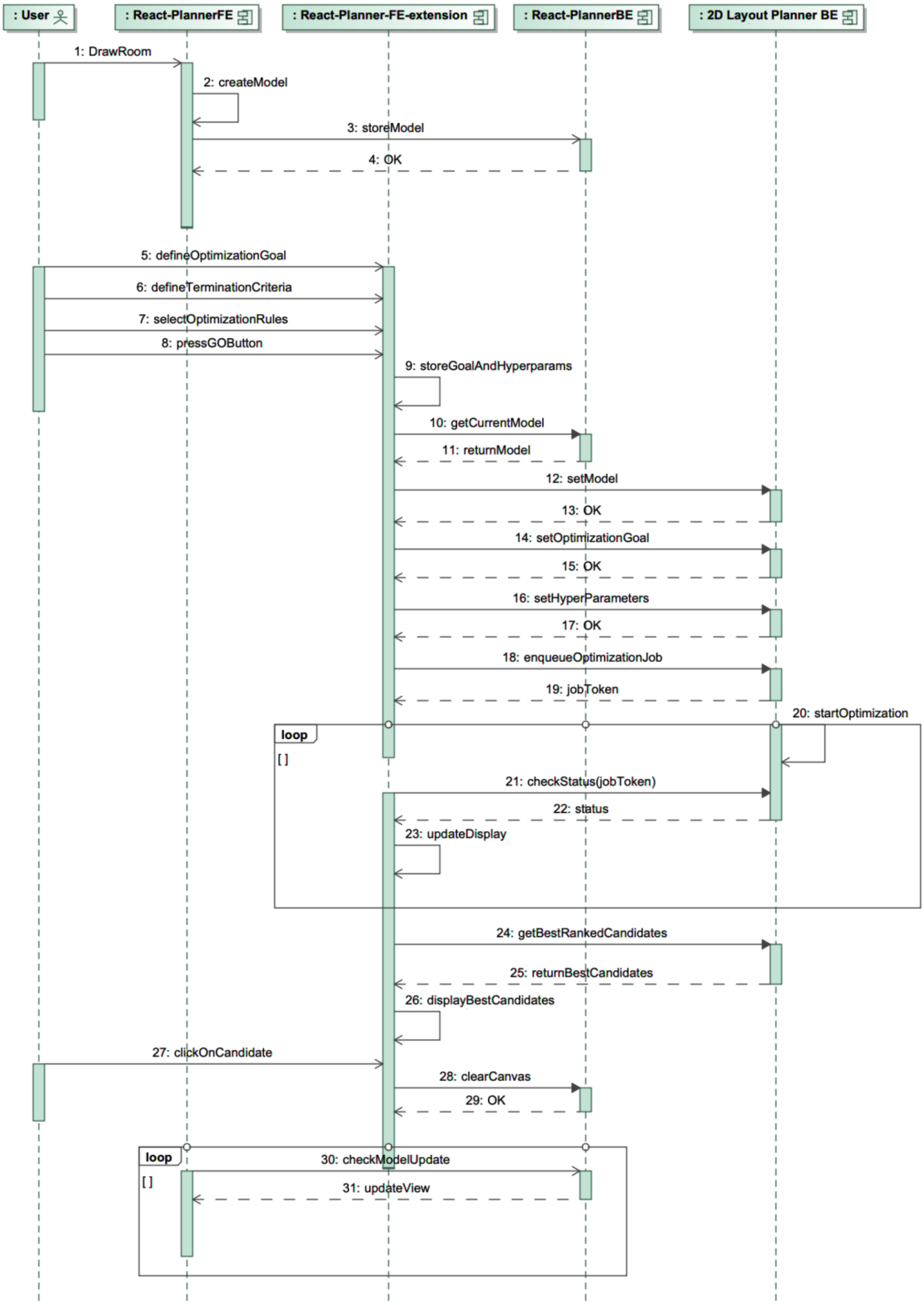

Figure 24 A typical 2D layout optimization sequence

4. Detailed Description of the Components: Current Development State

The Generative Design Platform (GDP) is currently implemented as a standalone module to run in a cloud infrastructure and offer a web-based frontend to user groups with different target uses. It will be initially deployed on Amazon Web Services (AWS) for the purpose of prototype demonstration and evaluation before the integration into the OpIS platform. As there will be a link to the platform, this link can be published within the OpIS. Thus, the GDP can be accessible through the OpIS for a light-weight integration.

The platform will be implemented as a social community enabling the users to communicate with each other by defining products together, contributing to design models, and influencing one another's results and processes. Because this is a prototype development, user management and access management won't be implemented in the GDP in the course of iPRODUCE. We will rely on the corresponding components from Marketplace in WP4, task T4.3.

The following table lists the main functionality of the GDP:

Features to be developed	Priority
Product definition	Must
Selection of basic 3D design for product	Must
Selection of basic 2D layout	Must
Extendibility of the product catalogue	Should
Parametric design, online modification of design parameters	Must
Design extendibility by further elements: electric, electronics, etc.	Should
Constraints specification in an intuitive way	Must
Combination of parameters and constraints into a target study = complex constraint	Could
Connections to community (for rating)	Must
User-friendly and intuitive interface	Must
Gamification for end users	Should
Results sharing (e.g., download/export)	Must

Table 6: Main Features of Generative Design Platform

As described in Section 3.1.1, there are three main components constituting the GDP: Mathematical Modelling (Companion), 3D Product Configurator, 2D Layout Planner.

In the following sub-sections, the three individual components of the GDP are described, including their current development state.

4.1. Mathematical Modelling (Companion)

This component is being implemented from scratch during iPRODUCE project. The software will allow automatic extraction of numeric and spatial information from plain textual description. It will be extended with a chat bot to refine the information extraction. This component shall receive input text either from the user directly via the chatbot or out of any textual inputs from other components. It can also be integrated with the Video Intelligence tool (also developed by SAG as a part of the iPRODUCE

Training Kit) to receive video transcripts as textual input and, thus, extracting numeric and space information from the users' videos or audios.

Main Functionality	Development state
Recognition of clear textual input in English	Implemented
3D objects recognition	Implemented
Object manipulation	Implemented
Spatial rules recognition	Implemented
Extendibility of the recognition pattern (recognize more elements, transformations or requirements)	Planned
Connection to 3D configuration tool	Planned
Combination of parameters and constraints into a target study = complex inputs	In progress
Gamification for end users through user-friendly interface	In progress
Result export	Planned

Table 7: Functional Requirements – Mathematical Modelling (Companion)

Currently, focus is laid on the interpretation of spatial instructions given in natural language. The implemented engine accepts small text passed from a chatbot-like UI running on an Ubuntu server via Python and outputs a formatted JSON response with the instructions to manipulate and place 3D objects in a 3D editor. The engine is optimized to understand English and the 3D editor will run in a web browser. For this the following classifications have been identified and processed:

1. Creation and Spawing of objects:
    a. Create/ Make primitive (box, cylinder, plane, sphere)
    b. Name object
    c. Scale primitive (size, "bigger than X", "smaller than X", "20% bigger")
    d. Position (in respect to world, next to G, next to H)
    e. Rotation (in respect to itself, in respect to object)

```
Enter command
> Put a torus at location 12,34,56
create
Please specify figure name
> Torus05
{'type': 'torus', 'name': 'Torus05', 'location': '12,34,56'}
Enter command
> Rotate it 30 degrees
rotate
Please specify figure name
> Torus05
Please specify rotation axis
> x
{'name': 'Torus05', 'rotation': '30', 'axis': 'x ', 'pivot': 'None'}
Enter command
> Make it 20% bigger than Sphere03
scale
{'name': 'Sphere03', 'direction': 'bigger', 'pivot': 'sphere03', 'scale': '20%'}
Enter command
>
```

Figure 25: Object recognition with Math Modelling

2. Object Manipulation:

a. Duplicate X
b. Remove X
c. Translate / Rotate / Scale X in respect to world or object Y)

```
Enter command
> move dragon to position 10,18,40
move
{'name': 'Dragon', 'position': '10,18,40', 'pivot': 'None', 'direction': 'None'}
Enter command
> translate dragon to 10,18 relative to mouse
move
{'name': 'dragon', 'position': '10 18', 'pivot': 'mouse'}
Enter command
> move dragon 10 units to the right
move
{'name': 'dragon', 'position': '10', 'direction': 'right'}
Enter command
> duplicate dragon
copy
Please specify a name for new object
> house
{'name': 'dragon', 'newname': 'house'}
Enter command
>
```

Figure 26: Object manipulation with Math Modelling

3. Object Selection and Grouping:
    a. Select all of Type
    b. Select by name
    c. Select all
    d. Group Selections, give Group Name

```
Enter command
> Select sphere01
select
{'name': 'Sphere01'}
Enter command
> select table
select
{'name': 'table'}
Enter command
> select all cylinders
select
{'type': 'cylinder'}
Enter command
> Group selection, give it a name Cylinders
group
{'name': 'cylinders'}
Enter command
>
```

Figure 27: Object grouping with Math Modelling

4. Detection of Spatial Requirements/Rules:
   a. Detect names, dimensions, context and rules for the following
   b. “An X should be placed every n meters along Y or opposite from Y or in n distance from Y”
   c. “Every X needs a Y”
   d. “X can never be placed near Y”

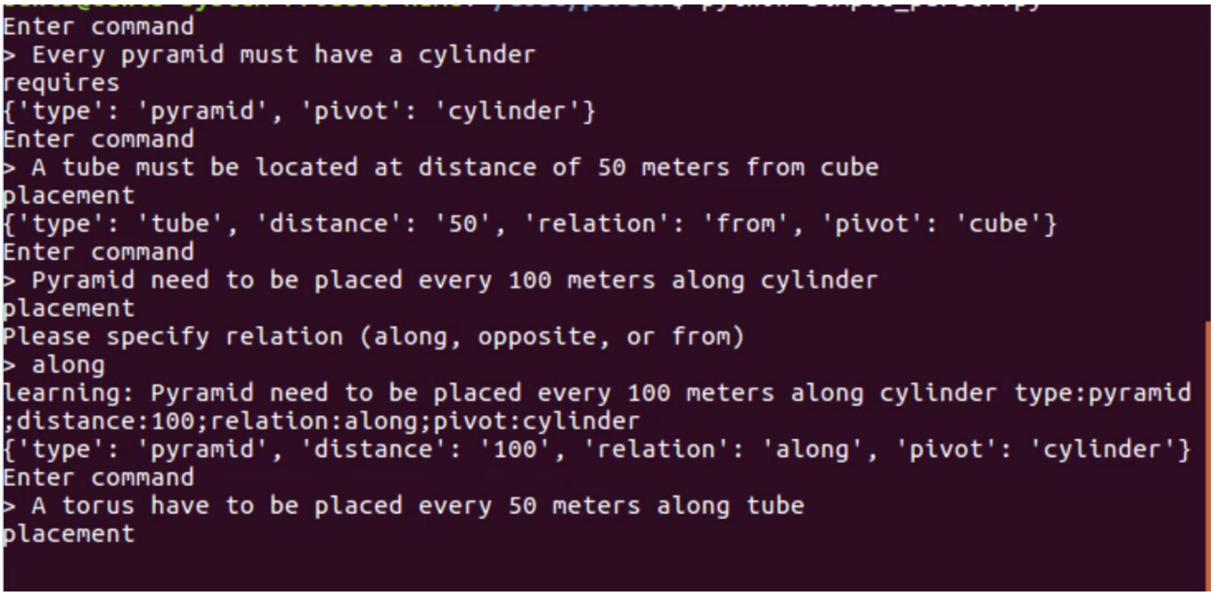

Figure 28: Spatial requirements detection with Math Modelling

5. Solution Generation with Genetic Algorithms. Input should specify the following:
   a. Which objects in context to other objects
   b. Ranges (All dynamic parameters of objects in their min/max potential)
   c. Constraints (Parameters to be left untouched)
   d. Min/Max statement (as little as possible, as big as possible, as wide as possible, as far away as possible, as close as possible)
   e. Validate whether requirement is possible and what conflict

Hence, to classify instructions as input for genetic algorithm, the system should look for phrases like "as big as", "as small as", "as far away from each other" or "as close together".

Then it will look for constraints, ranges and context, e.g., "There should be as many boxes on this plane as possible, without changing the size and rotation". In this case it knows that the "boxes" are to be manipulated or spawned and that it is their "placement" which can range in x from 0 to width of "plane" and y from 0 to length of "plane" with constraints of size and rotation being untouched. If it just said "There should be as many boxes on this plane as possible” with no constraints, the system would be free to change rotations and sizes too; those would also become ranges from 0-100%.

Next, it needs to measure fitness levels – so in the original genetic algorithm input it needs to detect if a minimum value or maximum value is to be reached. If the user says "as far away from each other as possible", the engine would detect that it needs a "maximum" value and the higher the better. If the user says "as close together", we would know that he/she wants a "minimum" value. Same with “as small as” = "min" (the ranges would be scale in this case) or “as big as possible” = "max". "As wide as possible" or "as many as possibly" are max values.

Regarding the conflicts, if the user input is classified as a range but also a constraint, e.g., "make them as big as possible, without changing the size", then the system should output a conflict.

The output JSON format is then sent to an open-source three.js-based 3D web editor for visualization:

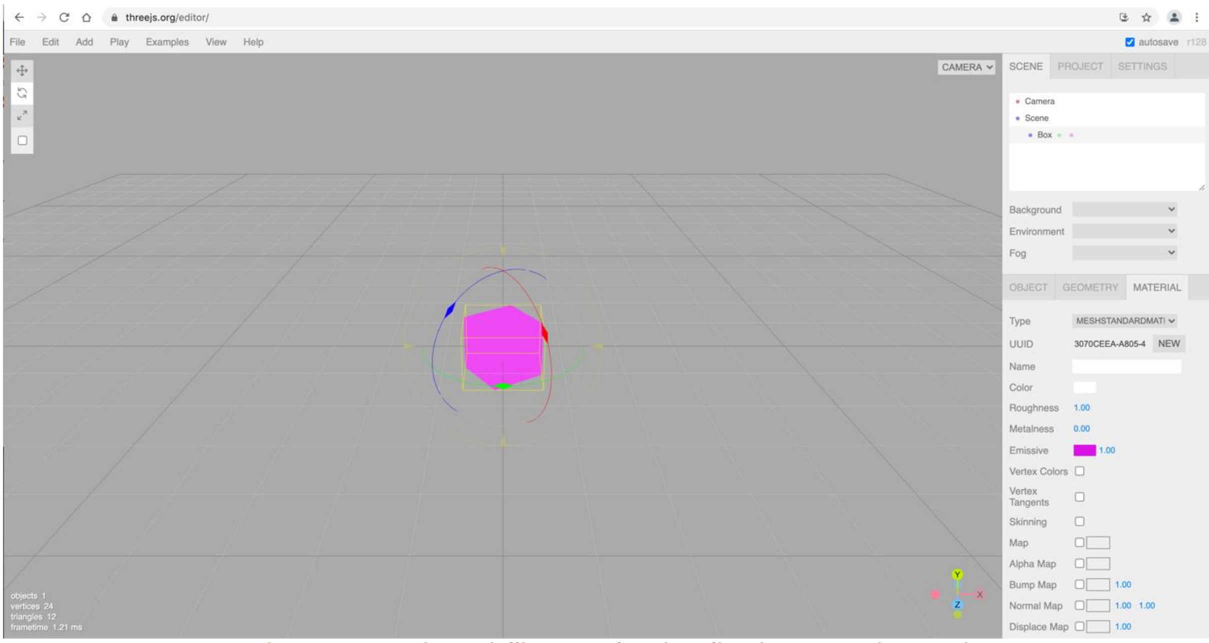

Figure 29: Math Modelling results visualization: UI to be used

This component is currently not integrated with other components, but it is planned to be used in combination with 3D Product Configurator and during product definition.

4.2. 3D Product Configurator

This component is implemented from scratch in the iPRODUCE project. In the first prototype, the software will allow end customers to select basic design of a product type (headboard, chair, bench, etc.) depending on a use case and configure it corresponding to his/her wishes. In addition, producers / involved MakerSpaces may use the online configurator to add new product types and manufacturing constraints to the new products. A product design configured by one end customer can be stored in the design catalogue within the corresponding product definition for further production or for rating by other users.

Skillful makers, engineers or researchers can add a new or improved/extended configurator to a selected product definition within the GDP. This will be possible via upload of a transformation script file (.gh) or its XML representation (.ghx) containing the instructions and constraints for the generation of a new product design and defining the freedom space for modifications. The corresponding web user interface will be generated automatically for each transformation allowing end users to interact with it via different UI elements (sliders for numeric values with limitation, input fields for text and unlimited numeric values, check-boxes, color selector, file selection) as well as via moving control points that define 3D geometries.

The main functional features of the 3D Product Configurator are listed in Table 8 below.

Main Functionality	Development State
Extendibility of the products catalogue: Product definition	Implemented
Specification attachment to product	Implemented
Online modification of 3D design: Configure parameters	Implemented for few use cases
Extendibility of the configurator catalogue: Add new configurator	In progress
Import basic 3D designs into 3D Configurator	Implemented for few use cases
Import texture specifications for models in 3D Configurator	Implemented for few use cases
Design extendibility by further elements: electric, electronics, etc.	Planned
Management of the results: export, download of different formats	Partially implemented
Social community: communication, sharing & rating	In progress
Dynamic selection of appropriate UI	Planned
Intuitive UI for end users: Design parameters & constraint setting	Partially implemented

Table 8: Functional Requirements – 3D Product Configurator

To enable the required functionality of 3D Product Configurator, the 4 sub-components are involved:

- Web browser – any internet browser on the customer side
- Frontend application developed in iPRODUCE
- Backend application developed in iPRODUCE
- Amazon Web Services (AWS) for the deployment and temporal hosting of the GDP

Frontend App and Backend App would run in AWS and users interact through the web browser, which interacts with Frontend App. Frontend App interacts with Backend App using REST API, and Backend App handles the API requests, such as fetching the data from the database and saving information into the database. Frontend App also interacts with the underlying 3rd-party cloud technology ShapeDiver over ShapeDiver API, responsible for the online visualization of the transformation scripts (from Grasshopper).

Currently, Frontend App renders the following pages:

- The main homepage listing all product types defined within the GDP
- Creation form for a new product type
- 3D Configurator page listing all product configurators available so far within the GDP
- Creation form for a new configurator of a selected product
- One site for each configurator to view a preset design configuration and to modify this for user needs.

Figure 30 shows a possible product catalog view, currently with fake examples, as we do not have real examples of product definitions. Complete browsing, filtering, and searching functionalities are not implemented in the current prototype but planned for the final release.

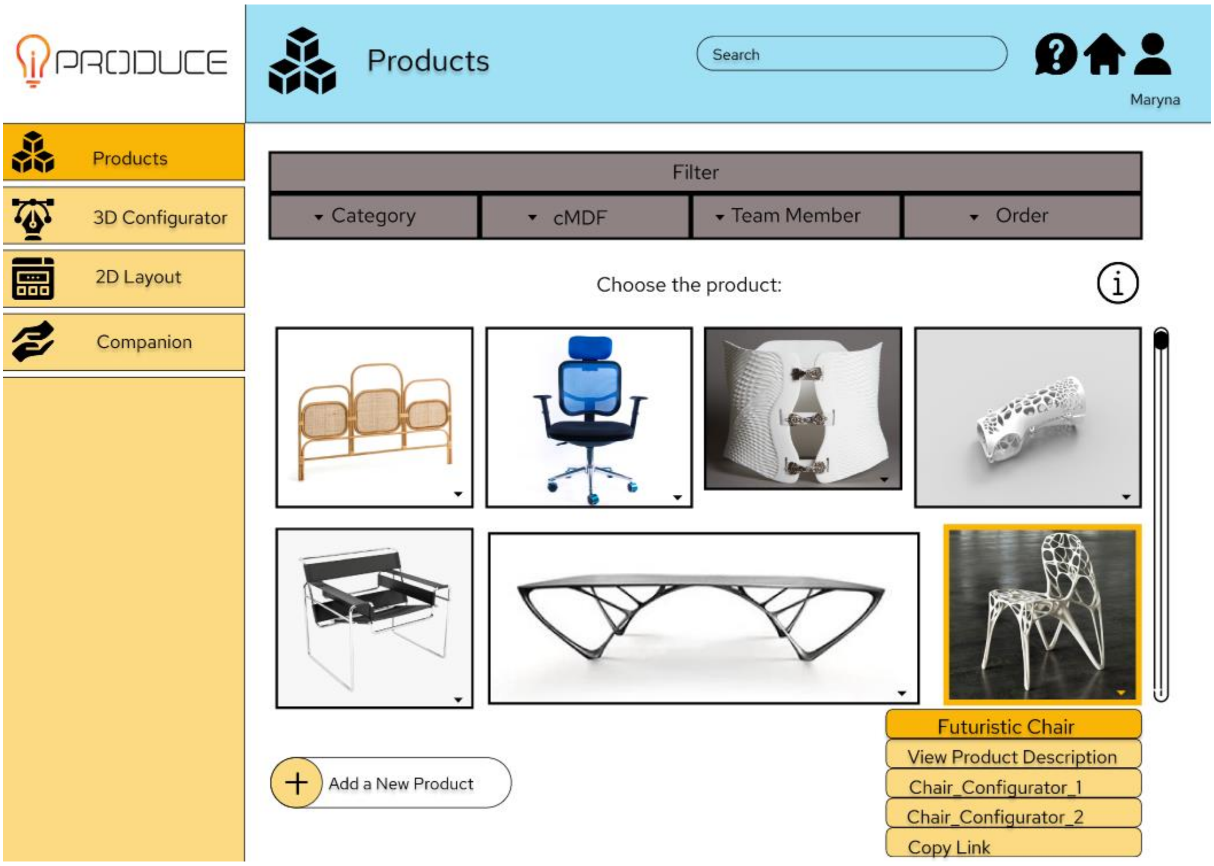

Figure 30: Product catalog

The definition of a new product shall be further discussed with the project partners to discover which meta data will be also important to keep and exchange with other tools. Currently, we assume that the following properties can be useful for product definition, as depicted in Figure 31:

- Product name, a text field.
- Category selected from a list: furniture, medical, robotics, etc. can be used for future browsing.
- cMDF selected from a list: Spanish, Danish, German, Italian, French, Greek; can be used for future browsing.
- Status field currently set and updated manually, because the current usage of this property has not been discussed with project partners to the end.
- Team members, a list of e-mails used as user ID within the iPRODUCE system, currently maintained manually.
- Product description, any short textual description of the aim of the product, main requirements and constraints.
- Product image, a small preview image uploaded by the user or obtained from Marketplace tool from OpIS.
- File attachment, any file with product description, basic design (in 2D or 3D), requirements specification or product resource. The results of the 3D configurator can be also attached here. Several files can be attached if necessary. They will be available for all team members.

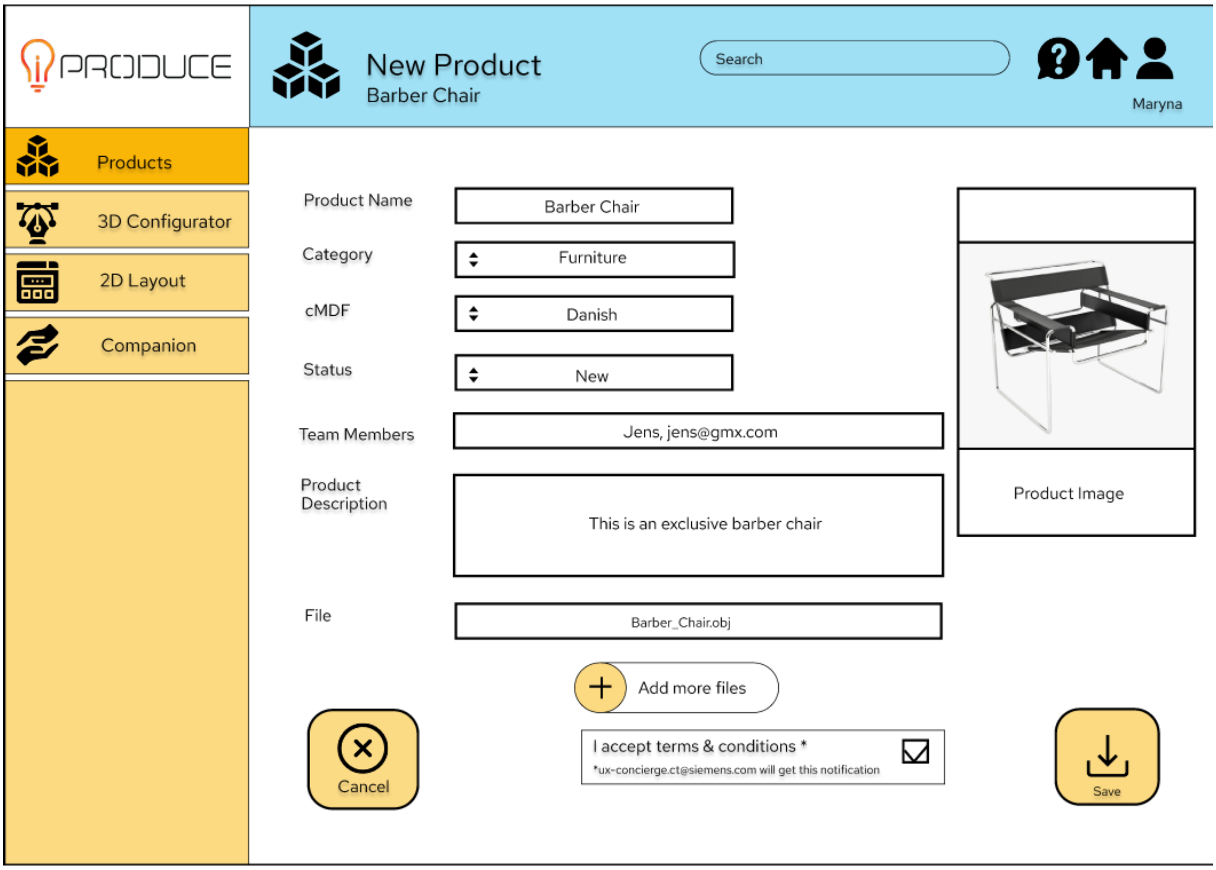

Figure 31: New product definition

Figure 32 below shows a catalog of currently available configurators in the system. Similar to the product catalog, complete browsing, filtering, and searching functionalities are not implemented in the current prototype. The functionality of the current configurators is limited to simple 3D modifications and will be extended during the next months with more interesting features like electronics modelling, calculation of values for optimization purposes, combination with further user inputs, and more output formats.

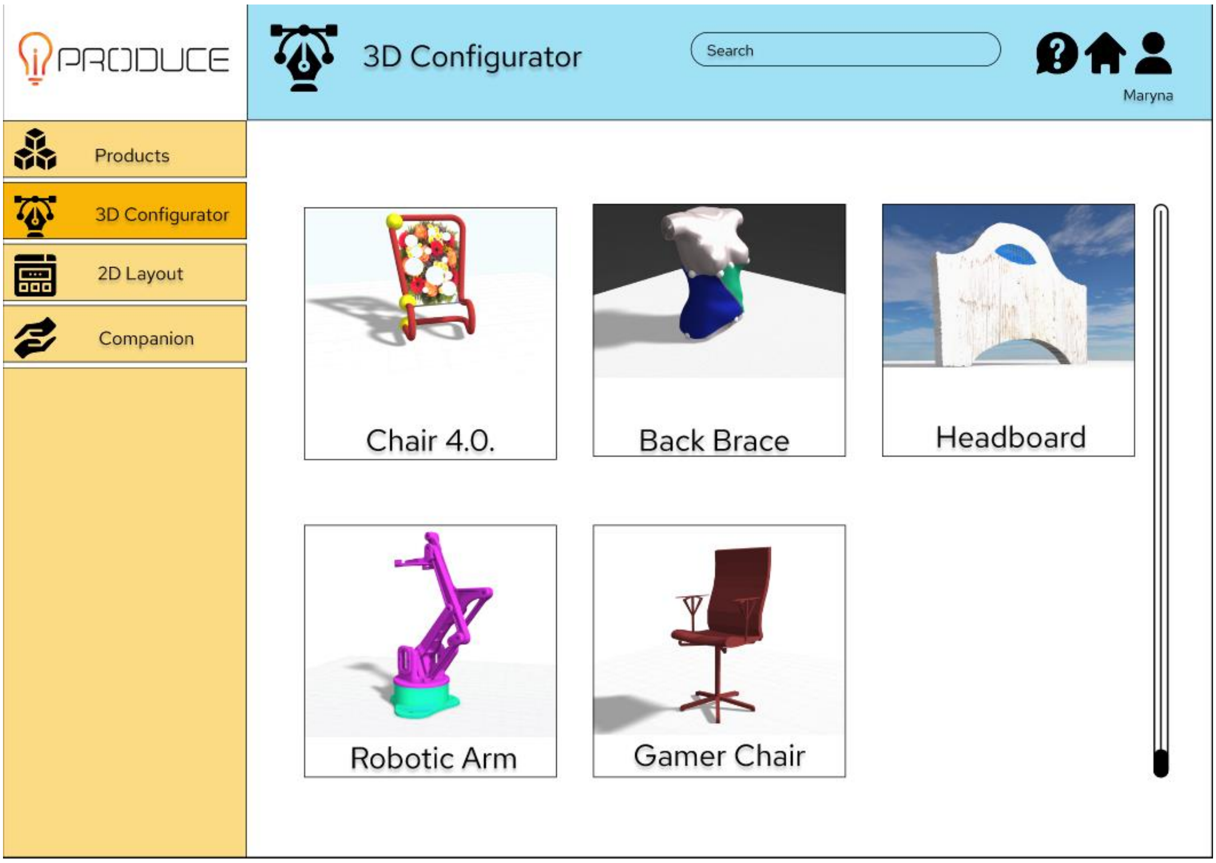

Figure 32: 3D Product Configurators catalog

The implementation of transformation scripts is based on Grasshopper (a plugin of Rhinoceros 3D for parametric modelling) and ShapeDiver with different libraries and add-ons. A configurator based on such a transformation script can accept 3D models (OBJ files) and 2D images (JPEG, BMP, PNG) as end-user inputs, as specified within the transformation script. OBJ format was settled by the iPRODUCE partners for the 3D model exchange among iPRODUCE tools.

The figure below shows an example of a 3D configurator for a parametric chair model with different types of standard UI elements for user interaction without CAD skills. Some operations are very time-consuming and thus, are disabled per default but can be activated by the end user, for example "Calculate Material", "Perforation", because they calculate the final 3D geometry with a complex volume form. This can be disabled for a while to allow the user to perform other modifications in a faster way, viewing only partial results of transformations. Another interaction type that is defined within this transformation script is changing the position of control points directly on the chair model in the pre-view area. New set position of the control points will define a new 3D form of a chair.

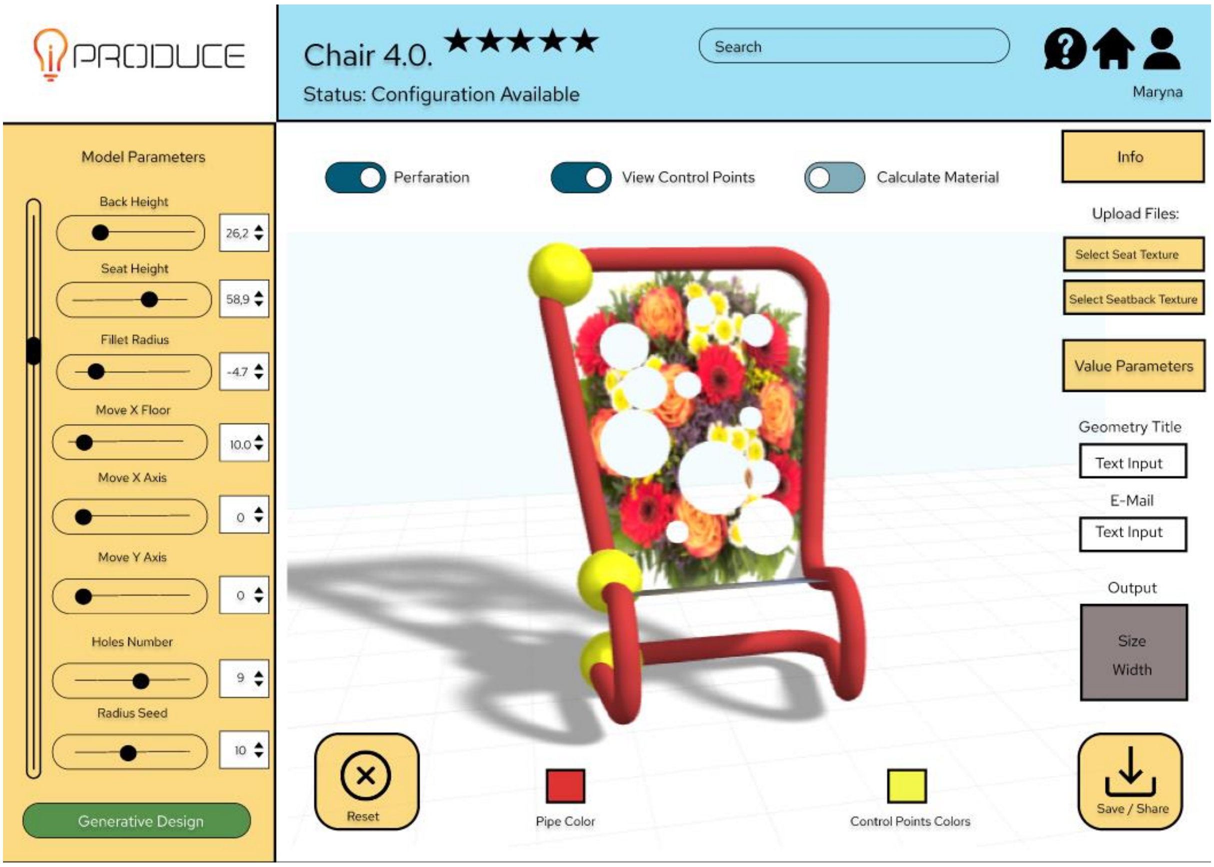

Figure 33: 3D Product Configurator capturing customer wishes through parameter setting

The catalogue of 3D configurators can be extended by a special group of users. A product may have one or several configurators defined by the product team and requested from internal or external sources. A new product configurator can be defined locally by a skilful expert (or expert group), who develops a Grasshopper script and uploads it to the GDP. Because this type of files is very special for the GDP, it is treated differently than the simple upload of attachment to a product.

The figure below shows an example for upload of a GH file (also XML representation can be used instead). After the upload of such files (.gh or .ghx) a complex processing of this file will start. Currently in the first prototype, we perform this processing manually. It is planned to automate this process until the end of the project. All UI elements are generated on the 3D Configurator site automatically, including the interaction over control points, obtaining all necessary the information over ShapeDiver API.

Each configurator will inherit most of the meta properties from the definition of the related product. Only the following properties shall be individual for configurator to be rated individually and to make it possible to contact the responsible person:

- Configurator title (Name)
- Status to be used for approval process within the product team
- Author of the corresponding transformation script
- Description of the configurator intents.

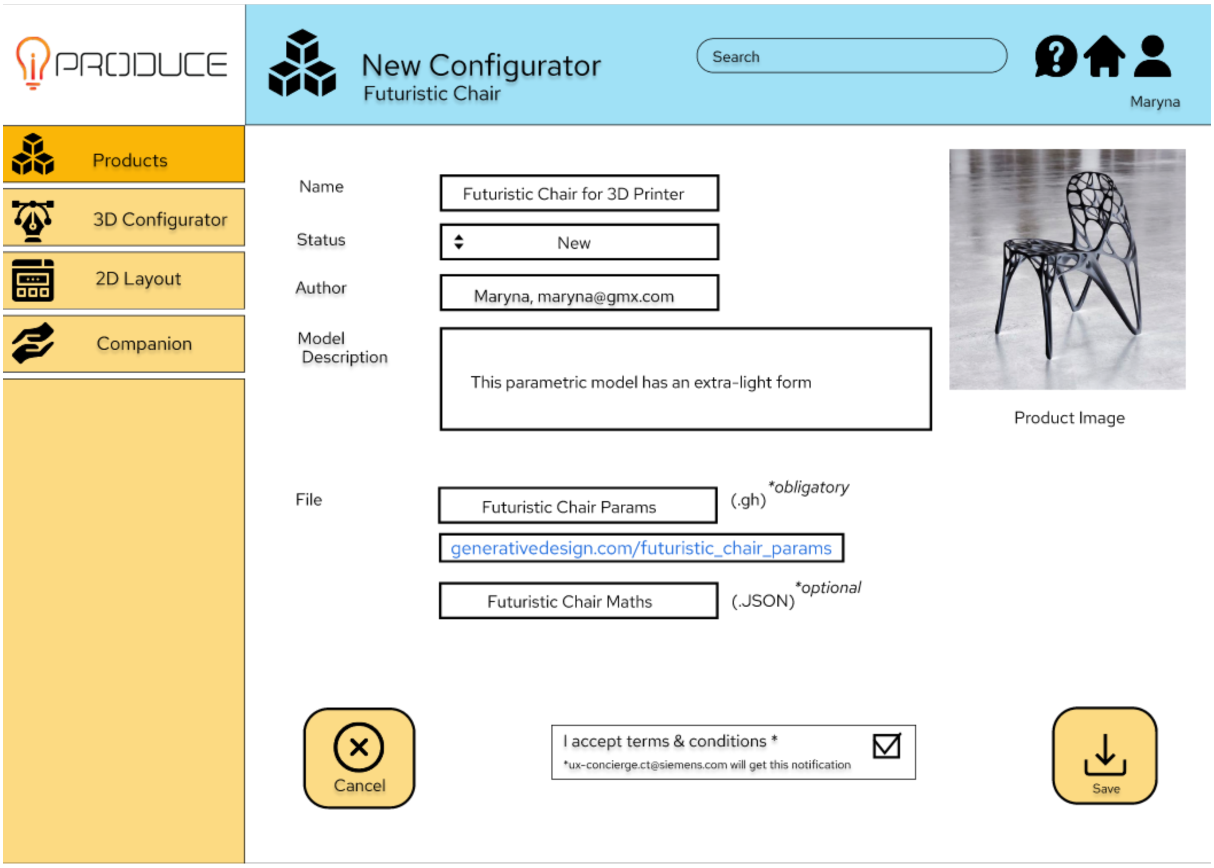

Figure 34: Defining a new configurator for a specific product

Some configurators will be extended with generative studies, where user will define constraints and freedom space as well as optimization target(s), and the system will generate the best solution. This is not currently covered by the first prototype but planned as the next high-prioritized feature to be implemented.

To obtain the results of the user interaction in form of 3D objects with or without textures or, if required, also other formats, the Frontend App will offer save, download, and share functionalities. Users could save any product update within the GDP. They could download files previously attached to the product as well as exported as results of product configuration out of the corresponding 3D Configurator and could share the link to view pages with others.

The resulting user-specific configured product design can be downloaded and forwarded to other tools in different formats:

- OBJ for 3D geometries (also STL or other formats are possible, but currently not requested by iPRODUCE partners)
- PNG & JPEG & BMP for textures or other images
- PDF for textual outputs
- DXF for 2D drawing (e.g., for manufacturing or construction plans; currently not requested by iPRODUCE partners)
- JSON for property definition to be exchanged between tools (also other text formats are possible, but currently not requested by iPRODUCE partners)

Because the status of the component 3D Product Configurator by M18 is a prototype, there are several limitations:

- There will be no authentication (as within the whole GDP). Authentication mechanism from different iPRODUCE partner tool(s) will be used, which is currently under clarification.
- No direct integration between 3D Configurator, 2D Layout Planner, and Mathematical Modelling.
- No rating functionality is implemented yet, only preparation for the future implementation.
- No filtering or searching functionality to browse product and configurator catalogues.
- Generative study with genetic algorithms is not yet integrated into 3D configurators (generating 3D geometries corresponding to the defined optimization targets). Not all current transformations contain necessary definition and calculation of optimization targets. They will be further prepared for generative study in the next months.
- The attachment of the 3D Configurator results to the corresponding product definition is not automated yet.
- Not possible to modify transformation scripts out of Rhinoceros 3D. Slight modifications of XML representations of transformation scripts can be possible, but there is no special UI support for this.
- Processing of the transformation scripts (.gh files) into ready-to-use configurator is not fully automated yet.

4.3. 2D Layout Planner

The SW 2D Layout Planner will allow end customers to define or select a 2D space which shall be filled with other elements (either 2D design elements or furniture for festival facilities or other elements that shall be placed on a limited 2D surface depending on the needs from the use cases). Users will be asked to select the elements to place within this 2D space and set constraints on this placement.

Producers / involved MakerSpaces may add product types either as 2D surfaces (splints, headboards, other kinds of boards) or as elements to be located within a bounded 2D space (e.g., electronic equipment, seating capability for festival facilities, etc.) and extend their description with constraints on their usage in future versions.

A genetic algorithm will calculate the best-fitting layout. A 2D layout calculated for one end customer can be stored in the catalogue for further production/arrangement or be offered to other users.

The main functional features of the 2D Layout Planner are listed in Table 9.

Main Functionality	Development State
Selection of basic 2D layout	partially implemented
Definition of user-specific 2D space	partially implemented
Extendibility of the products catalogue	partially implemented
Parametric design, online modification of design parameters	partially implemented
Constraint’s specification in an intuitive way	planned
Connections to community (for rating)	planned

Table 9: Functional Requirements – 2D Layout Planner

The 2D Layout Planner is currently split into a FrontEnd (FE) and BackEnd (BE) part. The FE implementation is browser-based and manually tested in Google Chrome; other browser engines are most likely working as well.

The main UI consists of an React component which is based on an open-source implementation. The editor enables the user to draw his/her own plan with a limited set of tools. The current prototype, as it will be delivered to the project partners at M18, will include one kind of each of the following items:

- Wall
- Window
- Door
- Desk

Figure 35 Available editor tools in M18 deliverable

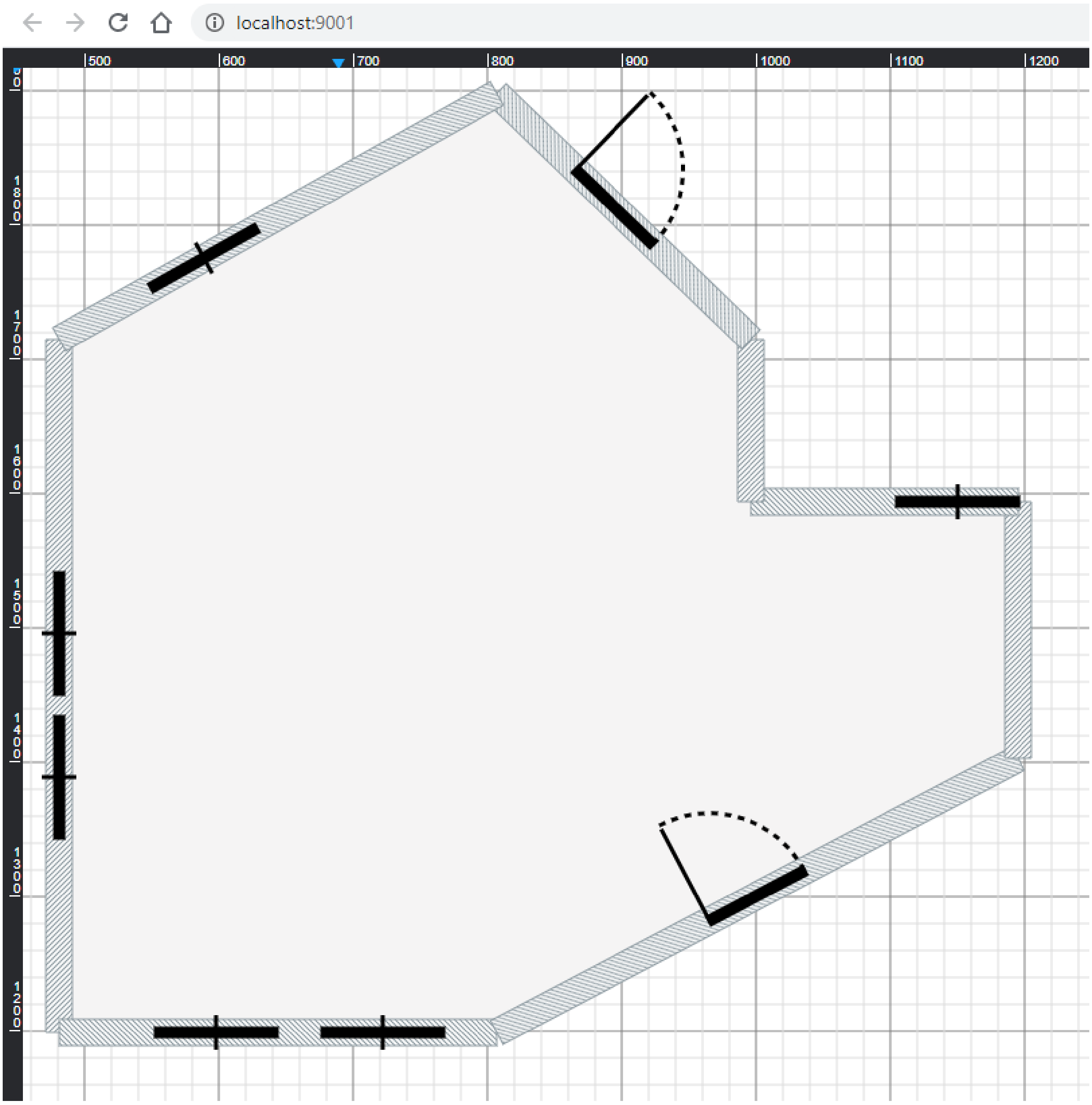

Figure 36 Web-based editor for floor planning

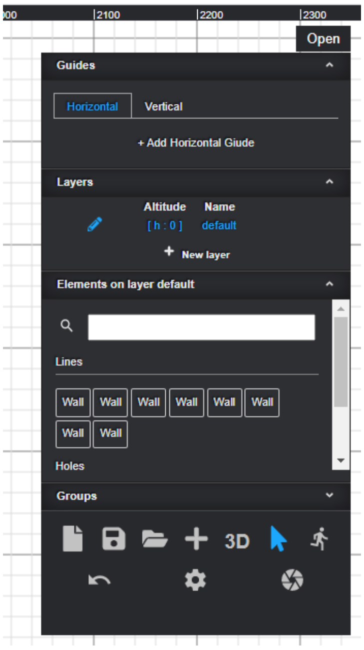

Figure 37 Editor's side panel, where users can select tools and context information

Once a layout is finished, like outlined in Figure 36, the user can decide, how many items (Desks) shall be placed in a specific area, which rules shall be applied in which priority (order) and what “exit criteria” shall end the optimization and the “hyperparameters” for the genetic algorithm, e.g., mutation rate and selection process. The following rules are implemented in the current (M18) deliverable:

- collision of items amongst each other
- collision of items with walls / doors / windows
- fitness value – how good item fits to the given optimization problem (number of desired desks in an area)

Especially for the last rule, an artificial “ghost desk” has been introduced in case there are no other placeable elements to test with. In that context, a ghost desk is the same thing as a normal desk, but it is simply counted as the wrong item chosen by randomization of the genetic algorithm.

Adding of custom rules by the user is not in scope for M18.

When the user has finished setting the exit criteria it is usually a combination (logical OR) of three targets:

- TimeEvolvingTermination with a given specified time – that means optimization is stopped if this time is exceeded.
- FitnessStagnationTermination with a given number of generations – that means optimization is stopped if the fitness did not change for a given number of generations.
- FitnessThresholdTermination with a given percentage threshold – that means optimization is stopped if the desired threshold is reached – 1 is for 100%. 100% does not necessarily mean a perfect solution is found but rather one which fulfils all required rules.

If the user has set all parameters, he/she can send the room geometry including the optimization target (including rules) to the backend, and the optimization process is started immediately.

The Frontend queues the optimization task to the backend, which can then cyclically poll for the status of execution. The UI gives visual feedback on the amount of passed time, generations, current fitness value and best fitness value – when optimization is done, the best 10 candidates are displayed as table entries; clicking them will update the editor and display the chosen solution visually.

The Backend implementation is DotNet-Core-based (Core5.0). It exposes a REST endpoint; in development a SWAGGER endpoint definition displayed to the developer. The genetic algorithm implementation is based on GeneticSharp for reasons of performance, platform independence and freedom of dependence towards other frameworks.

An endpoint specification will be included in a later state of this document, as the data definition is internal and currently experiencing heavy updates.

5. Next Steps in the Platform Development

The main efforts in the next period of the development will be dedicated to the integration of the individual components within the GDP and the integration of the GDP into the iPRODUCE's OpIS platform.

Also, all three components of the GDP will be improved and their features extended.

Currently, only few of the project's use cases are covered by the examples of 3D Configurators. Although more use cases shall be covered by providing new configurators (in form of transformation scripts .gh or .ghx) from project partners, SAG plans to develop further configurators for other use cases and to provide their defining scripts as samples to our project partners.

5.1. Integration Activities

In the next months, we plan to integrate 2D Layout Planner and Math Modelling (Companion) into the current GDP containing 3D Product Configurator by linking the reserved buttons within the GDP interface to URLs of the further components because they will also be deployed on AWS and accessible online. The results generated within the three components will be stored in the definition of the corresponding products and thus, can be exchanged among tools, e.g., by references.

Math Modelling will generate JSON results with structured spatial information out of natural language conversations, which will be used as input for 3D configurators to define default or pre-set parameter values.

After the implementation of the common repository within OpIS, the results of the GDP components (OBJ, JPEG, PDF, JSON files, and possibly other formats if requested) will be stored within this repository, keeping the references within product definitions. Any other OpIS tools will access the resulting files on this common repository; properties of the GDP results will be stored in JSON format. The exact scheme of this JSON format will be defined in the following period in cooperation with other project partner (in WP4).

5.2. Feature Extension for the Main Components

For the GDP, we plan to continue to develop:

- Social community support: allow product team members and end users to rate product designs and 3D configurators. Also, results sharing will be improved further.
- Improved browsing through products, configurators, and results: implementing filtering and basic searching functionality.

Component 3D Product Configurator will improve the following functionalities:

- Automation of the transformation scripts processing (.gh & .ghx files) into ready-to-use configurator. This depends on the corresponding feature of the 3rd-party technology ShapeDiver, which will be available in the next months and accessible via API.
- Genetic algorithms will be integrated into 3D configurators to allow running generative study and generate 3D geometries corresponding to the defined optimization targets. The current preparation for this extension for 3D configurator is shown on Figure 38.

- The current transformations will be extended to address settings necessary for generative study: definition and calculation of optimization target(s), selection of relevant parameters and setting constraints on them.

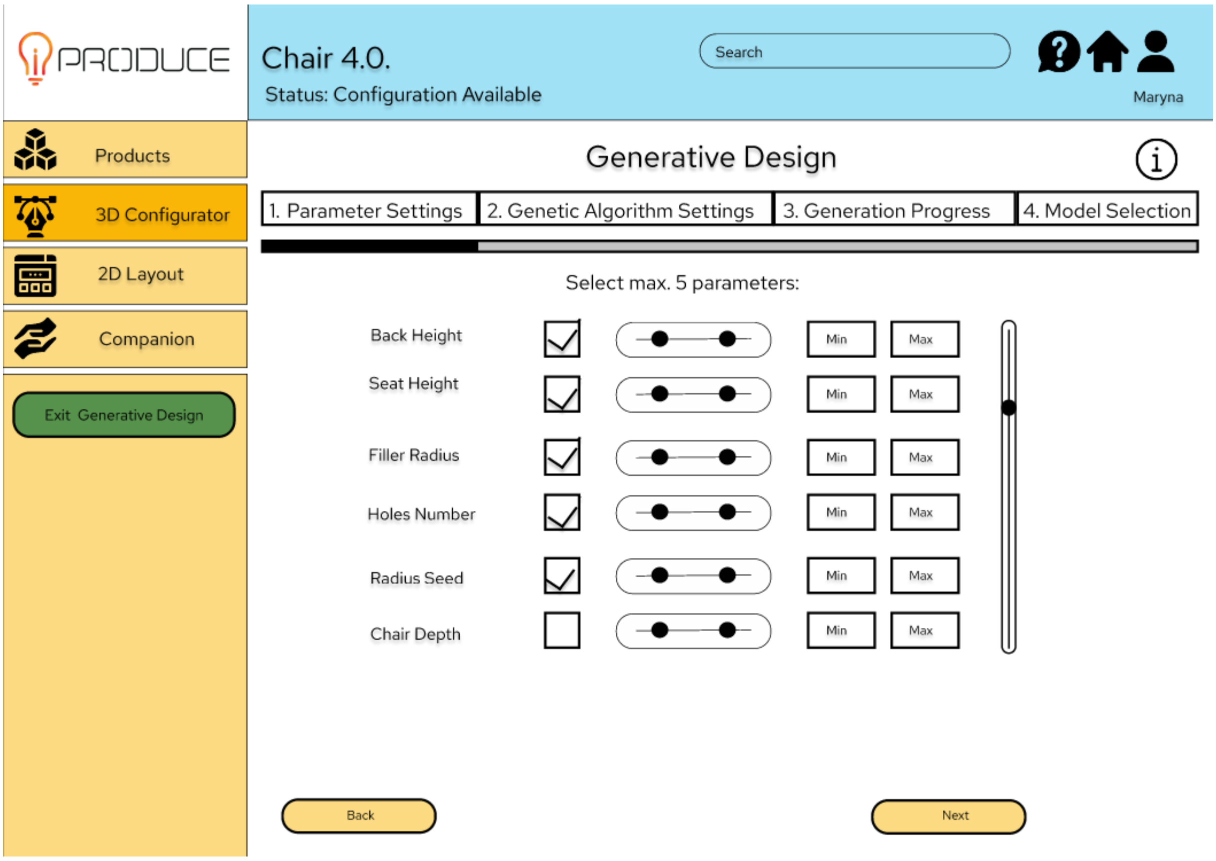

Figure 38: Generative Design: defining constraints (vs. level of freedom)

A fitness function is a part of generative study settings. It defines the optimization target: it is a numeric value that needs to be maximized or minimized when a genetic algorithm runs. This numeric value shall be first defined within a transformation script as an output. The script shall contain calculation of this value, dependent on the 3D geometry resulting from the parameter values settings and some other factors. 3D Configurator then will provide a UI to set up parameter constraints and settings for the fitness function. The genetic algorithms will result in one or several optimal model configurations (defined through parameter value settings). The end user shall decide about the resulting model and download or store it within the GDP.

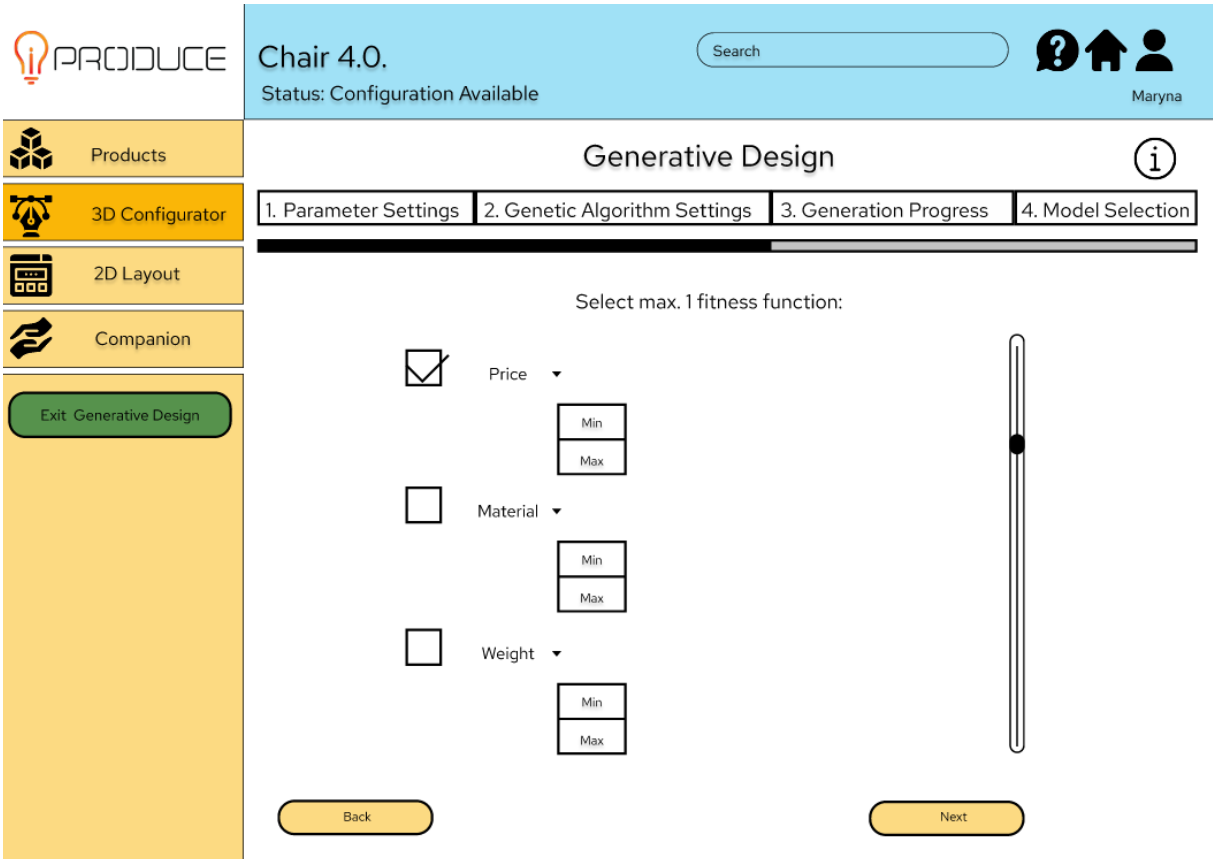

Figure 39: Generative Design: fitness function

5.3. Extension of the iPRODUCE Use Cases Coverage

Currently, few use cases are demonstrated with simple 3D Configurators: a chair generation (for Danish UC), gamer chair (for Spanish UC), a robot arm (for Italian UC), a back brace (for Greek UC), and a headboard (for Spanish UC). For 2D Layout Planner, currently only Danish UC for festival facilities planning is considered. After the initial evaluation for the mentioned use cases, we will decide together with our project partners which further use cases shall be addressed by the GDP components. Mathematical Modelling is also planned to be evaluated on some concrete iPRODUCE use cases.

We plan to provide more configurators that can transform input 3D models (OBJ files) independent on the specific product. Thus, one or several so-called general transformations can be applied to different products, for example to help during 3D printing.

Currently, the covered use cases do not contain modelling of electronics or allocation of additional (3D or 2D) elements. The use cases are planned to be extended with IoT and/or gamification concepts (examples: smart headboard, back brace, gamer chair, robot).

6. Conclusion

This technical report describes the first prototype of the tool “Generative Design Platform as Social Community”, developed by SAG by M18.

The Generative Design Platform (GDP) is being developed based on the needs from the use cases of iPRODUCE. This report describes the activities performed during the period of M7-M18 within Task 5.2. It includes the specification of the GDP architecture designed, aligned to the overall OpIS architecture from [3]. The report describes the main functionalities of three components of the GDP, developed within the current prototype, and plans for their further development in the next period.

The tool GDP will contribute to the project objectives 3 and 4 as it is planned to integrate the GDP into the iPRODUCE framework and platform OpIS. As explained in the introduction to this report, the GDP will also contribute to several project KPIs. It will provide functionality to support co-creation & co-design and an appropriate and intuitive interface for end users to allow interaction for different user groups, in particular for consumers.

One major limitation of the current development state is that the GDP’s components are not yet integrated with each other and with other iPRODUCE tools constituting the OpIS platform. This limitation will be addressed in the final version of the GDP application.

References

[1] iPRODUCE D5.1 Assistive and Collaborative Designing Methods and Tools, FIT, June 2021.
[2] iPRODUCE D2.5 Definition of iPRODUCE Demonstration Framework, MAT, December 2020.
[3] iPRODUCE D4.1 OpIS Architecture & Design for Social Manufacturing I, CERTH, November 2020.
[4] Rhinoceros 3D, CAD tool, https://www.rhino3d.com/
[5] Grasshopper 3D, a visual scripting plugin within [4] for parametric design, https://www.grasshopper3d.com/
[6] Siemens NX, CAD tool, Siemens AG, https://www.plm.automation.siemens.com/global/en/products/nx/nx-for-design.html
[7] Solid Edge, CAD tool, Siemens AG, https://solidedge.siemens.com/en/
[8] Fusion 360, Autodesk, CAD tool, https://www.autodesk.com/products/fusion-360/
[9] The Flow Of Generative Design, Cadabra Studio: UI/UX Design Agency, https://cadabrastudio.medium.com/the-flow-of-generative-design-631d69d4343e
[10] Thingiverse, Online community for maker activities, https://www.thingiverse.com/
[11] Architectural Optimisation and Evolutionary Systems with Genetic Algorithms, http://www.interactivearchitecture.org/architectural-optimisation-and-evolutionary-systems-with-genetic-algorithms.html
[12] Open SCAD, The Programmers Solid 3D CAD Modeller (used in [10]), https://openscad.org/
[13] AWS, Amazon Web Services, https://aws.amazon.com/
[14] ShapeDiver, Cloud service for Grasshopper [5] visualization, https://www.shapediver.com/
[15] Rhino 3D Compute Server, Cloud service and REST API for Grasshopper [5] https://developer.rhino3d.com/guides/#compute
[16] Ultimaker, 3D printer, https://ultimaker.com/ , different models.
[17] Arduino, open source HW platform, https://www.arduino.cc/
[18] Figma, the collaborative interface design tool, https://www.figma.com/
[19] MQTT: The Standard for IoT Messaging, https://mqtt.org/
[20] Generative Pre-trained Transformer 3 (GPT-3), Open AI API, https://openai.com/blog/gpt-3-apps/
[21] ThreeJS editor, 3D editor for Web and three.js library, https://threejs.org/editor/
[22] Genetics, an open source evolutionary algorithm, https://www.npmjs.com/package/genetics-js
[23] Amazon Simple Storage Service (AWS S3), https://aws.amazon.com/s3/?nc2=h_ql_prod_fs_s3
[24] Microsoft Container Registry, http://mcr.microsoft.com/mssql/server
[25] MinIO Object Storage for Hybrid Cloud, https://min.io/
[26] OpenAPI / Swagger specification, https://swagger.io/docs/specification/about/
[27] *J. Devlin*, *M.-W. Chang*, *K. Lee*, *K. Toutanova*, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019, accessible via https://arxiv.org/abs/1810.04805
[28] GeneticSharp, https://github.com/giacomelli/GeneticSharp
[29] ISO/IEC 25010, Software and Data Quality, 2021, available at https://iso25000.com/index.php/en/iso-25000-standards/iso-25010
[30] ISO/IEC 9126-1:2001, Software engineering – Product quality – Part 1: Quality model, 2001, available at https://www.iso.org/standard/22749.html

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 870037.