TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DIPLOMARBEIT

Augmented Reality as Onsite Communication Tool

ausgeführt zum Zwecke der Erlangung des akademischen Grades

eines Diplom-Ingenieurs / Diplom-Ingenieurin

unter Leitung von

Ass. Prof. Dipl-Ing. Dr.techn. Peter Ferschin

E259.1

Institut für Architekturwissenschaften

Forschungsbereich Digitale Architektur und Raumplanung

eingereicht an der Technischen Universität Wien

Fakultät für Architektur und Raumplanung

von

Jakob Lugmayr, BSc

01226488

Wien, am 30.5.2023 ________________________________

Kurzfassung

Auf Baustellen ist die derzeit vorherrschende Praxis zweidimensionale Pläne zur Errichtung von Gebäuden zu verwendet, obwohl heutzutage ein großer Teil dieser Pläne aus dreidimensionalen Gebäudemodellen stammt. Durch diese Abstraktion entstehen häufig Fehler im gebauten Abbild und Intentionen der Planer werden falsch interpretiert. Dazu kommt das nicht aktuelle Planstände im Umlauf sein können, was zu weiteren Fehlern führen kann.

Diese Probleme führen zu Mehrkosten, Verzögerungen auf der Baustelle oder das Gebäude entspricht schlicht nicht den geplanten Vorstellungen. Dies führt zu Streit zwischen den Parteien und oftmals langwierigen Rechtsstreitigkeiten.

Moderne Planung in Kombination mit Augmented Reality (AR) kann die Möglichkeit bieten Baufehler, die durch schlechte Kommunikation entstehen, zu reduzieren und somit Mehrkosten zu vermeiden, sowie auch die Qualität zu steigern.

Das Ziel dieser Arbeit ist es Möglichkeiten und Arbeitsprozesse aufzuzeigen, welche helfen können sich von den alten Praktiken abzuwenden, um moderne Technik für moderne Planung zu nützen. Zuerst soll ein Einblick der Möglichkeiten die Augmented Reality derzeit bietet geschaffen und die technischen und historischen Hintergründe umrissen werden. Ebenso wird ein Überblick der derzeitigen Planungsmethoden von Hochbauten, insbesondere Building Information Modeling (BIM), gegeben und gezeigt wie ein aus dieser Praxis resultierendes Gebäudemodel für eine neue Art der Plandarstellung mit Hilfe von Augmented Reality auf Baustellen genützt werden kann.

Um die Machbarkeit der Einbindung von Augmented Reality in den Bauprozess zu bewerten, wird im praktischen Teil der Arbeit ein Projekt in verschiedenen Maßstäben augmentiert und die Arbeitsschritte von BIM-Model zu gebauter Realität überprüft und dokumentiert.

Abstract

Two dimensional plans, as communication tool, are the state of the art on building sites for a long time now, although today most of these plans have a three-dimensional building model as their origin. This compression of Information often leads to construction errors and the planners' intentions may often be misinterpreted. Furthermore, outdated plan statuses may be on site, which can lead to more errors

These problems lead to additional costs, delays of the building process or the building just does not meet the planned expectations. This often leads to long lawsuits and fights between the parties.

Modern planning methods in combination with Augmented Reality could be a tool to decrease these errors, lower the risk of additional costs and improve quality.

The aim of this work is to show the possibilities and workflows, which can help to leave these old methods behind and establish modern techniques for modern plan presentation. First an insight of Augmented Reality and the possibilities of its usage will be shown. Also, the historical and technical background will be explained. Furthermore, an overview of planning methods in building construction, especially on Building Information Modeling (BIM), will be given and how the resulting 3D-buildingmodel can be used in combination with Augmented Reality to establish a new method of plan presentation on site.

To evaluate the feasibility of the inclusion of Augmented Reality in the building process, a project will be augmented in different scales and the workflow from BIM-Model to built reality will be tested and documented.

Index

Introduction

For a long time, people are building and try to improve their buildings. Progress was seen in building technics and building materials, so that the buildings could get more complex and efficient over time. The bigger the structures got, the more material had to be moved and other inventions like the steam engine and excavators where key to further innovation in the building process. [12]

When computers got affordable in the 1980s, information technology got part of the building industry and new workflows developed and work got a lot more efficient. This started with more easy communication over email, where other participants got building plans or important information in a few seconds. When talking about building plans, the evolution of "Computer Aided Design" (CAD) helped to get more precise and organized. It also allows to make complex simulations. [12]

Another big innovation was the mobile phone, which allows to communicate in real time and from person to person over a big distance. This is very important in construction, when urgent information must be delivered to prevent errors. This was a big change to the work on the construction site. [12]

Also, other mobile devices like tablet computers have become part on building sites. They are used as monitoring tools or even as tools for construction surveying in civil engineering. In combination with powerful GPS sensors centimeter-accurate measurement is possible.[12]

Although these innovations and digitalization, the building industry still has some of the old problems like delays, additional costs, bad communication between parties and poor quality. Furthermore, new requirements like sustainability, energy efficiency and lifecycle-management. To solve these problems new workflows and processes will have to be developed. [12]

One key component for developing these new workflows and processes was the further development of CAD. It led to a more complex approach, where not only 3-dimensional geometries and their 2-dimensional images are produced, but information is woven into the geometry. "Building Information Modeling" (BIM) is a tool, which can be harnessed in multiple ways. The goal is to bring all the available information about a project (e.g. cost, performance characteristics of used materials,..) together into one digital model. This model can be used for simulations, cost estimations, lifecycle-management and much more. A whole new branch of possible workflows and processes can be accessed and developed to solve the problems mentioned before.

Another technology, which is rapidly developing, is “Augmented Reality” (AR). AR is a variation of a Virtual Reality. But instead of replacing the real world with a virtual one, AR adds virtual objects to the real world. So, the user is still navigating in the real world, but gets additional information. The idea goes back into the 1960s, where Ivan Sutherland developed his head-mounted three-dimensional display. Since then, technology developed vastly and AR-systems got more accessible in form of modern mobile phones. This led to a greater number of AR-developers and more innovation in this field.

The following thesis will describe the development of CAD, methods of communication on site and current processes and developments in “Architecture and Engineering Construction” (AEC). New methods of planning and communicating, the technical and historical background of AR and BIM.

In the project study the combination of AR and BIM, as new communication tool, will be tested in different scales. The used software and hardware will be described and a workflow from BIM-model to a step-by-step-augmentation of it will be developed, tested, and documented.

1.0 Status quo

In the following, an overview about the different topics will be shown. The development of “Computer Aided Architectural Design” (CAAD), communication on site and processes in AEC will be described.

1.1 Drawing Generation and History of CAAD

In this section the development of drawing generation is shown. From hand-drawings to “Computer Aided Design” (CAD) and finally to the newest generation, BIM. In each section it is discussed how drawings are/were generated. Also, a brief history of the development of modern CAD and BIM is shown.

1.1.1 Milestones in CAD [11]

A milestone in CAD was the development of Ivan Sutherland’s Sketchpad, where the user interacted with the software through a light pen on a CRT monitor.

Figure 1: Ivan Sutherland and his Sketchpad (Tornincasa & Monaco, 2010)

The first generation of CAD systems were internally developed by American manufacturers like General Motors with “Design Augmented by Computer” (DAC), McDonnel-Douglas with “CADD”, Ford with the “Product Design Graphic System” (PDGS) and Lockheed with “Computer Augmented Design and Manufacturing” (CADAM) in the mid-1960s. These where typically 2D drafting applications.

Figure 2: DAC-1 by GM and IBM (Tornincasa & Monaco, 2010)

In the 1970s the commercial use of CAD started. 1975 the first successful system was the Unigraphics System, used for 2D modeling and drafting, produced by United Computing. In the same year Avion Marcel Dassault started the development of a 3D CAD named “Conception Assistée Tridimensionnelle Interactive” (CATI). In 1979 Boeing, General Electric and NIST defined a new 3D data exchange format called “Initial Graphics Exchange Specification” (IGES).

In 1981 “UniSolids”, the first solid modeling system, was introduced. The same time, the first commercial version of CATI was released.

In 1983 Autodesk which was founded a year before released an AutoCAD, a CAD program for a price of $1000 running on a PC, which was a breakthrough at that time.

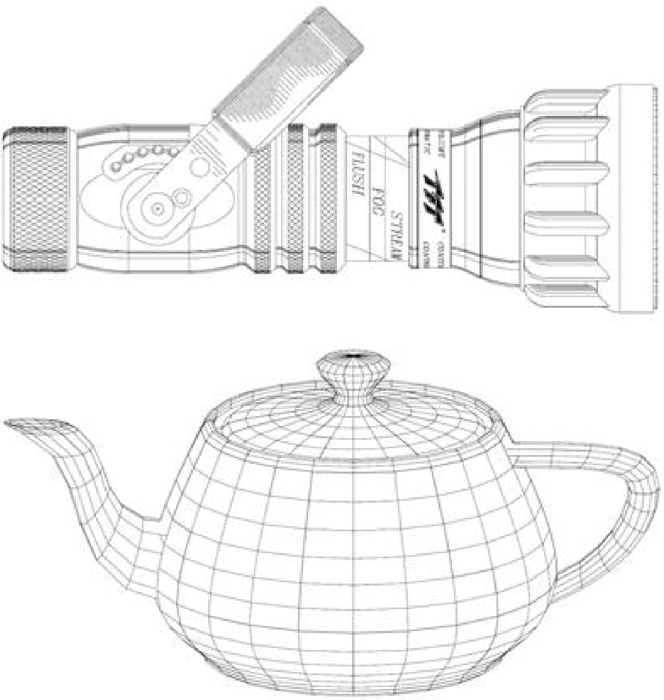


Figure 3: AutoCad 2.6, with the first 3D wireframe model (Tornincasa & Monaco, 2010)

In 1984 Apple presented the first Mac 128 and a year after its release, “MiniCAD” was published and became the bestselling CAD for Mac. Also, Graphisoft released “Radar CH” on Mac, which was renamed to “ARCHICAD 1.0”. Developing for Macs continued but the big break throughs in CAD were made on the UNIX Workstations, because Macs had not a high enough performance in the mid-1980s.

In 1987 a big revolution in the CAD industry accrued with the release of “Pro/Engineer”, the first parametric and associative solid modeler, which functioned on the UNIX Workstation and was created by Parametric Technology Corporation.

In 1989 Unigraphics retired UniSolids and released a new application named “UG/Solids” which was based on “Parasolid”, a modeling core for 3D-CAD-Systems. Also, ACIS, another modeling core, was released and licensed by HP.

In the early 1990s nearly all CAD software ran on UNIX Workstations and no longer on minicomputer and mainframe-computers. The market was dominated by IBM/Dassault Systems, EDS-Unigraphics, Parametric Technology and SDRC.

In 1994 Microsoft released its first 32-bit operating system and Intel released its first Pentium Pro processor. ACIS and Parasolid were quickly available on Windows NT, and so the development of CAD for desktop pcs was kickstarted.

1995 the first “SolidWorks” was released for desktop pc and so a new mid-price 3D CAD category was born. In 1996 Intergraph released an CAD very similar to SolidWorks and based on ACIS named “SolidEdge”. Autodesk, whose “AutoCad” lost market share, released “Mechanical Desktop”, which quickly became the number one selling CAD in the world.

1.1.2 Building Information Modeling (BIM) [2]

Even though the whole geometric layout of the building and all its assets is covered in a BIM-model and additional information like properties, which are discussed later, is given, drawings will be still required for a longer period. Whether presented digitally or on paper, the current work-culture and bureaucracy will not allow a fast change away from drawings.

With BIM every instance of a building object, like shape, properties, and placement in the BIM-model, are nonredundant. Based on that, drawings, reports, and datasets can be extracted and are consistent as long they are taken from the same version of the BIM-model. This alone prevents a lot of errors, because updates no longer must be made in each drawing manually. Current BIM design tool capabilities come close to automated drawing extraction.

As an example, the location of a section is automatically recorded by a section line symbol as and is cross referred in all relevant plans or elevations. Its position can be moved if needed and the associated drawing will change accordingly. The drawing is generated automatically according to the section line and if needed more detail can be added to the drawing. This happens with 2D elements. These elements can inherit some automation, but most of the time if changes are made in the 3D model, the 2D information which was added must be updated manually. The detail which must be added in 2D is dependent on the quality of the 3D model.

For production of drawings, each plan, section, or elevation is composed by the above rules as a combination of 3D sections and aligned 2D drawn sections. They are then placed on layouts with borders and titles. 3D model generated drawings have gone through a series of refinements to make it easy and efficient to produce them.

- A low level of 3D model drawing production is that drawings are produced of orthographic sections cut from a 3D model. The user extracts the geometry and then manually edits line formats, adds dimensions, details, and annotations. As long the section exists, these added details are associative to this section. On this level the drawing generation may be done in the BIM tool itself or in an external drafting system.
- An improvement is the definition of drawing templates associated with elements for a type of projection like sections, elevations, plans. These templates generate dimensions formatting of the element, assigns line weights and generates annotations from predefined attributes like window dimensions, etc. This change speeds up the drawing setup and productivity. Predefined layout defaults can be overwritten if needed and custom annotations can be added. 2D drawing edits

cannot be made in the model projections, but must be made in the 3D model itself.

- The current top-level BIM-systems support bidirectional editing between models and drawings. Changes to model annotations are the same as above, but changes made to the model itself can be made in all drawings which are connected directly to the model itself. Bidirectional views, strong template generation and real-time updates in all views further reduce time and effort needed in drawing generation.

The current goal of drawing production is a high degree of automation. This can only happen if all parties working on a project, adapt to BIM technology and then drawings will not be required that much, because it will be worked directly with the BIM-model. Full automation and interoperability are unlikely to be achieved because drawings can be generated in various of ways depending on drawing conventions and designs.

Current BIM tools allow the designers to choose which degree of detail the 3D model will achieve on his own and what must be edited in 2D. The benefits of BIM like: automated cost estimation, clash detection and other data exchange, are lost with these added 2D elements. As it can be argued that complete 3D object modeling is not wanted in every situation, sophisticated BIM-projects like Medina Airport, Dublin New Children's Hospital and other examples showed that steps have been taken towards it. (Further examples in Sacks et al., 2018)

1.2 Communication on Site

The communication of a design is a key element to make sure it will be built as planned. Often intentions will be misunderstood and critical errors are made during the construction. So, information must be delivered precise and coherent. Historically there have been a lot of different methods of communicating design. In the following a few different methods will be described and evaluated.

1.2.1 Architectural Drawings [13]

In architecture, drawings are one of the first tools of communication for building complex buildings. The modern architectural drawings can be handed down to the 13th century. Before that a continuous custom cannot be proven. But there are scratch drawings from the Greek antiquity, which demonstrate problems which handle the column order or the entasis and kuvatur. Floorplans and elevations from Egyptian times on papyrus, the “forma ubris” from the roman antiquity, which shows the city plan and some buildings, and more ancient drawings were found.

Figure 4: Fragment of Forma Ubris Saverina on marble, 300 (Philipp, 2020)

After this there is a big hole in history and the next known architectural drawing is the "Sankt Galler Klosterplan" from the early 9th century. This plan builds a bridge between antiquity and Middle Ages. Although the plan only shows the configuration of rooms and is not viable to build after it, as the project Campus Galli, which is a rebuild of this plan, shows. Some distances between buildings are too close, so that the transportation of new building material is impossible.

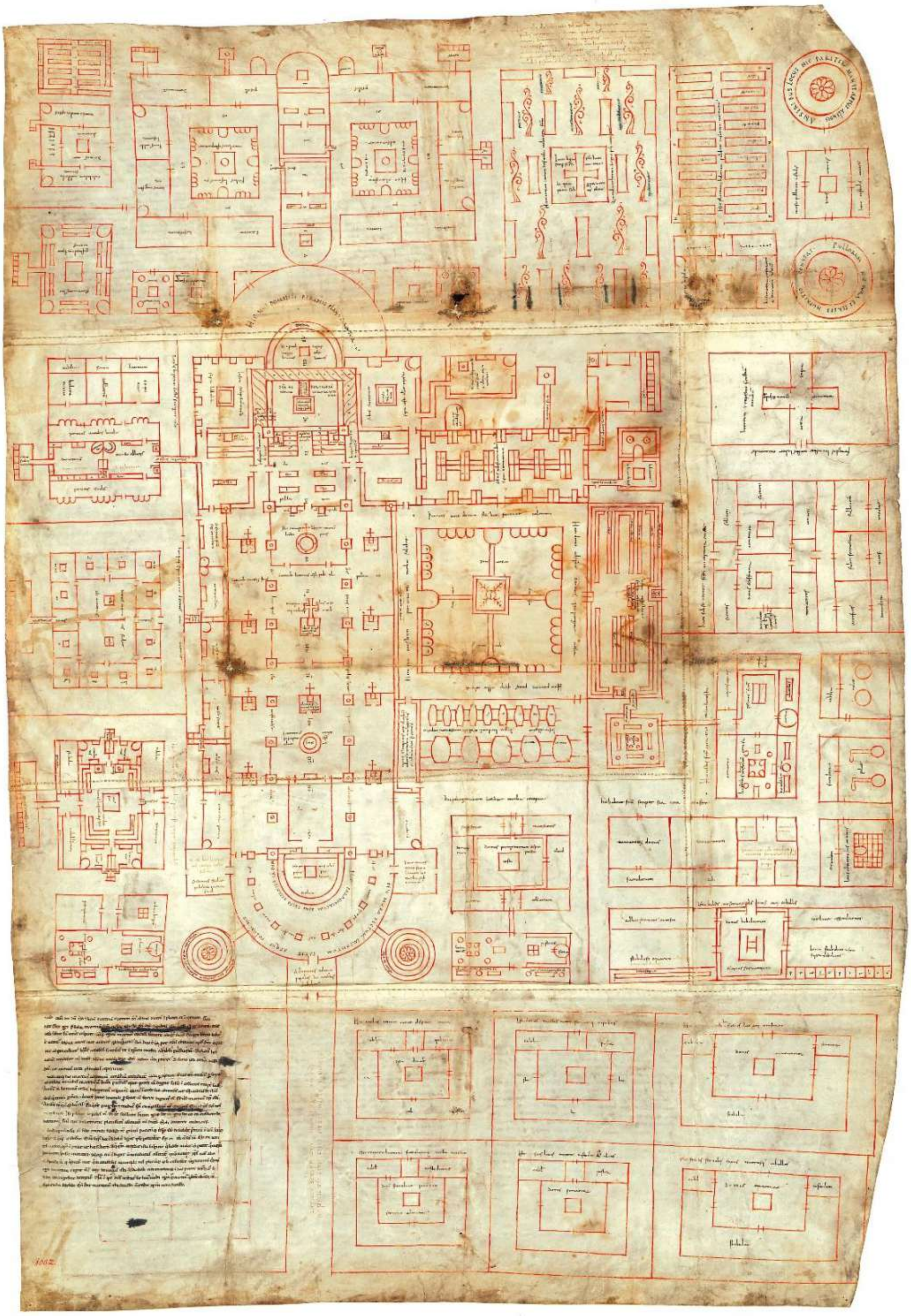


Figure 5: Plan of the monastery of St. Gallen, ink on parchment, 800 (Philipp, 2020)

First modern architectural drawings were found in the sketchbook from architect Villard de Honnecour from 1230. Villard documented the French cathedrals in his sketchbook. The drawings consist of floorplans, elevations, sections, some kind of perspectives and notes.

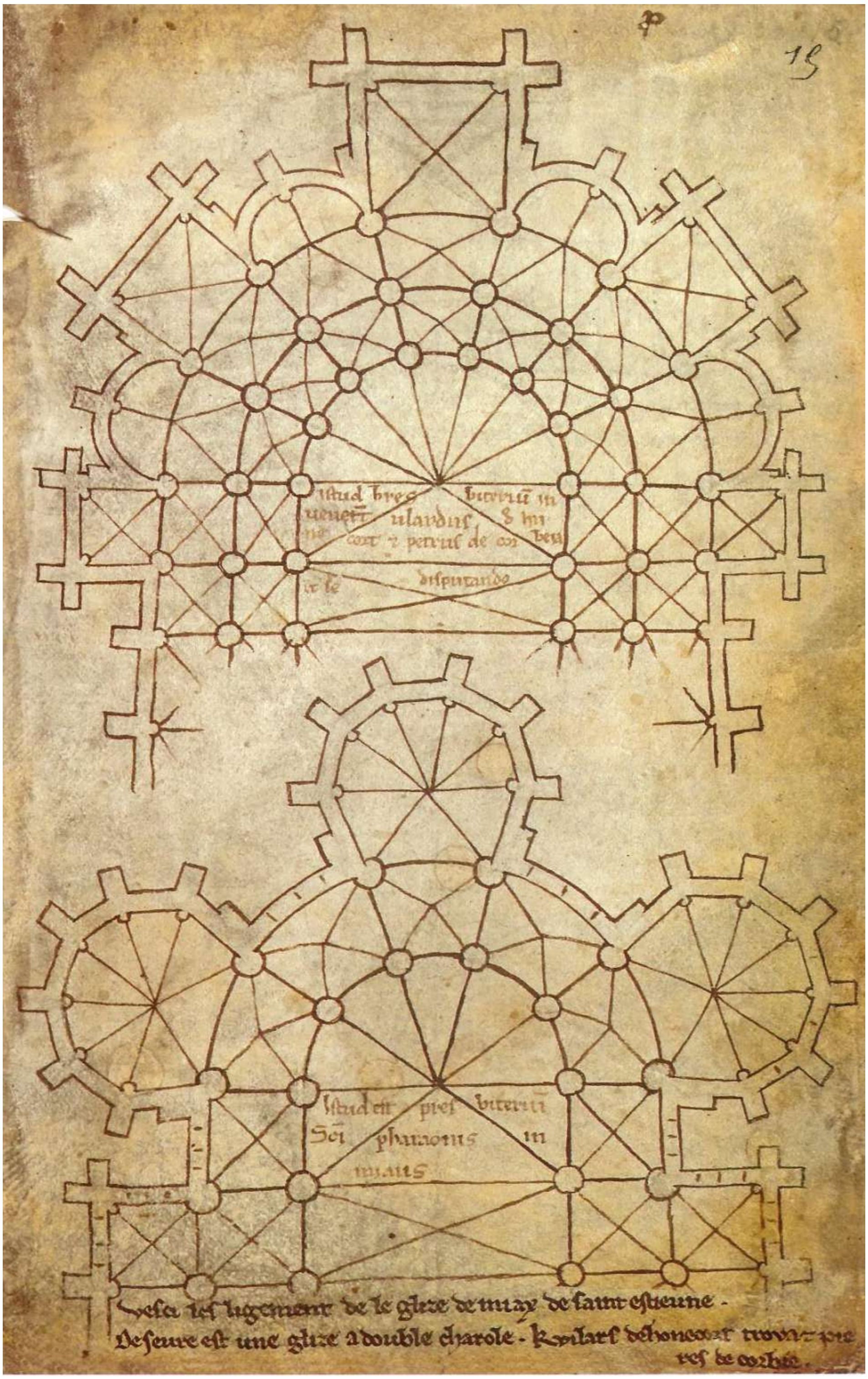


Figure 6: Villard de Honnecourt, ink on parchment, 1200 (Philipp, 2020)

These are all elements, which are used until today as communication tools on construction sites.

Floorplans

Floorplans of buildings or topographic situations are one of the oldest cultural techniques of humanity. Already 4000 years from now, floorplans from Mesopotamian buildings were created, and they nearly do not differ from plans, drawn from today.

Figure 7: Floorplan on a clay tablet, about 2000 B.C. (Philipp, 2020)

The drawing was carved into a clay tablet and shows a rectangular house with different proportioned rooms and their entries.

Modern floorplans contain more information than their 4000-year-old counterpart from Mesopotamia. In modern floorplans, materials are distinguished, additional information and measurements are provided and in general they are of much higher detail. But the principals, still are nearly the same.

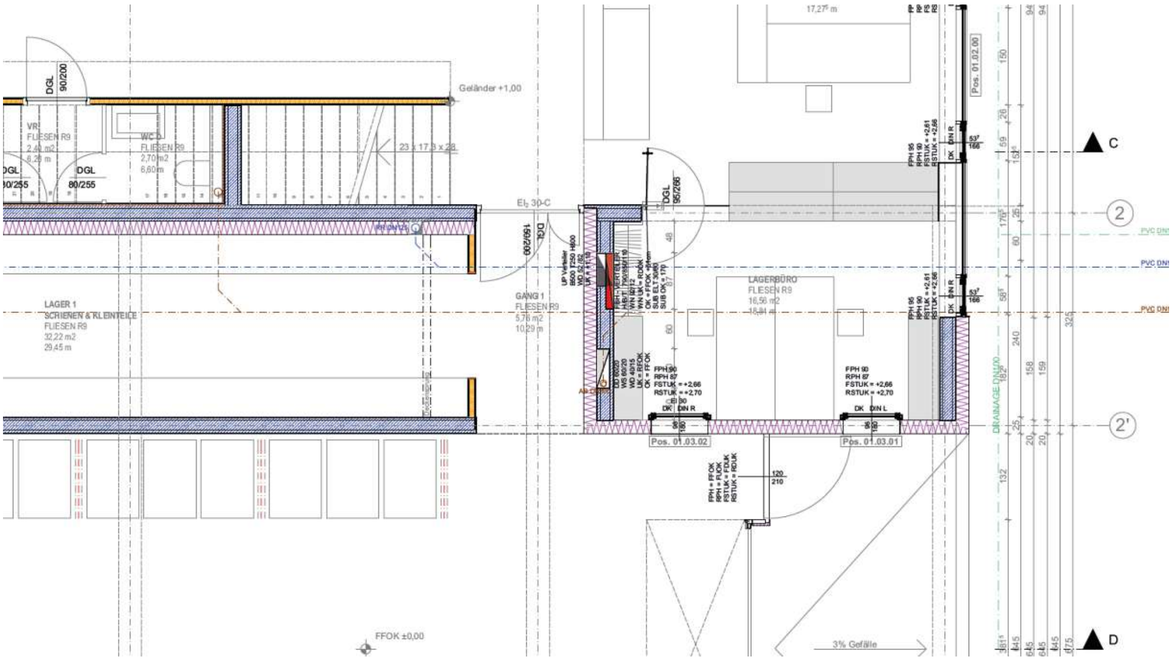


Figure 8: modern construction-floorplan with information relevant for the builder

The first definition, what a floorplan is, is found by Vitruv and he called it “ichnographia”. This word consists of two terminologies. First “ichnos”, which means “sole of a foot”, and “grapho”, which means “I draw or I write”. A “ichnographia” is created by using circle and ruler and shows the footprint of a building and its interior in a smaller scale. This definition is still viable today and is nearly similar in modern standards like the DIN1356-1 or ISO 128-1.

Elevations

The second kind of presentation of Vitruv is “orthographia”. This word consists of two Greek terminologies. The first is “orthos”, which means “correctly or upright” and the second one is “grapho”, which means “I draw or I write”. In connection to “orthographia”. The “orthographia” is a geometrical correct image of the frontside of a building. The term elevation comes from the latin word “elevare”, which means “lift up”.

One more term in connection to elevations is the facade. The facade describes the face of a building and often the “orthographia”, the completely correct geometrical description of it, cannot describe all aspects of the facade. The problem is that the facade of a building is in most cases not in one level. Windows, doors, part of the buildings that are receding build a relief and can be misunderstood if there are just two-dimensional presentations of it.

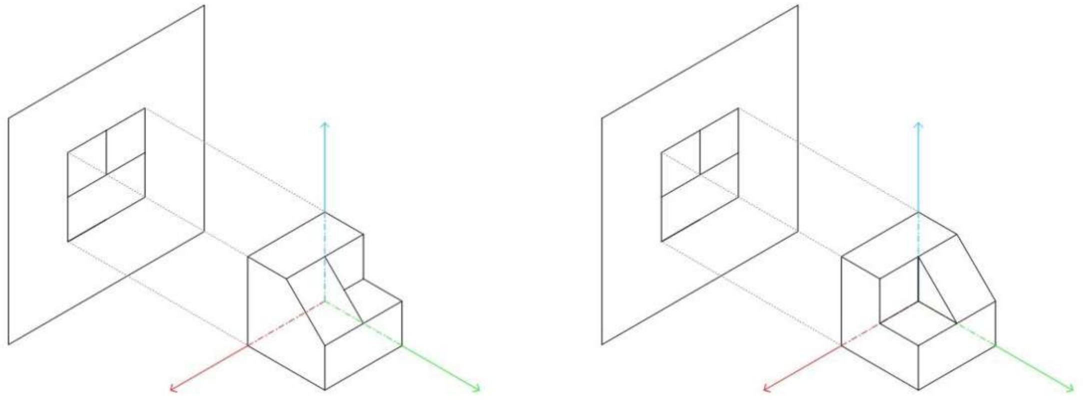


Figure 9: Elevation of two different objects appears the same way

To prevent such misunderstandings, graphic tricks were used since the first known modern elevations, like drawings from Villard de Honnecourt. In his drawing of the cathedral of Reims, he violates the rules of pure orthographical drawing very subtle to make it easier for him or other viewers to understand the three-dimensional structure of the facade. Situational some parts, like angled reveals, are shaded or subtle overlaps are used to describe different levels.

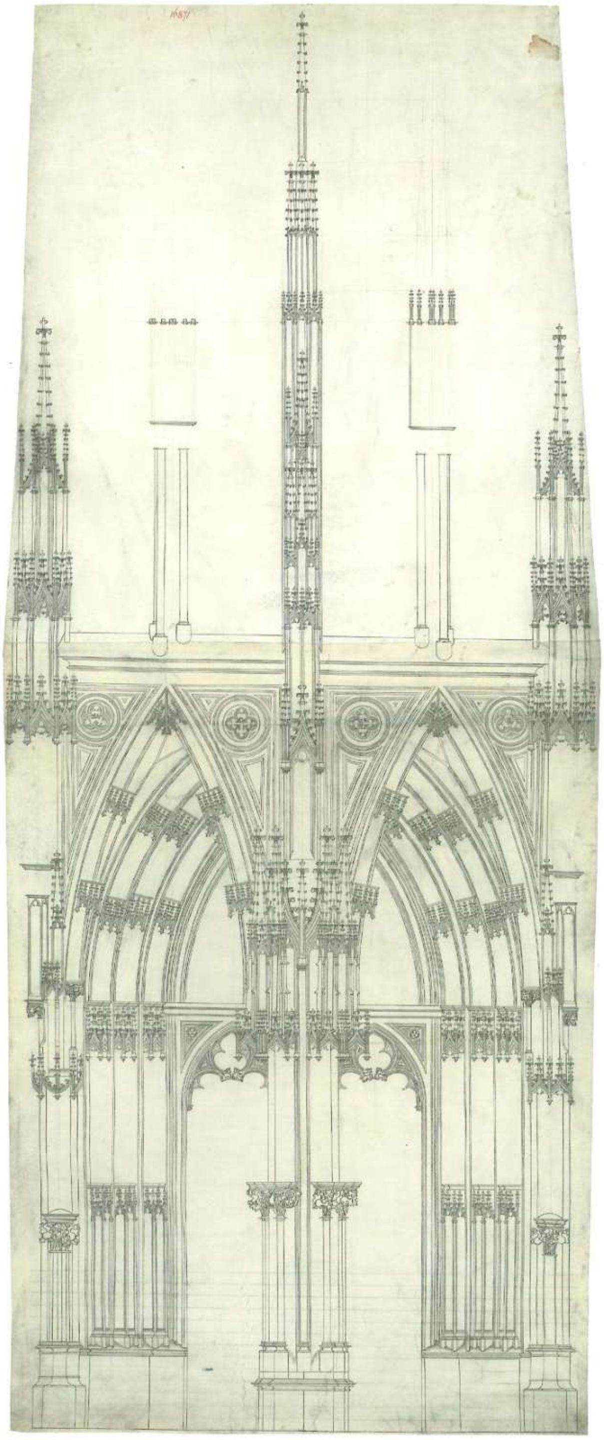


Figure 10: Villard de Honnecourt, Elevation of a chapel in the cathedral of Reims, about 1200 (Philipp, 2020)

To help describe facades better and still use orthogonal presentation, orthogonal drawings got so called cast shadows. These shadows use a light which usually comes in an angel of 45 or 60 degrees. This helps to understand the different levels of a drawing. Cast shadows are an analytical element, which help to understand the orthogonal drawings. An example are the drawings of the “Quattro Libri” from Andrea Palladio (16th century) and Ottavio Bertotti Scamozzi (18th century).

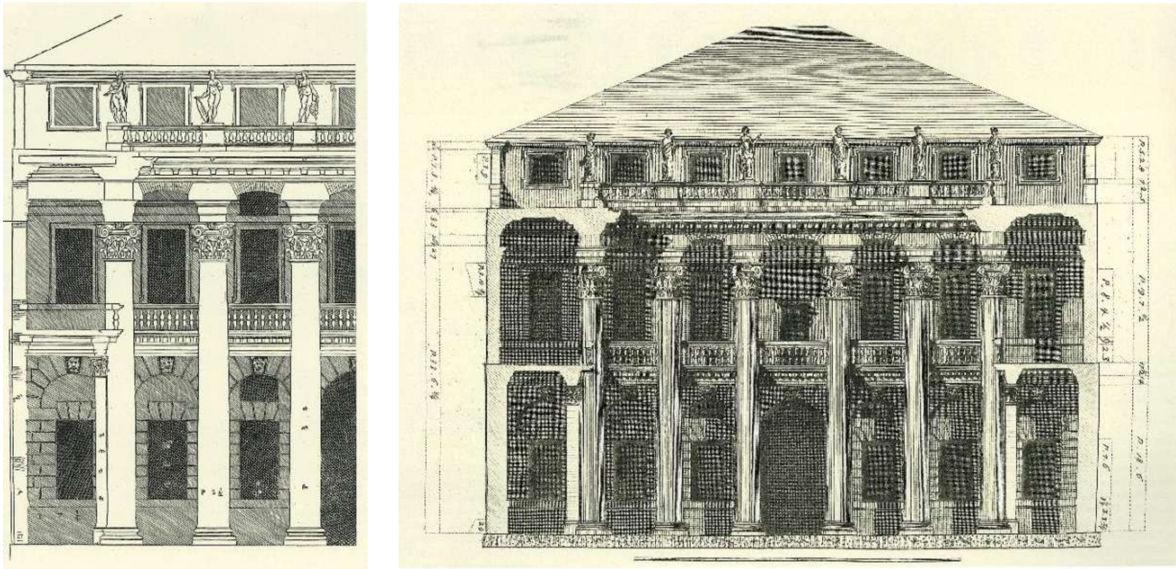


Figure 11:Left: Andrea Palladio, Elevation as copper engraving,1570; Right: Ottavio Bertotti Scamozzi, Elevation, 1783 (Philipp, 2020)

Sections

Vitruv did not know longitudinal or cross sections – they were none of his methods of presentation. Today sections are key elements to understand complex buildings and how they are structured. A section cuts through a building on a point which is useful for the understanding of this building.

The discussion about sections as standard method of architectural plans, was first held in the letter to pope Leo X. The process of creating a section is described by Raffael as follows: It is created from the floorplan, with parallel lines like in elevations and it describes the inner parts of the building like it has been cut in half. Furthermore, all measurements should be exact and not abstracted.

The right positioning and the correct drawing of a section is one of the more difficult works of planners. A section cuts or splits a thing and the part which is before the section is taken away. If the section is perpendicular to the viewer and parallel to the sides of the building, an orthogonal view is created. With this method doors, stairs, wall thicknesses and more can be seen. The section-line must be noted in the floorplan for better understanding. This line can be shifted to present more levels of the building for better understanding. A combination of section and perspective can reveal more of the building for a better understanding.

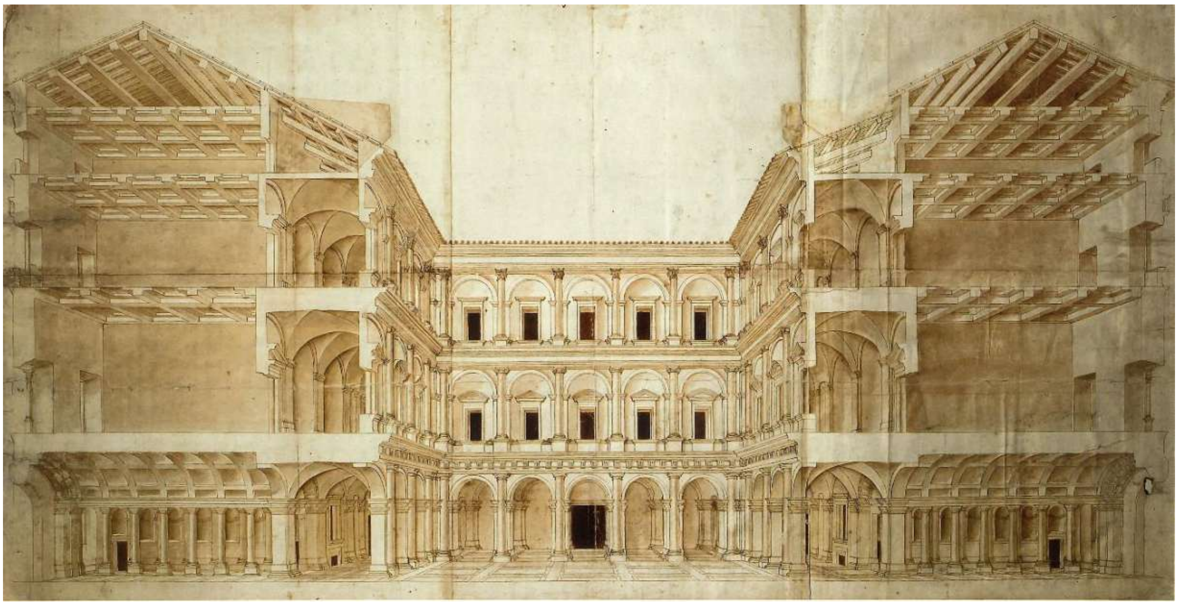


Figure 12: Anonymus, Combination of section and perspective, Design for the Palazzo Farnese (Philipp, 2020)

The problem with round or triangular buildings is, that lines in elevations are distorted, but in combination with floorplans, the true measurements can always be read. The same problem is with domes or arches, which are drawn in floorplans. Their real measurements can be read in sections or elevations. So, a correct combination of these three methods leads to a good understanding of a building and makes it possible to build it like intended.

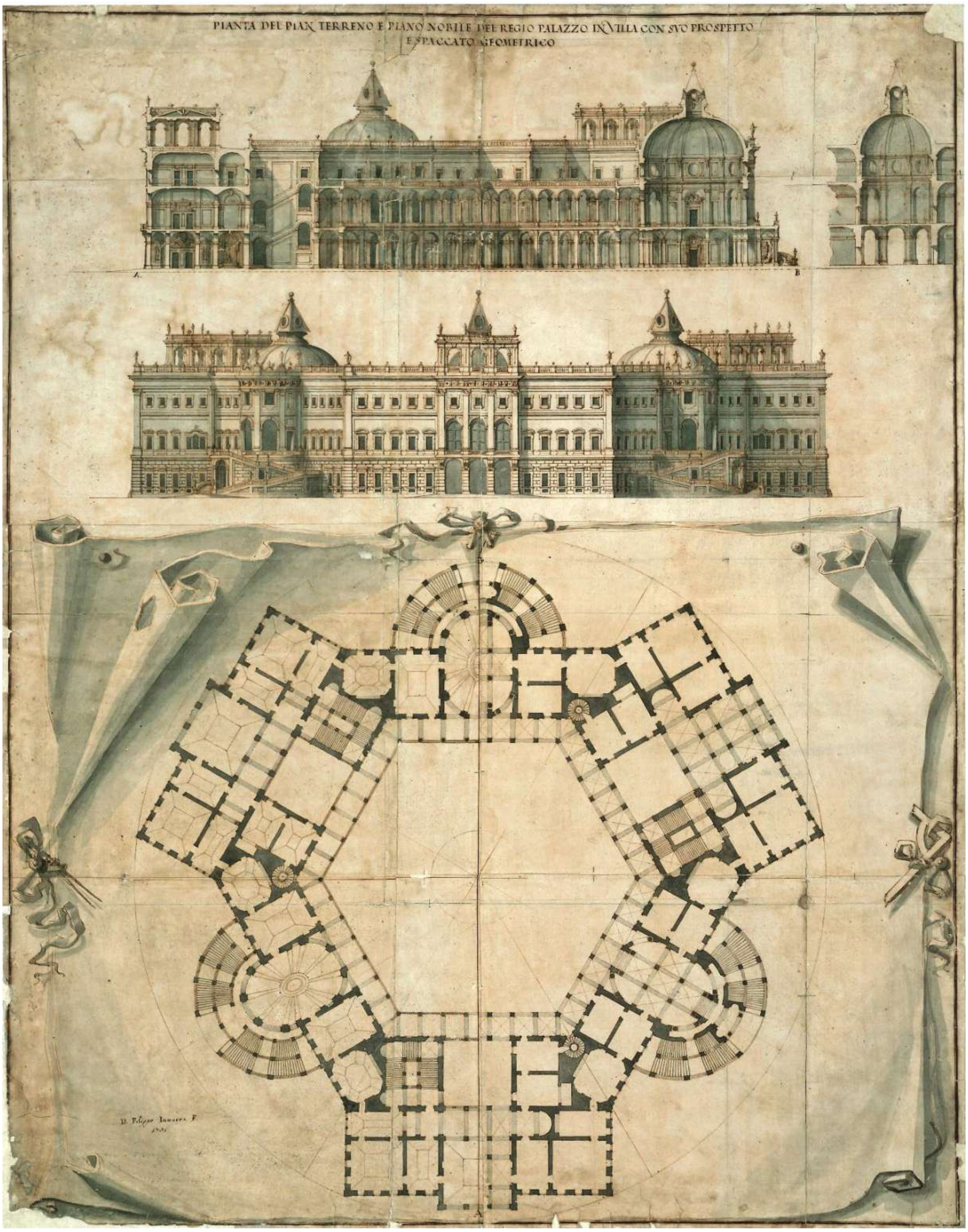


Figure 13: Filippo Juvarra, Design of Concerso Clementina in Rome, Ink on Paper, 1705 (Philipp, 2020)

1.2.2 Models and 3D-drawings

For faster understanding of a building, three-dimensional drawings were commonly added to the orthogonal drawings.

The wish for faster understanding of a design through layperson, is satisfied through architectural models, which has the feature of three-dimensionality and through that a better clarity then two-dimensional plans can offer. Models are self-explaining and do not need knowledge about the different sections through a building. They are built in different scales, to match their purpose.

Figure 14: Jaques Lemercier, Drawing of Michelangelo´s model for San Giovanni dei Fiorentini in Rome, 1607 (Philipp, 2020)

Models have been expensive and time-consuming to build in the past hundreds of years – so often 3D-drawings like perspectives or non-perspective drawings like isometrics or axonometries were added to the orthogonal drawings.

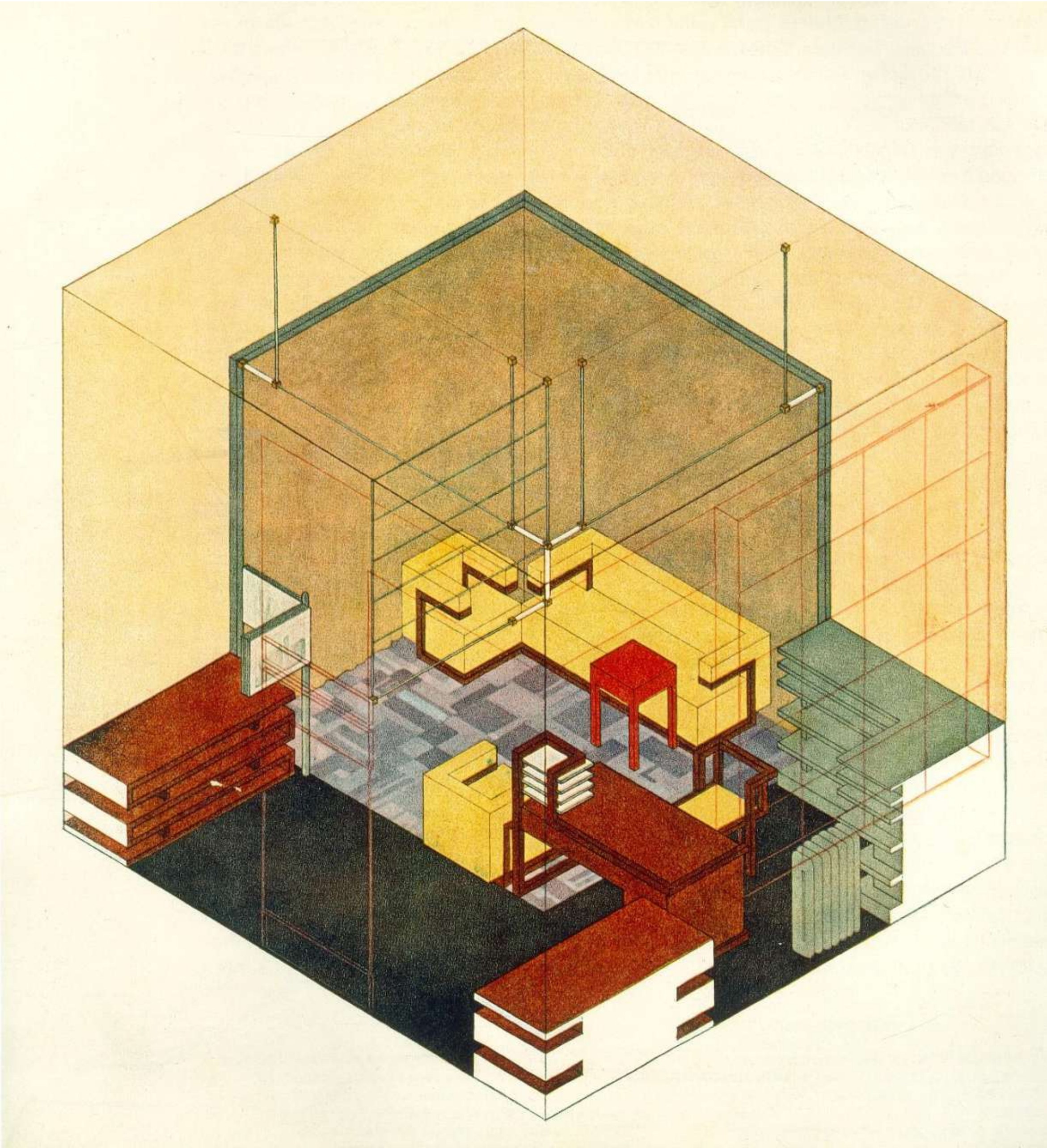


Figure 15: Herbert Bayer, Isometric drawing of the headmaster's office at Bauhaus in Weimar designed by Walter Gropius, 1923 (Philipp, 2020)

The third kind of presentation of Vitruv is "scaenographia". He described it as the combined reproduction of the facade and the sides of the building which go back and all the lines at the center of the circle. This is not easy to understand and is disputed between two camps. The first one thought it is the description of the not present section in Vitruv´s presentation methods. The other one thought it is the description of a central-perspective, like Filippo Brunelleschi invented it in the 15th century.

But the discussion was more about, if there is a need for perspective in the presentation of architecture, or are floorplan, section and elevation enough. Although the analytic presentation of a building is enough in theory, the perspective is till today an easy-to-understand method and is used in many different variations. The perspective shows all the efforts of the planner to keep floorplan, section and elevation congruent and represents the building as it will be built.

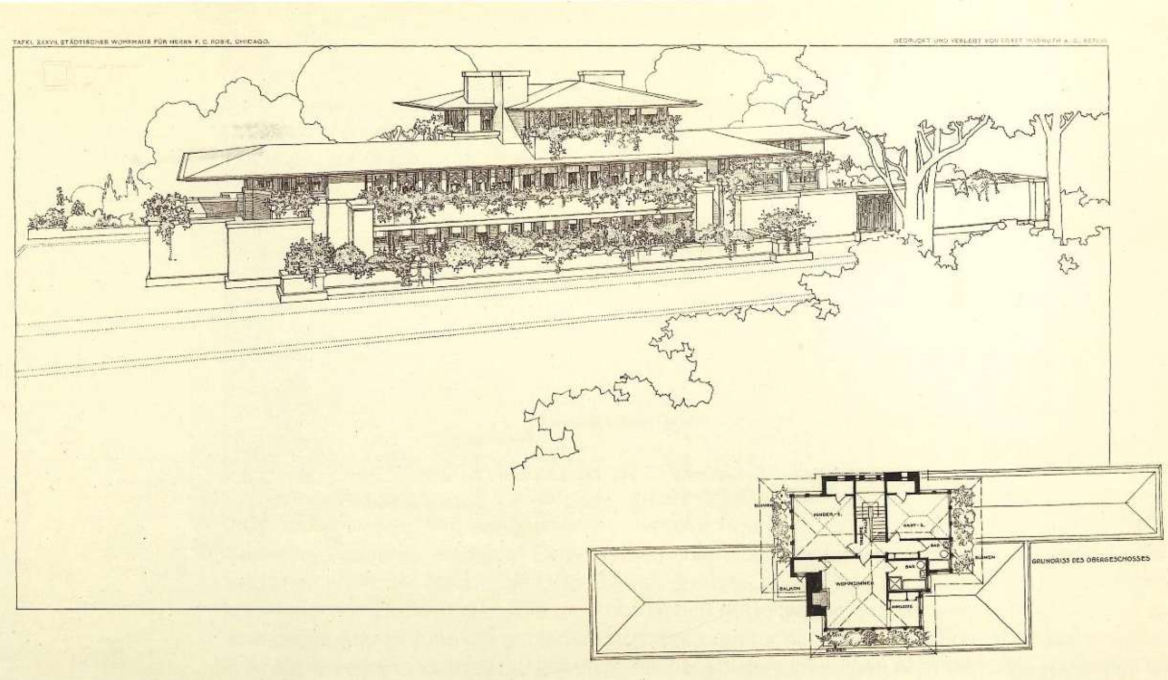


Figure 16: Frank Lloyd Wright, Residential building Robie in Chicago, perspective and floorplan, 1908

1.2.3 Modern Tools of Communication [2]

Since CAD, the production of architectural drawings became more efficient and with the development of more sophisticated CAD-programs, not only floorplans, elevations and sections became digital, but also the building models. Isometrics, axonometries and perspectives can be produced in an instant. Also, physical models are 3d-printed and to a certain scale, easy to produce.

Communication tools move from paper to other mediums, with the development of tablet computers and smartphones. Also, these devices are more convenient for construction workers than rugged laptops and information kiosks.

The development of mobile computing devices, has also led to innovation, of applications from CAD and BIM vendors and start-up companies, for these devices. Field BIM apps have three main uses:

- to deliver design information to the field
- to coordinate the construction process between all project partners
- to collect information about site conditions.

These applications utilize standard smartphones, tablets or utilize special equipment like VR/AR-head mounted devices, special GPS antennas or other equipment and a combination of various devices.

1.3 Current Process in Architecture, Engineering and Construction (AEC) [2] [8]

In 2017, the university of Innsbruck has created a representative survey about the current BIM situation in Austria. This survey showed that only 28% of the participants work with BIM in their current company. Experts estimated a portion of 10-15% of BIM-users ahead of the survey.[8]

This shows us that most of the participants of this survey use traditional methods to conduct an AEC-project.

These methods include using 2D drawings and most of the time splitting up the information of each trade working on the project. Errors and omissions in paper documents cause unanticipated costs, delays and sometimes also lawsuits between the parties. Although organizational structures like the use of project websites for sharing plans and documents improved the timely exchange of information, there are still severe and frequent conflicts caused using paper documents or their electronic equivalents.[2]

One of the biggest problems associated with 2D-based communication during the design phase is that critical assessments about cost estimates, energy-use analysis, structural details, and so on, are normally done at last, when it is already too late to make important changes to the design. This often results in compromises to the original design, which should be prevented. [2]

These problems often occur because of the nature of the whole designing and building process. There are a big three main contract methods which are currently used in AEC. A fourth one, is currently becoming popular with sophisticated building owners. In the following these methods get described in detail, to give an overview, which methods allow an easy implementation of BIM in an AEC project. [2]

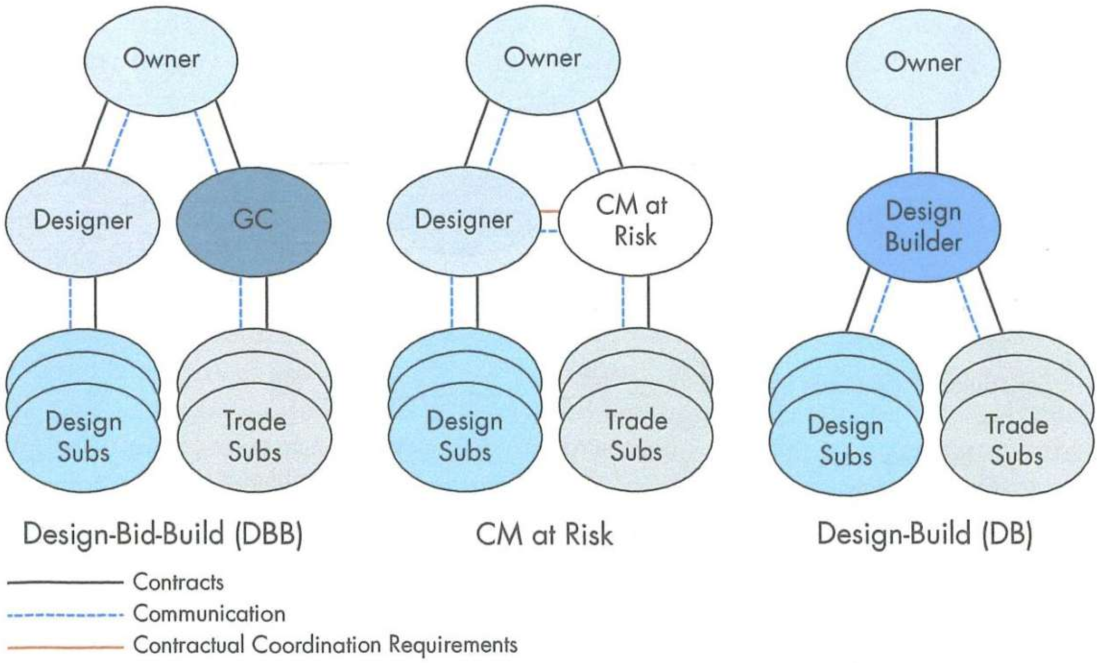


Figure 17: Schematic diagram of different contract methods used in AEC (Sacks et al., 2018)

1.3.1 Design-Bid-Build (DBB) [2]

This is the most used method. Design-Bid-Build (DBB) has two major benefits which are a more competitive bidding, to achieve the lowest possible price, and less political pressure to select the contractor, because you select the lowest price most of the times.

The first step in DBB the owner hires a planner (architect), who develops a building program in coordination with the owner’s needs. Then the projects design objectives are developed in various of phases like: schematic design and contract documents. The documents then need to fulfill the local building codes. In further planning HVAC (heating, ventilation, and air-conditioning), piping and plumbing, electricity, structural components must be planned and documented (3D-visualizations, 2D-plans, elevations) in more detail. This can be made from sub-planners or internal. The final set of documentation must be sufficient to facilitate construction bids. Because of legal reasons the planner may choose to include fewer details or inserts language that the drawings cannot be relied on for dimensional accuracy. These can lead to disputes with contractors and extra costs can arise, because of errors or misunderstandings.

In the second step, bids form general contractors are obtained. Each contractor gets the same set of documents and each makes an independent quantity survey. These surveys include all quantities from the contractor and the bids from subcontractors, to estimate the costs of the project. After checking all bids, a contractor is chosen, usually the one with the lowest, plausible price. Before work begins, it is often necessary for the contractor to redraw some of the drawings to reflect the construction work. The subcontractors and fabricators also produce their own shop drawings to reflect accurate details, such as steel connections, wall details, etc.

Shop drawings must be accurate and complete, because they are used for fabrication. If these are inaccurate or incomplete, or based on old or simply false drawings, then cost expensive conflicts can arise. Because of inconsistency and inaccuracy, it is difficult to fabricate building elements off-site and most of fabrication and construction take place on-site.

Often during construction, changes are made to the design as result of unknown errors and omissions, new client requirements and new technologies. Each change needs a procedure to determine how the issue will be resolved and who is responsible for costs etc. This can lead to legal disputes, delays and added costs.

The third step is the commission of the building after the construction is finished. Here the building systems like heating, ventilation, etc. are tested. Depending on the contract - final drawings are produced to reflect an as-built-situation with all changes made during the process. After handing all manuals, drawings and warranties to the owner, the DBB-process is completed.

The owner then needs to hand all relevant information to the facility management. If the information is conveyed in 2D (on paper or electronic equivalent), this process is time consuming, prone to error and costly.

1.3.2 Design-Build (DB) [2]

The Design-Build (DB) method was developed to simplify the process by giving the responsibility for design and construction into a single entity.

Therefore, the owner contracts directly with the DB-team, which is normally a contractor with design capabilities or working with an architect. They create a building program and schematic design that meets the needs of the owner. Then cost estimates for design and construction are made and after changes made by the owner are implemented, the final budget is established.

It is important to mention that the DB method allows to make design changes, due to construction details, earlier in the process. This leads to lower costs which are needed to implement those changes.

The DB contractor then consults with specialty designers and subcontractors as needed. After that the construction begins and changes to the design become the DB contractor's responsibility. The same counts for errors and omissions.

As a result of the simplifications of the DB model, the building is completed faster, with less legal complications and reduced total costs, but there can be concerns over quality management, because planning and building is in the same hand.

1.3.3 Construction Management at Risk (CM@R) [2]

In this method the owner retains a planner, who is responsible for the design, and a construction manager, who is responsible for construction management services for the project from preconstruction to the end of construction. These services include cost control, coordination of bid packages, scheduling and more. The construction manager is normally a licensed general contractor and guarantees the costs of the project. The owner is responsible for the design before a final price for the project can be calculated.

In comparison to DBB, CM@R brings the contractor much earlier into the design phase, where input can be made, and extra costs can be avoided. It also protects the owner from cost overruns. Still quality management should be at a high level.

1.3.4 Integrated Project Delivery [2][9]

Integrated Project Delivery (IPD) is a very new process and is still in development. There is not one clear approach for this method, because the industry is experimenting with it. There are sample contract forms published by various institutions like the American Institute of Architects (AIA).[2]

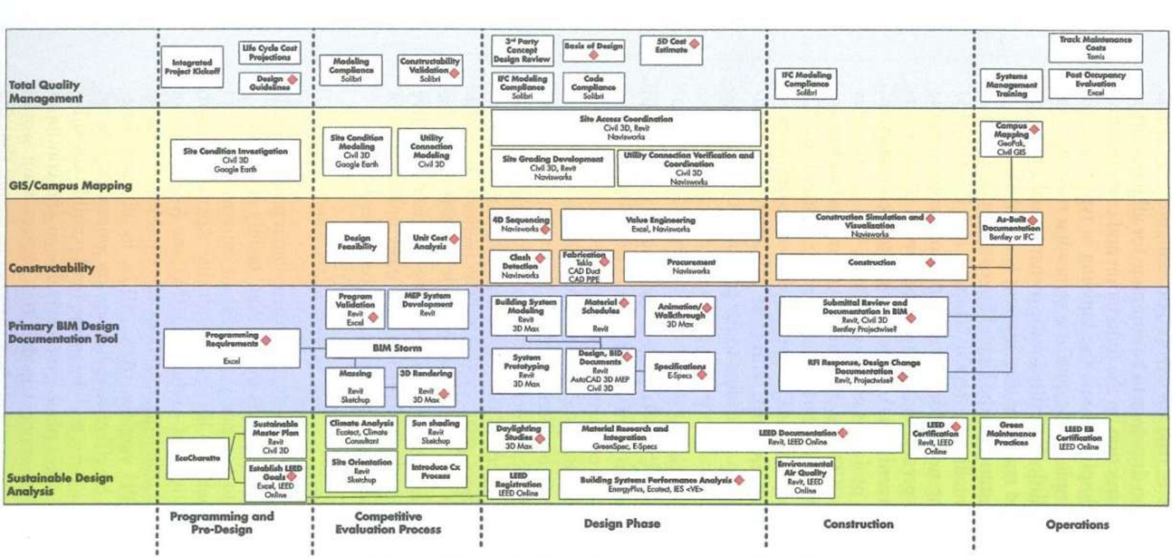


Figure 18: Example of Project Deliverables and BIM Software Interoperability Diagram (Sacks et al., 2018)

The core principles of IPD are mutual respect and trust. Effectively structured, trust-based collaboration makes the involved parties focus on the project goals and not on individual goals. Without this IPD will not be successful. IPD distinguishes from other methods by effective collaboration between owners, designers, and contractors. Collaborative tools must be established and early involvement of key participants leads to success of the project.[2][9]

To lead the project to success, the owner or a consultant must manage the process and represent the owner's needs. The tradeoffs, which must be made during the project can best be evaluated by using BIM and by using predefined standards and deliverables for each project-partner.[2]

The main beneficiary of IPD is clearly the owner, because IPD breaks with the current linear processes that protect and restrict information flow. But also, the owner/hired consultant must step up and must have enough understanding of the whole process to participate and specify in the contracts to the needs of all parties.[2]

IPD does lead to a more effective workflow on the whole project, but this does not mean that some stages are not more intense than in other processes. There is clearly an increased effort in planning, but this leads to a more cost-effective execution phase, which clearly can safe more money as the most costs accrue in construction.[9]

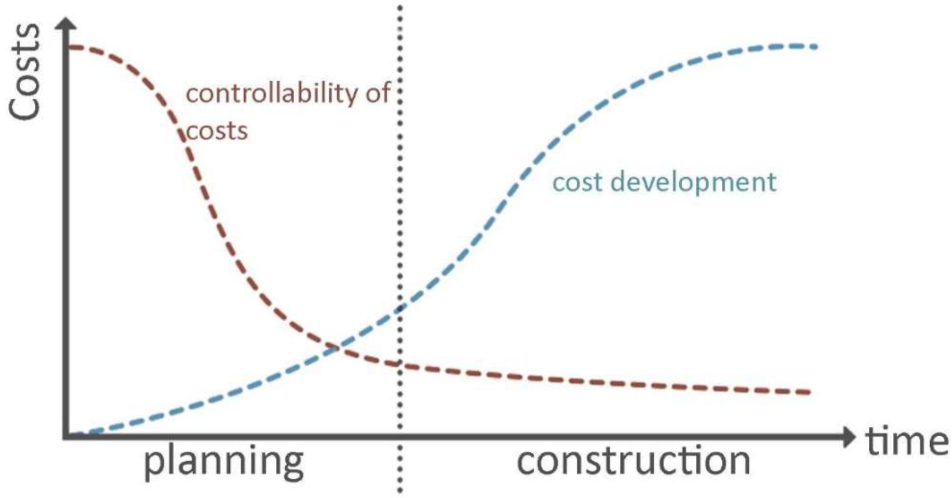


Figure 19: Predictability of costs during a project

The benefits of IPD for each party are the following: [9]

<u>Owners:</u>

The early and open sharing of information, which is key to IPD, streamlines the project and allows the owner to communicate his goals for the project more precisely and strengthens the team's awareness of the owners needs and desired outcomes. Therefore, the ability to manage the budget and control the costs is improved and set goals like schedule, lifecycle costs, quality and sustainability will be achieved more likely.

<u>Constructors:</u>

IPD allows the contractor to contribute his knowledge and expertise early in the design process. This leads to improved quality and better financial performance during the construction phase. The constructor's participation in the design process allows strong pre-construction planning, more timely and better understanding of design intentions, anticipations and resolving of design-related issues, early construction sequencing, and improving cost and budget control. Therefore, the ability to manage the budget and control the costs is improved and set goals like schedule, lifecycle costs, quality and sustainability will be achieved more likely.

Designers:

The early involvement of the constructor in IPD allows the designer to benefit from his expertise, such as accurate budget estimates which leads helps to make important design decisions. Also, design-related issues can be resolved before construction, which leads to better cost and budget control. IPD increases the effort during early design phases, resulting in reduced time in documentation, improved budget management and cost control. Set goals like schedule, lifecycle costs, quality and sustainability will be achieved more likely.

2.0 Building Information Modelling (BIM)

In this work there will be a strong focus on BIM. The reason for that is that there are a lot of potential values, which often will not be used to their full extent. In the practical part, side products like information about schedules which are part of a BIM-Workflow, are tried to be reused to fit an AR-communication-tool into the existing workflow as seamlessly as possible.

In the following part BIM and its development in the recent decades will be described to give an overview about this subject.

2.1 BIM overview and development

“BIM is the acronym of “Building Information Modeling,” reflecting and emphasizing the process aspects, and not of “Building Information Model.”” (Sacks et al., 2018, P.14) The BIM-Model is the product of this process and it is used in various ways in the building industry. The more sophisticated BIM got, the more information-loaded the BIM-Model got. This development started in the early 1990s and is still progressing fast and steadily.

When talking about BIM as a process, the objects which it operates with are building models, or BIM-models. They can be characterized by: [2][8]

- Building components that are loaded with information about geometry and data attributes that identify them to software applications, as well with parametric rules that allows them to be manipulated in intelligent fashion
- Components that include data, which is needed for further processes like, cost estimates, specification, and energy analysis.
- Consistent and nonredundant data, which reflects in all represented views of the component in which it is part.

A common description of the development of BIM is in Levels. They describe the maturity of application of information technology in construction. Also, the sophistication of use of individual tools and standards.

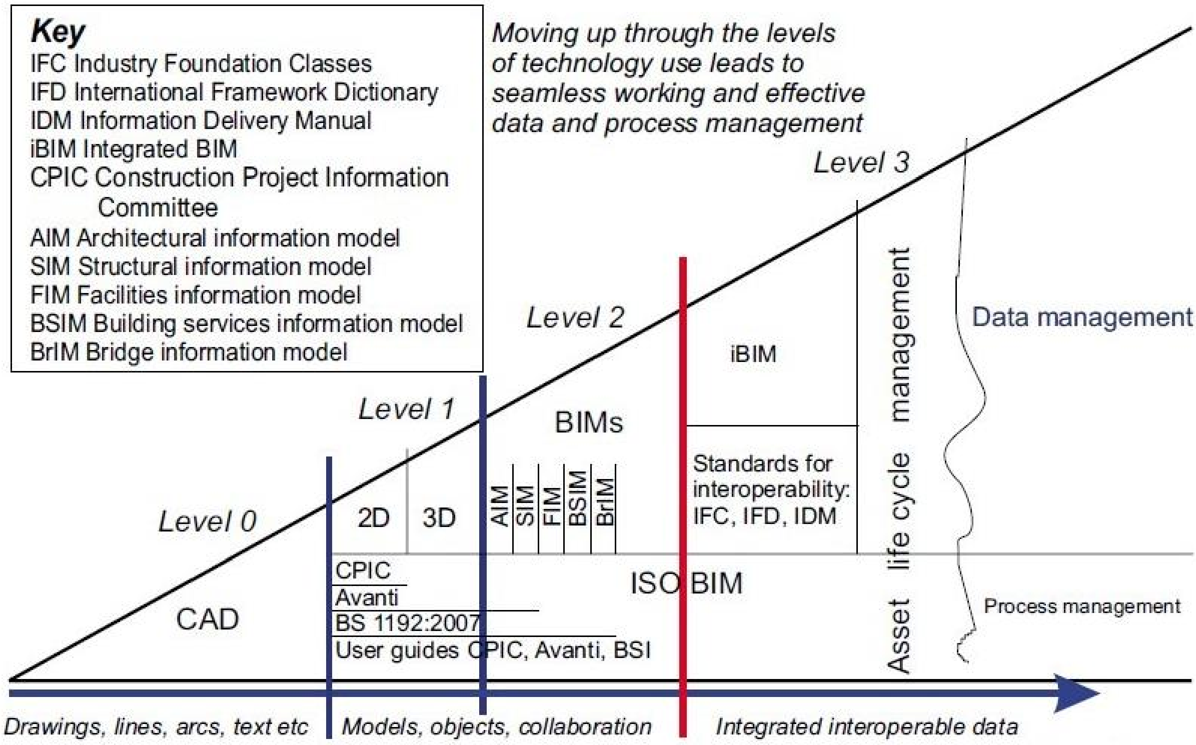


Figure 20: BIM maturity levels (Bew & Richards, 2008)

<u>Level 0</u>

This Level of BIM are the early stages of Computer Aided Design (CAD). Drawings were made of lines, arcs, text, etc., but it operates only in 2D. The information is shared by paper drawings or in some cases digitally via PDF.[2]

<u>Level 1</u>

Many companies operate on this level of BIM. It´s a mixture between 3D CAD for design work and 2D CAD drawings for approval documentation and sharing of information for production. Information exchange is over paper drawings or over a common data environment (CDE). [2] CAD standards are managed to BS EN ISO 19659-1:2018. [3]

<u>Level 2</u>

Collaborative work is the key asset of this level. Every party of the project has his own 3D model and the exchange of information must be managed. This is also called Closed BIM. Exchanges are made through a common file format such as Industry Foundation Class (IFC). Furthermore, the information of all parties is carried together to make a common

BIM model to run collision detection and other checks. The UK government has set this level as a minimum for all public-sector work by 2016. [2]

Level 3

This is currently the highest level of BIM and it means full collaboration between all parties in all disciplines. That means one single project model is used and shared on a centralized repository (normally on a cloud storage). This should remove the last layer of risk for conflicting information. This is also called Open BIM. Current concerns are about copyright and liability. These issues should be resolved by robust appointment documents and software originator/read/write permissions and by shared-risk solutions like partnering. [2]

Therefore, BIM is moving the industry forward. Current task automation of project and paper centric tasks (Level 0) develop toward integrated and interoperable workflows where these tasks are integrated in a collaborative process that takes maximal advantage of computing capabilities, web communication and integration of all data into one single entity (Level 3).[2]

2.1.1 Virtual Design and Construction (VDC) [2][10]

Virtual Design and Construction (VDC) is using BIM as a first-run study of the whole construction process. First-run studies are a standard procedure in lean manufacturing to monitor the problems of a new product chain and try to alter processes if needed before going into serial production. With the help of BIM and VDC, designers and builders test both the used products and construction process virtually. Supply chains, workflows, schedules, etc. can be tested and altered if needed to prevent extra costs in the construction phase.[2][10]

The VDC-models are multi-disciplinary. That means that they represent AEC and the owner. These models are performance models, as they try to track the performance of the building process and can also track the performance over the life-cycle of a project.[10]

2.1.2 Parametric Objects [2]

To understand BIM and its distinguishing from traditional 2D drawing, the understanding of Parametric Objects is key. Parametric BIM objects are defined as follows:

- Consist of geometry and associated data and rules.
- Geometry is nonredundant and no inconsistencies are allowed in all project views. So, a 3D object's dimensions must be correct in all project views.
- Parametric rules for objects automatically modify associated geometries. That means that a placed door, also modifies the wall and its shape. Or a light switch will be placed automatically on the proper side of the door.
- Objects can be defined at different levels of aggregation, so the wall can be changed but also its related properties like the material it is made of. So, if you change the weight of the material, also the weight of the wall changes.
- Object rules can identify violations of maximum input. So, object feasibility regarding size, manufacturability and so on is not violated.
- Objects can link, receive, broadcast, or export sets of attributes to other applications and models.

Technologies that allow users to produce building models that consist of parametric objects are considered BIM authoring tools.

Parametric modeling is not only used in BIM, it is also used in many other industries. There are a lots of different parametric modeling tools used for different industries. Because of performance issues and drawing production capabilities that use architectural conventions, which are needed in BIM, only a few general-purpose parametric modeling tools are implemented in BIM design applications.

The simplest level of parametric modeling is the definition of complex shapes or assemblies defined by a few parameters. This is called "parametric solid modeling". Editing these objects consists of the user making changes to the parameters of the object and then manually or automatically updating the object.

"Parametric assembly modeling" is the next step and allows calling in instances of individual parametric objects and relations between them. So, one object is changed and others also change according to their connected rules.

Another improvement is to allow the user to implement script-based rules or topology-based parametric objects. Most complex buildings today, like the Louis Vuitton Building in

Paris, by Frank Gehry, or the Dongdaemun Design Plaza, by Zaha Hadid, are designed and build using these technologies.

When using parametric functions, object classes can be defined and a set of these are called “BIM object library”. Each BIM application comes with “predefined parametric objects”, for its own field of use. Most allow their users to create their own “user-defined parametric objects”.

Often the “predefined parametric objects” can cover standardized scenarios like for steel construction, but the design is not standardized most of the times. So, a lot of tailor-made objects must be developed and there are a few ways how users can cope with that:

- Creating an object in other programs and importing it to the BIM tool as a reference object, without local editing capabilities.
- Creating a nonparametric solid model object and assigning the objects instances manually. Here it is important to update any changes!
- Defining a new parametric object family that includes external parameters and design rules to support automatic updates, but the updates are not related to other object classes.
- Defining an extension to existing parametric object families. So other object families can still communicate with this extension.
- Defining a new object class that fully integrates and responds to the context.

The first two options listed above take away the parametric representation and reduce the editing to pure 3D-solid-object-level. The other methods allow the user to integrate objects, which automatically update. The challenging part is when creating new object families with an integration of existing predefined objects such as doors and windows. Not all BIM tools provide such a level of user editing, but when objects are used frequently over more projects, this can be a key factor in deciding which platform is used.

2.1.3 Properties and Attributes [2]

Object-based parametric modeling deals with geometry and topology. This is good for 3D-modeling, but BIM includes a lot more.

For interpreting, analyzing and data exchange, properties and attributes are a key element. Properties come into play in various stages in the building lifecycle. Design properties address spaces and areas, like occupancy, activities, equipment, etc. Also, an aggregation of spaces can be defined as a zone, which can be used for thermal controls and loads. Also, properties address materials and quality specifications for purchasing. For production materials and connections may be refined to include all information for delivering the right product. At the end of construction, properties provide information and data for maintenance, so that the building can be properly used.

BIM can provide an environment to manage and integrate properties and attributes, to use them to their full extent. However, the tools for using them properly are still in development and integrated to BIM systems. Properties are frequently used for multiple purposes. As example a lightning application needs material color, a reflection coefficient, a specular reflection exponent, a texture and bump map. For energy analysis a wall needs additional attributes, like a λ-value of each used material. Because of the different purposes, properties are organized in "property-sets" associated with a certain function. Well organized libraries of property-sets are an integral part of a well-developed BIM environment. Although property-sets are an integral part of BIM, they are not yet standardized for common use, currently they are left to users to set up, or each organized in different ways in different environments.

Even seemingly simple properties can be complex. As an example, space names are used for various purposes. May it be in spatial program assessment, functional analysis, facility management and operations, early cost estimation and assigning energy loads. Space names are specific in nearly each building-type and even then, there can be different name coding in different companies.

Current BIM platforms just give a minimal default set of properties for most object families and provide the capability to extend these lists as pleased. Users of 3rd party application or developers must extend these lists or even create new lists as fit for their purpose. Managing property-sets becomes problematic when different programs for the same tasks need different sets of properties and units, such as for energy and lightning.

There are at least three different ways for managing property instance values for a set of applications:

- By predefining them in object libraries, so every time an object is placed, the set properties are an instance of this object. This is a good option for production work, but requires the user to carefully add sets to custom objects. Also, each object carries extensive information for various purposes. So, when exporting to a specific application, loads of data which is not needed is also exported. This can slow down analysis or simulations in the programs and enlarges the model size.
- By the user adding property-sets manually from a library if needed for an application. This allows the user to select the right property-sets for each program which leads to lean model size and fast processing, but the exporting process is time consuming. When testing iterations of a project to optimize energy performance as an example, it may be required to add the properties each time when testing a new iteration of the project.
- By the property-sets being added automatically from a database when exporting them to another application. The application needs to be identified by a key so that the right set is added to the export. This makes the model in the design application lean and so faster working is possible. But 3rd party programs need to develop exporting translators to associate a property set for each object. The different naming conventions in different programs, also make this difficult.

2.1.4 Digital Twin Concept [16]

With the rise of the “Internet of Things” (IoT), which is the connection of devices and virtual objects or information and their communication with each other, the digitalization of buildings is currently further developing in something new called a “Digital Twin” of a building.

The concept of a digital twin has been used for predictive maintenance in industries where capital-intensive equipment, like aircraft engines, are used. The main benefits of such digital twins are cost savings and reliability improvements.

To achieve a digital twin, 3d-models are connected to their real counterpart with sensors in the real world. This is achieved with the construction of smart buildings, which are fitted with sensors for different information. Over machine learning algorithms the data of these sensors is integrated and analyzed and connected with the 3d model. With this digital twin, predictive maintenance, the improvement of building operations and what-if-analysis can be undertaken.

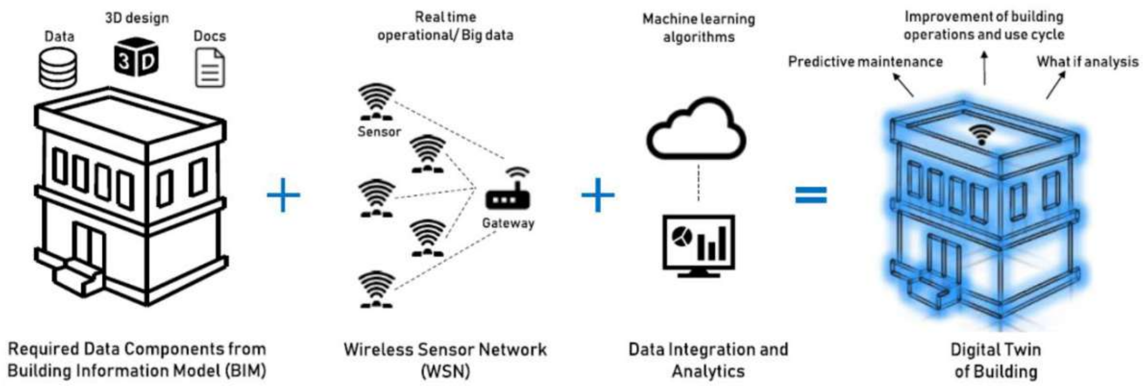


Figure 21: Components of a digital twin for buildings (Khajavi et al.,2019)

In contrary to a VDC-model, mentioned earlier, a digital twin always needs these components: the physical object and sensors, to run simulations. The sensors can provide real-time monitoring and data acquisition, like air temperature, airflow, relative humidity, lighting conditions, etc. With the help of this data and a simulation of the ideal environment within the digital twin can be undertaken and irregularities can be recognized. After these measures can be taken to provide the ideal environment in the real world.

2.2 Collaboration and Interoperability [2]

AECO (Architecture, Engineering, Construction and Operation) is a highly collaborative field. Over the lifetime of a building, multiple participants deploy multiple applications with overlapping data requirements for different tasks. From design to construction, operation and maintenance, the whole lifecycle requires collaboration between the participants. To improve design and construction management, better workflows for sharing and exchanging information between the parties must be developed. "Interoperability" is the ability to exchange information between different applications, which leads to better workflows and sometimes to their automation. There are various ways of exchanging information in BIM, so it is important to understand the limitations and benefits of each method for a good work-process-management.

Interoperability relied on file-based exchange, limited to geometry like DXF (Drawing eXchange Format) and IGES (Initial Graphic Exchange Specification). Also, direct links based on APIs (Application Programming Interfaces) are one of the oldest, but still important forms of Interoperability. In the late 1980s, data models or schemas were developed to support exchange within different industries. Led by the ISO-STEP (ISO 10303) international standards effort.

The main data model for building planning, design, construction, and management is the "Industry Foundation Classes" (IFC). In addition, many XML-based data models such as "Green Building XML" (gbXML) and "OpenGIS" are in use.

Many features of these standards describe the same product in different ways. To harmonize these standards, big efforts from organizations like the "National BIM Standard" (NBIMS) and "buildingSMART International" (bSI) are being undertaken.

While file and XML-based exchanges try to make exchange between different applications possible, a growing need is to exchange data from multiple applications through a BIM server, which acts as a common data environment, a building model repository, or a BIM repository. A prime aspect is that BIM servers would allow collaboration on object-level, rather than at file-level. A main purpose of a BIM server is to manage the synchronization of different models representing a project.

2.2.1 Data Exchange Methods [2]

Since the early days of 2D CAD, the need for exchange between different applications was a thing. The most used program in the late 1970s and early 1980s was "Intergraph" and some companies focused on translating Intergraph project files to other software.

After an effort of NASA, the biggest software companies came together to develop one common file format for their applications. The result was the "Initial Graphics Exchange Specification" (IGES) and by using IGES each software company only had to develop two translators, as it was thought. One for exporting from and one for importing to their application. IGES was a big success and was widely used in the CAD community.

But IGES was not the solution for all problems in interoperability. The main problems for seamless data exchange are the following and are separated in three groups.

1. Technical factors

Limited coverage of a data model. It means that the data of interest is not within the scope of a data model or exchange format. In case of IGES it supports only certain types of doubly curved surfaces, so other forms of geometric data are not supported.

Translator problems. The translator does not support the data of interest, although the data model does.

Software bugs or implementation issues. The data model is supported and the translator works, but the application has a problem loading or visualizing the data due to bug.

Software domain problems. The application does not support the data of interest. For example, a 3D software does not support building schedule data.

2. Procedural factors, especially when multiple people work on a project using multiple BIM models through different phases of the project.

Version control and concurrent engineering issues. If there are changes in design, while the structural engineers work on an old version of the BIM model, then the structural analysis may be redundant.

Level of development (LOD) issues. It would be perfect if there is only one BIM model for all stages of the lifecycle of a project, which has all the information needed. However, it is practically impossible to include all details for all uses in all stages. A recent BIM capability has been to develop guidelines for different levels of detail of BIM models for different uses. The BIM models in different LODs do differ in detail and in content of information. Consequently, BIM models need additional data adjustments for different LODs.

3. The human factor

Unwillingness to share information. Most of the technical and procedural factors can be overcome with strategies for data exchange, as far as project participants agree to share information with other team members. Sometimes, team members are unwilling to share information due to intellectual property, security, contractual issues, or sometimes without rational reason. This willingness-to-share-problem can only be resolved through negotiation or contractual mandate.

To achieve interoperability a data model or schema must be defined. This conceptually defines elements which are required for a target domain and the relationships between the elements. In general, data models are defined on three levels.

1. External level

   This level is the user's view on information exchange requirements. Each user needs their specific set of information to do their work. A data model specified from a specific user's perspective is called a subset, a view, a model view, a view definition, a model view definition, or a conformance class. The first and last step of data modeling is to specify and generate these views.

   The first step is called "requirements collection and modeling phase". Here the user's information requirements are collected.

   For the last step an export module for the user's software application must be developed.
2. Conceptual level

   This level is independent of an implementation method or an application system. The data model on this level is referred to as a "logical schema". This logical schema is a product of the consolidation of multiple user's views. An example for a logical schema is IFC.
3. Internal level

   On this level the logical schema can be implemented in various ways. For example, as a translator between different systems or database management systems. To implement a logical schema into a software application a mapping process between it and the internal structure of the application must be done. The internal structure is referred to as "physical schema".

Because of the wide spectrum of schemas, different data exchange methods can be deployed. The three main ways of exchange methods are the following:

Direct Links

This method is managed through Application Programming Interfaces (APIs). An API of one system can be used if a data model is not supporting the exchange between two applications. Some Direct Links may write a temporary file in the exchange and other may rely on real-time exchanges calling one application from the other one. Direct Links are implemented as programming level interfaces and come in different languages like, C++, C# or Visual Basic. These interfaces make some portion of the building model accessible for editing, checking, deletion, etc. Many such Direct Links exist, often within product families of a company and sometimes through a business arrangement between different companies.

Software companies often specialize on specific software, to support this one better. Analysis-tools and their interfaces can be tightly connected with a design software or even directly embedded. This can extend the capabilities of a certain software to attract more customers. The Direct Links and the information which is exported are most of the time connected to certain use-cases, defined by the two companies working together. Sometimes this use-cases are documented, but often they are not and so it is difficult to evaluate these interfaces for another purpose. Public definitions of BIM standards for use cases are driving the recognition that all building model exchanges need to be tied to a use case specification, if they are to be relied upon. But because Direct Links are developed and maintained by two companies involved, they are typically robust for the versions of the software, they were developed and functional for the level of detail they were intended. Direct Links are mostly functional if the business relationship between the companies holds.

File-based data exchange

This method relies on exchanging data through a model file either using a proprietary exchange format or a publicly open standard format. A “proprietary exchange format” is a data schema developed by a company for their software products. The specification for the schema may be confidential or publicly known. As an example, there are DXF and RVT, both developed by Autodesk or PLN by Graphisoft and DGN by Bently. Each of these formats is dealing with different kinds of geometry and has its own purpose.

"Standard exchange formats" are open and publicly managed schemas. As example there are IFC or CIS/2. IFC will be discussed in greater detail.

Model-server based data exchange

This method uses a "database management system" (DBMS) from which data is exchanged. A DBMS for BIM models is sometimes referred to as a model server, a BIM server, an IFC server or a common data environment (CDE). They are structured often based on a standard exchange format such as IFC to provide a common data environment.

The model server-based data exchange has the advantage over the file-based data exchange in regards of version control and concurrent engineering issues. It also has a great potential to reduce many interoperability problems by adding AI functions to a model server and therefore enabling automated analysis of the status and quality of data and filling in missing and conflicting information based on the analysis results.

2.2.2 IFC and buildingSMART [2]

As the CAD field has progressed from 2D to 3D and more complex shapes and information was added, the number of data types represented has grown constantly.

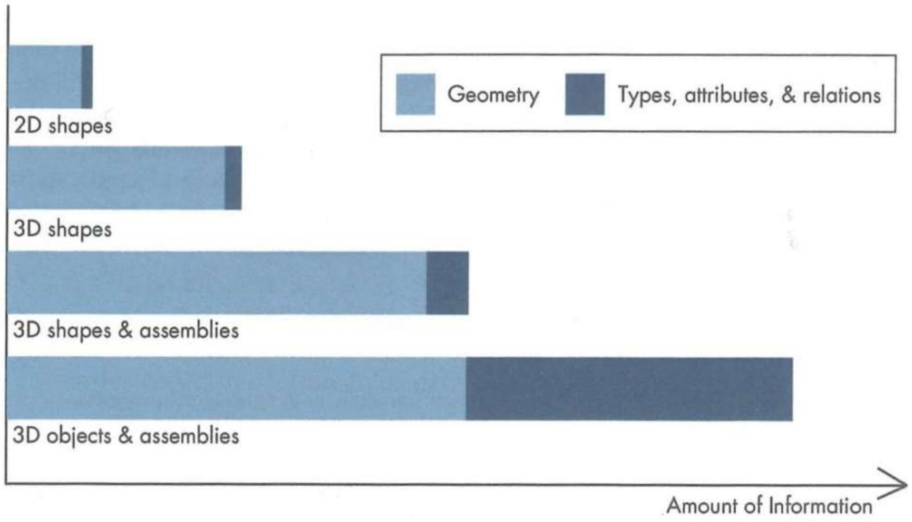


Figure 22: The increasing complexity of data for different exchange-types. Horizontal: approx. number of object classes within the schema (Sacks et al., 2018)

While 3D geometry of assemblies is complex, the addition of properties, object types and relations has led to an increase in types of information represented. So, the importance of data exchanges gradually increased. As the number of information inherited in a building model increases, the focus shifts from accurate translation to filtering just the information needed, and the quality of information.

With all this different kind of data, it is a natural desire to mix and match software tools to provide functionality beyond what can be offered by any single software platform. This is especially true when more organizations collaborate on a project as a team. Gaining interoperability is most of the time easier than forcing all involved parties onto a single platform. The public sector also wishes to avoid a proprietary solution that gives any software company a monopoly. IFC is a publicly and internationally recognized standard, which is likely to become the international standard for data exchange and integration within the AEC industry.

"The Industry Foundation Classes (IFC) is a schema developed to define an extensible set of consistent data representations of building information for exchange between AEC software applications" (Sacks et al., 2018, P.100). It is an industry-developed product data model for the AEC industry. It covers the full lifecycle of buildings and it is supported by buildingSMART and is also supported from most software companies. IFC is also an ISO standard (ISO 16739) and is developed on ISO-STEP technologies, but is not part of ISO STEP.

In 1994 Autodesk initiated an industry consortium to advise the company in creating sets of C++ classes that could support integrated application development. The consortium was called Industry Alliance for Interoperability and opened membership in 1995 to all interested parties. In 1997 the name of the consortium changed to International Alliance for Interoperability (IAI) and the main goal of this non-profit industry-led international organization was to publish the IFC as neutral AEC product data model. In 2005 the IAI changed their name to buildingSMART, reflecting their goal. There are meetings twice a year, to share best practice BIM projects and to develop IFC further.

While most of the other ISO-STEP efforts focused on exchange formats within specific domains in engineering, IFC was designed as extensible framework model. This means that it should provide broad, general definitions of objects and data from which task-specific models for exchanges could be defined. IFC has been designed to address the whole building lifecycle.

3.0 Augmented Reality (AR) [4][15]

There are a lot of different definitions what Augmented Reality is. The first AR was created by Ivan Sutherland in the late 60s, but the most backed up definitions comes from Ronald Azuma from his paper "A Survey of Augment Reality".

"Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality as it is more commonly called. VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it." (Azuma 1997)

An AR-System has, as Azuma says, three characteristic features:

1. It combines reality and virtuality
2. It is interactive and in real-time
3. Virtual content is registered in 3D

The combination of reality and virtuality is managed through an overlay of virtual objects to the real world. This means that the user registers his surroundings and the virtual objects at the same time, so that they build a whole new experience together. The virtual objects behave like real objects and are registered in a 3D space, so that they have a fixed place, which does not change as long it is not intended through an animation or user-interaction. Even if the user changes his perspective, the object stays in the same place through real-time rendering.[14]

As a more general definition, AR is an interactive and a real-time, real-world experience for all human senses, enriched through virtual content. These virtual content tries to mimic real content as good as it is possible, so that in a perfect case there is no difference between real and virtual elements. [4]

Implicit, this definition also has the three characteristics from Azuma, but it tackles AR from the side of perception. Most of the time AR is only an augmentation of visual objects, but AR can also stimulate other senses. Besides AR there is also the expression

"Mixed Reality" (MR), which also means that real and virtual content is mixed. Although MR and AR are often used as synonyms, MR is described in literature as a continuum.

3.1 Historical Development and Technical background [4][15]

The history of AR started in the 1960s with the work of Ivan Sutherland. With his research for immersive technologies, he took the first steps to AR in 1965, long before personal computers where invented. He wrote the essay "The Ultimate Display", in which he describes the possibilities of virtual worlds, where situations can be explored and tested, which are not allowed in our physical world.[15]

His "Head-Mounted Display System" consisting of a data helmet and a mechanical and ultrasound-based tracking system, allowed the user to see a simulated basic 3D-environment in the right perspective according to their position. The system can be seen as the first AR-system ever documented.[4]

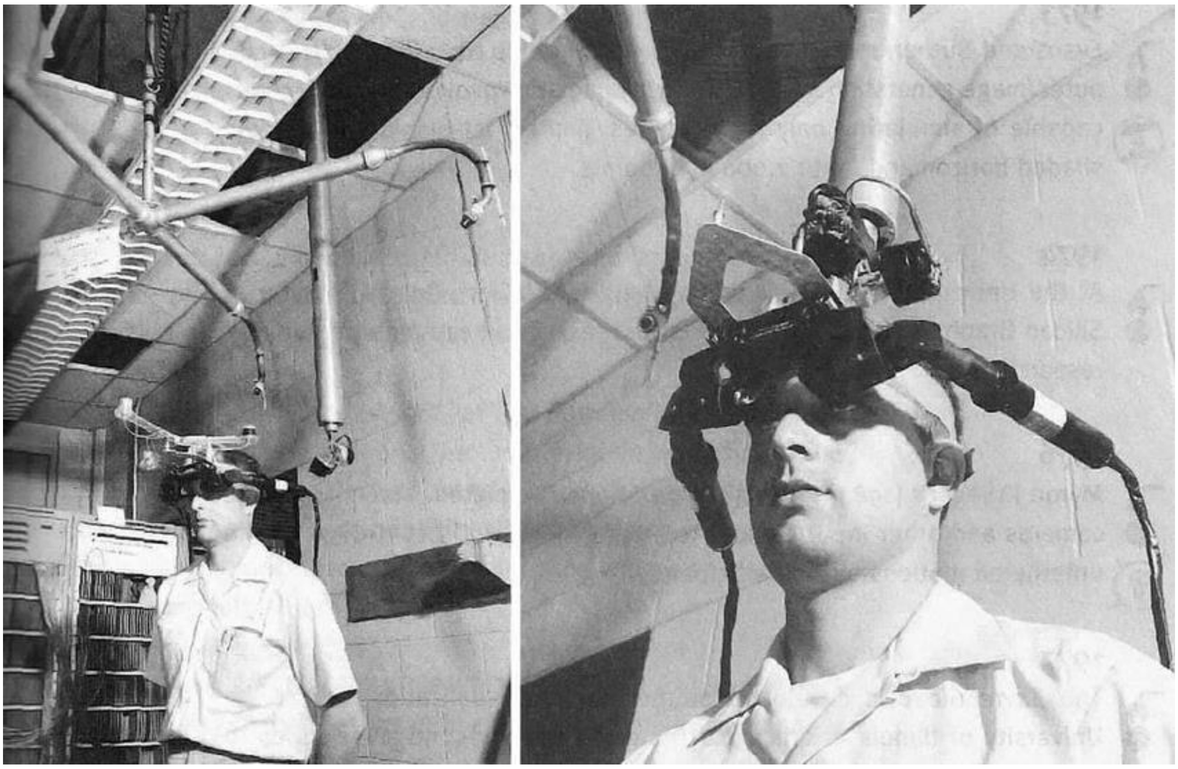


Figure 23: Ivan Sutherland´s "Head Mounted Display" (Sutherland; 1968)

Another big step in the development of AR was the "MARS" (Mobile Augmented Reality System), designed at the Columbia University in 1997.[4}

Figure 24: Replication of the MARS-System from 1997 (Dörner et al., 2019)

The publication of “ARToolkit in 1998 made computer-vision based tracking for AR possible and triggered a wave of research in AR around the globe.[4}

For a long time, the development of new AR and VR applications was in the hands of research facilities and big companies, but that changed 2013 with the presentation of the low-cost, high-end data eyewear “Oculus Rift”. Since the consumer-delivery in 2016 and the development of comparable hardware like “HTC Vive”, “PlayStation VR”, “Microsoft Mixed Reality Displays”, etc. VR boomed.[4]

Figure 25: Oculus Rift (www.oculus.com)

With the presentation of various AR software platforms in 2017, a big movement in AR-application development started. With Apple´s “ARKit” and Google´s “ARCore” two modern frameworks influence commercial AR-application development. Also, modern mobile phones and tablet computers are now equipped with strong hardware, which makes AR-application development more desirable.

3.2 General functionality of AR [4]

In unison to the definition in above AR is an enrichment of reality with digital components. In this process it comes to a merging of these two components.

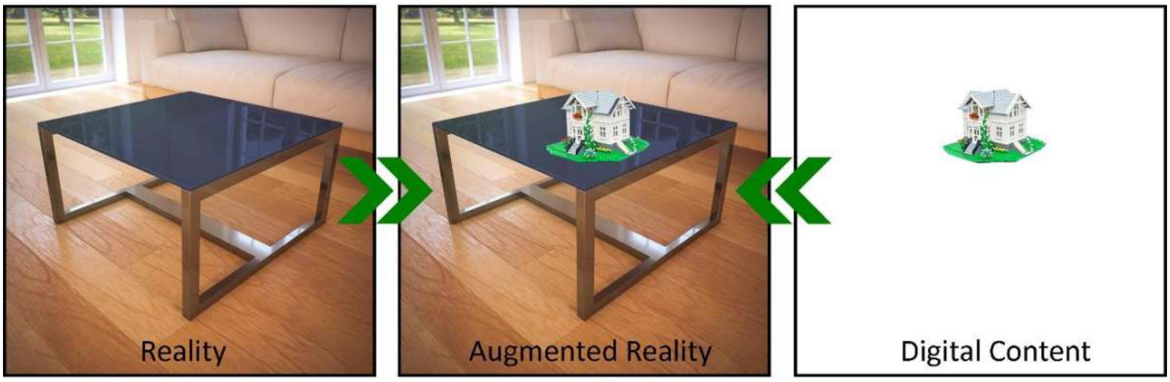


Figure 26: Schematic depiction of AR

It is critical that this merging is not static, but continuous and fitted for the user´s point of view.

3.2.1 Basic Implementation of AR [4]

Simplified AR can be implemented with the following five cornerstones:

- Videorecording
- Tracking
- Registration
- Rendering
- Output

In the following a quick summery of each step is given to provide an overview of the topic.

<u>Video-recording</u>

In the first step a video stream of the user´s surrounding is recorded. This happens over a camera. The better the quality (frames, resolution) of the video stream is, the better the following steps can be managed.

<u>Tracking</u>

Here the location and rotation of the camera is estimated. The better the estimate is, the more accurate the augmentation can be done. In the following chapter tracking and its various ways will be discussed in more detail.

Registration

Registration is the exact fitting of the virtual objects in the video stream. The quality of registration is inherited of the quality of tracking. The virtual objects must be fixed into the coordinate system. This gives a virtual, immobile object a place in the video stream, the same way like a real immobile object and is not altered by the users' movements.

Rendering

Based on the registration the virtual object gets transformed to match the user´s point of view and is rendered. In the process the video stream gets perspective correct overlayed and for a seamless merging the resolution and sharpness of the virtual picture must be corrected. Alternatively, to the overlayed video stream a direct overlay of the user's sight can be achieved with different hardware. This will be discussed in later chapters.

Output

At last the augmented video stream gets displayed on a screen. This can be the screen of a handheld device like smartphones and tablets, a head mounted display, or other devices.

This is the general workflow of an AR sequence. But there are a few different forms how an Augmented Reality is achieved.

3.2.2 Types of AR Implementation [4]

In the following, three different types of achieving AR will be presented and compared. These types are the following:

- Video See-Through-AR (VST-AR)
- Optical See-Through-AR (OST-AR)
- Projector-based AR

VST-AR

VST-AR is basically described above in the basic AR workflow. To get a good result it is important that point of view, direction, and viewing angle of camera and virtual camera are the same. Otherwise, there is a difference between the real surrounding and the recorded one.

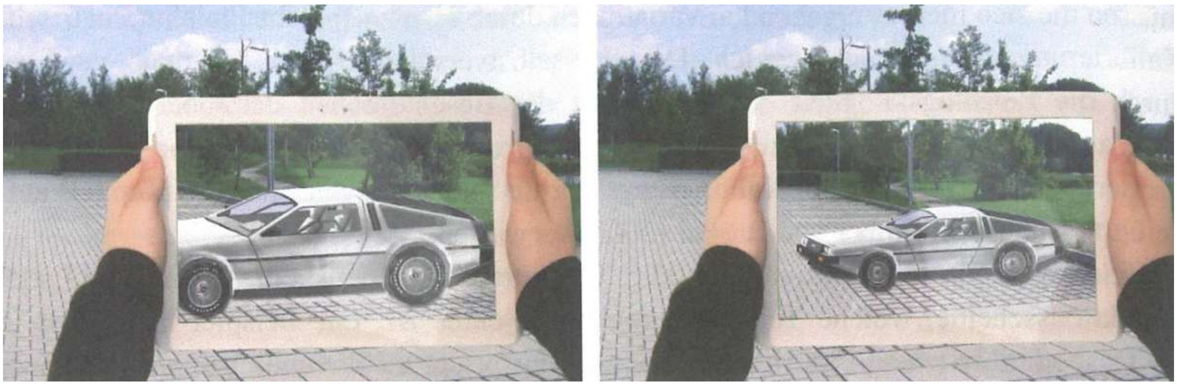


Figure 27: Perspective on the left overlaps correct with reality. Perspectives on the right do not match (Dörner et al., 2019)

OST-AR

In difference to the method described above, OST-AR does not necessarily need a video recording of the real surrounding. The real surrounding is directly perceived through the user and virtual objects are overlayed on a semi-transparent display. To align the perspective of the real and virtual objects, it is necessary to know the user´s point of view in relation to the display and generally there must be a display for each eye. Otherwise, the use of a stereoscopic display is essential.

Projector-based-AR

This is based on projectors which display virtual objects on top of the surrounding. As there is no way to create new spatial structures projector-based AR is mostly limited to manipulate surfaces like color or structures and add 2D information like symbols, highlights, or explanations.

Comparison

Generally speaking, the different methods differ in the possibility how far they can augment or alter reality. OST for example has problems with the overlay of dark virtual content with bright backgrounds. In difference to projector-based AR, the other two methods allow to place virtual objects at any position in the room.

As projector-based and OST-AR only add light to the scene, dark objects might get very transparent, because shadows are not possible to be added with these methods.

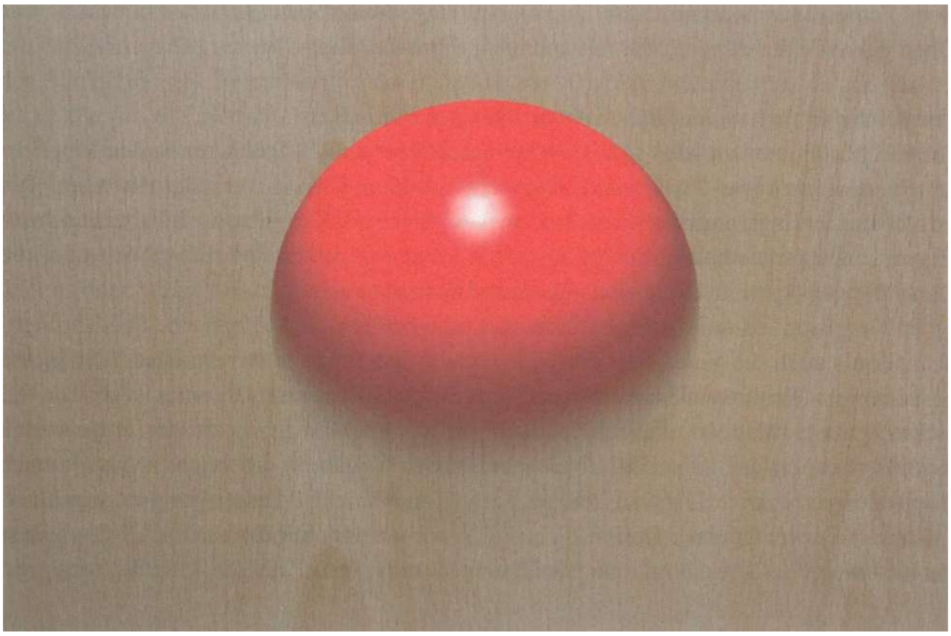


Figure 28: The dark parts and shadows of the red ball appear transparent in the augmentation (Dörner et al., 2019)

Generally, it can be said that with an OST-AR, objects always seem semi-transparent and the real surrounding is always a little dark tinted. The quality of the surrounding is not bound by the resolution of the display. With VST-AR the resolution of the real world is always limited by the display and camera, but virtual objects have more possibilities and are better merged with the video stream. It is important that the camera and the display have the same resolution, otherwise the virtual objects might set apart from the background.

Figure 29: Typical OST-perception on the left vs. VST-perception on the right. (Dörner et al., 2019)

The following tables give a quick overview about the differences of the different augmentation types.

Table 1: Display of bright virtual content in coherence to AR-implementation-type

bright virtual content	bright background	dark background
Optical See-Through	not good – high transparency	good – low transparency
Video See-Through	good	good
Projector-based	not good	good

Table 2: Display of dark virtual content in coherence to AR-implementation-type

dark virtual content	bright background	dark background
Optical See-Through	not possible – very high transparency	not good – high transparency
Video See-Through	Good	good
Projector-based	not possible	not good

Table 3: Display of virtual shadows and positioning of virtual objects in coherence to AR-implementation-type

virtual shadows and free object-positioning	virtual shadows	virtual object-positioning at any position
Optical See-Through	not possible	possible
Video See-Through	possible	possible
Projector-based	not possible	not possible

3.3 Tracking [4] [6] [7]

Tracking is one of key points in AR. Connecting the virtual world with the real world properly is one of the big challenges in AR to provide a good user-experience. Therefore, precise tracking is important to merge the two worlds.

Tracking is the continual determination of location and orientation of an object. To track a rigid body, where all distances between points are constant, 6 “Degrees of Freedom” (DOF) must be determined. DOFs are independent directions of movements and for a rigid body there are 3 for translation and 3 for rotation.[4}

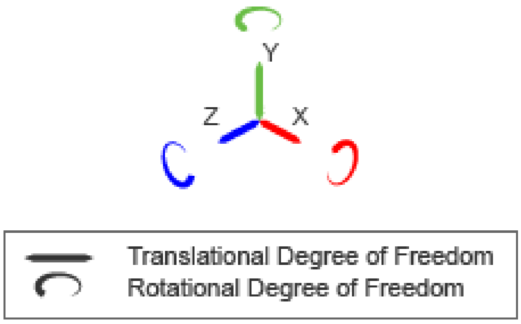


Figure 30: Schematic depiction of 6DOF

The goal of tracking is to determine values of the 6DOF of the followed objects for a continuous movement. The start point is normally the origin of the tracking system. When more than one tracking system is involved, then there must be a global frame of reference.[4}

There are a lot of different technics to track the user’s position, but in general there are two different types:

- **Inside-Out-Tracking**
  Here the sensors are attached to tracked objects directly. The location and orientation of the object is tracked in the relationship to its surroundings.
- **Outside-In-Tracking**
  Here the sensors are placed around the object and in correlation to each other the object is tracked.

The determination or estimation of an object´s position and orientation happen in a given coordination system. One possibility is the estimation in relation between objects. In this case each object´s relative transformation between camera- and object-coordination-system is calculated. [4]

Another possibility is that some objects share a coordination system, but then the translation between them in this given system must be known. In this case the translation between camera and the shared coordination system is calculated. Also, a mix of both approaches is possible. [4]

In the following part, different tracking-systems will be presented and their advantages and disadvantages will be discussed.

Acoustic Tracking

Acoustic tracking systems use differences in the time of flight (TOF) of sound waves. Ultrasound, which is not audible to the user, is used to determine the distance between sender and receiver. One of those must be attached to the object. A combination between 3 senders and 3 receivers can determine 6 DOF.

Acoustic tracking systems are relatively cheap in comparison to other 3d-tracking-systems. A downside of this systems is that they are very sensitive to temperature or air pressure. Every change needs the system to be recalibrated. [4]

Internal Measurement Units (IMU)

IMU base on sensors which measure acceleration. IMU are mostly used to track orientation and can be distinguished in linear sensors, which measure acceleration along an axis, and gyrosensors, which measure angle-acceleration around the axis. In most cases three sensors of each type are combined to cover 6DOF.

IMU combined with magnometers are standard in most phones by now. One thing what must be kept in mind though, is that IMU measure acceleration. So, errors accumulate over time, very quickly. So, a proper calibration and software is needed to overcome these errors. [4] [5]

Magnetic Tracking

Magnetic tracking uses artificial magnetic fields or the earth´s magnetic field to determine the position of objects. For this so-called fluxgate-magnometer are used to determine horizontal and vertical components to get 2 DOF of the current location. Smartphones use three orthogonal magnometers in combination with an IMU to get better results and prevent errors.

A very good thing about magnetic tracking is that cover from the sensors is not a big deal, as long there are no ferromagnetic metals nearby. This can also lead to the point that plastic screws are used to apply the sensors. When measuring the earth´s magnetic field, the sensors are very sensitive to artificial magnetic fields. Especially indoors it is not possible to get good results because of electric devices or cables. Here stronger, artificial magnetic fields must be generated to ensure good tracking. [4]

Laser Tracking

Laser Tracking always consists of a base and sensors which can detect laser beams. The base includes two lasers which rotate around a horizontal and vertical axis. When only one base is used, the sensors might be covered a lot through the user or other obstacles. So, in most systems more than one base is used to minimize that problem. To synchronize all bases infrared is used or the information for synchronization is sent directly with the laser beam. The time difference of the beams hitting the sensors is used to calculate the position of the sensors.[4]

For indoor applications laser tracking can be very accurate and low cost. Accuracies sub decimeter can be achieved with costs less than 1000€. But larger areas are difficult to monitor with low-cost-systems.[6]

Global Navigational Satellite Systems (GNSS)

When measuring outdoors, often GNSS are used. There are different Systems like GPS. Glonass or Galileo which can be used. When these systems are used for navigational systems, the accuracy can be enhanced by overlaying street maps. When using an AR-application the position is more difficult to determine. Accuracies above 10m are not rare and are not viable for most applications. To enhance accuracy Satellite Based Augmentation System (SBAS) or Differential GPS (DGPS) is used. SBAS uses geostationary satellites which provide data to correct the GPS signal in certain areas (currently available in North Amerika, Europe, and Japan). The signal can be corrected to an accuracy of about 1m. DGPS has a local reference receiver, of which the position is known, so that a corrected signal can be calculated. In this case the accuracy is in centimeter range, which is viable for most AR-applications. Other options are Assisted GPS (A-GPS), which uses the location of the used cellular cell, and WiFi location tracking, which uses known WiFi-networks to enhance the GPS tracking. [4]

Good things about GNSS are that they are easily accessible and nearly every modern device uses them. But without proper connection to the satellite-systems, good results are hard to achieve. This is especially the case indoors or in urban canyons (narrow streets, with high buildings). Though countermeasures to prevent inaccuracy can be taken.

Camera-based Tracking

Camera-based or optical tracking got very popular, because of high accuracy and flexible use cases. There are different methods of optical tracking, but they all try to calculate the position and orientation of the user in correlation to known objects in the video stream. Generally, there are marker-based approaches, which use easy recognizable markers (because of their color, contrast, shape, etc.) or approaches without marker. Marker based approaches will be discussed later in more detail. Markerless approaches use lasers or the camera recognizes so called features in the video stream, which help determine the position of the user.

Feature-based-tracking will also be discussed later in more detail.

Also, procedures can be distinguished by the way the cameras are used. Outside-In-procedures watch the object/user from a distance. Inside-Out-procedures connect the camera with the object/user and watch the surroundings.

Most Outside-In-procedures use more than one camera to monitor a bigger area and to facilitate coverage. Positive for the user is that less equipment must be carried around, but it can get expensive when covering a big area.

Inside-Out-procedures require the user to carry a camera, battery pack, senders, receivers and more, but they also do not restrict the interaction area. So, the user can move more freely.

For the user, a markerless outside-in-procedure would be the ideal situation, but markerless procedures are not as stable and precise as marker-based ones. [4]

3.3.1 Marker-based-tracking

Marker-based-tracking is used since the late 1990s and is still a widespread method today.

Markers can be very different in their appearance, but they all have in common that they are easy recognizable. Therefore, threshold filters are used to find the markers in the video stream. Basically, markers can be distinguished in active markers, which send light, or passive markers, which reflect light.

Passive markers can be black and white markers with a certain size. They are easy to find in the video stream of RGB-cameras. Also, these markers can be colored, but often this will lead to worse results when the scene is not good illuminated. Also, passive markers can be reflectors. To illuminate the scene better, without blinding the user, infrared-cameras can be used.[4]

Active markers can be electric lights (PlayStation Move controllers) or light sticks (chemical reactions are used – they don´t need electricity). Also infrared-LEDs can be used as active markers (Nintendo Wii). [4]

Tracking of black and white markers

This is the most common method of marker-based-tracking. Black and white markers are easy to use and have an advantage over colored markers, because they are less sensitive when the lightning is not stable. They are easy recognized with the help of threshold filters. [4]

<u>The basic functionality of this tracking method is the following:</u>

1. The camera is recording a video
2. In the frame of the video an area with four adjacent lines is searched

3. It is checked if the found area is a predefined marker
4. If it is a marker, then the position an orientation of the camera is calculated in relation to the four corner points of the marker.

Most markers are squared and have a completely black border. Different systems which use black and white markers are ARToolkit, ARTag, ARToolkit+, IS 1200 VisTracker, etc. [4]

To use marker-based-tracking the size and shape of the marker must be known. Some systems like ARToolkit let you choose the inner template of the marker, others like ARToolkit+ have own templates. The latter have better performance with multiple markers and markers can be partly covered in some cases. Generally, markers should always be seen whole in the video stream. The right measurements of the marker are also very important. If the marker is too big, it will be covered partially when standing near the marker. If the marker is too small, it can be difficult to be recognized in the video stream, because of to less marker-pixels. So, it can be beneficial to use markers of different size simultaneously. [4]

A way of coping with this problem are fractal markers. They have markers within the marker, so that regardless of which part of the marker is seen in the video stream, there is a marker in proper scale. [7]

Figure 31: Fractal marker (Herout et. Al., 2012)

Another problem markers have is, that when they are viewed from a low angle, problems with the calculation of the transformation-values show up and the marker is not recognized. Also, lightning is a crucial part. If it is too bright, then problems with reflection can accrue. If it is too dark or shadows overlap with the marker, the white and black parts can´t be separated anymore and the tracking is not possible. [4]

The main reason to use black and white markers, is that they are easy to implement in media like books, etc. Also, it is easy to print them out and attach them on walls, floors, ceilings.

A disadvantage is that markers must be attached near the augmented object, because otherwise they won´t be recognized when looking closely at the augmented object. Also, they cannot be attached everywhere because sometimes it is not appropriate (like on art, etc.).

3.3.2 Feature based tracking

Next to marker-based-tracking, feature-based-tracking is another very promising tracking-approach for camera-based-tracking. Features, which are known and saved in a databank, are recognized in the video stream of the camera. These features can be based on 3D- or 2D-objects.

Geometry-based-tracking

Features like edges or corners of geometry are recognized and extracted. Based on an extrapolation (an update) of the translation of the last picture of the video stream, all distances to the object are calculated which are needed to determine the 6 DOF.[4]

Often single features are not unique, when dealing with simple geometry like a cube. So, there can be more than one valid position to the current camera-picture. Based on the current position the next position will always be the one which represents the least change to the last position. Decisive is the initial position of the tracking and for right calibration other approaches like marker-based-tracking are often used.[4]

Visual-tracking

Other visual features, contrary to edges and corners, are often hard to recognize for the human eye. But they are tracked reliable with corresponding feature-detectors in the video stream. When enough of these features are detected, they can be compared to databases which hold information of the underlying 2D- or 3D-geometries. After sorting out false detected features, the position of the camera can be calculated.[4]

Another possibility to achieve camera-based tracking, is the combination of color-cameras and depth-cameras. These are called RGB-D cameras. They can be used to determine the camera position and to track user interactions. Most of the time RGB-D cameras use an infrared projected pattern or a time-of-flight approach.[4]

3.3.3 Visual SLAM (Simultaneous Localization and Mapping)

For the tracking approaches above, which are computer-vision based, it is always necessary that marker, pictures or objects and their corresponding features are known. That made it possible to determine the position of the camera. But with SLAM-approaches it is also possible to achieve the position tracking of the camera without knowing any features about the environment before. This technology was developed in robotics and is rather advanced in this field. The use of LiDAR-sensors (Light Detection and Ranging) is common in robotics, but in AR still in the early days.[4]

To initialize the SLAM, the coordinate system will be based on the first position and the features found in the video stream are constantly added to a map. The map consists of only a few features, but with camera movement, new maps are created compared to the existing ones. New information is added and based on the updated map the current camera-position is estimated. At first this leads to errors in the quality of the map and localization, but the more information can be recognized from previous stages of the map, the better the result will be. The key factor in this process is that old features are reliably recognized in updated versions. Another problem are dynamic objects like moving cars or people. Information about dynamic objects must be filtered and ignored, otherwise they might cause errors in the map and errors in tracking.[4]

3.3.4 Hybrid tracking methods

For AR often combinations of different tracking-methods are used. The cause for this is to improve results by using the right tracking method for the right situation. A typical example for this is a marker-based approach, which provides very good results for certain situations. Although, when the marker is partly covered, not in the video stream, or too far away, tracking becomes impossible and the augmented objects get lost. To not lose the illusion of an augmented reality, the combination of more methods is necessary.

3.4 AR Development [4]

AR development is a very complex task. Various skills are needed to develop an AR-application. Programming skills, knowledge over real-time computer rendering and image editing, designing of human/machine interfaces and know how in usability. Also, the creation to build assets like 3d-objects, sounds or textures is needed and knowledge in the specific use scenario. Most of the time one single developer cannot cover all this fields and support from outside is necessary. This can be covered by different people or toolkits which have been developed.

Usually support comes in form of programming libraries, together with programming toolkits like Software Development Kits (SDKs) or Application Programming Interfaces (APIs) to different software packages or -systems.

Another way of support are software tools, which help the developers in different tasks. These are often not tools developed for AR exclusive tasks, but for wider use, like photoshop or animation software like blender or 3ds Max. Typically a lot of different tools are used for the different aspects of AR-development. The process of exporting data from one application for the use in another application is called tool-chain and is one of the main sources of errors in the process of AR development. If there is no or only restricted interoperability between different software tools, then data loss might take place when converting to a supported format.

The choosing of the different tools and interfaces is called authoring process and must be planned carefully, when more developers are part of the process to ensure a smooth workflow. Unfortunately, there is no universal guideline for authoring processes and a fitting one must be developed when creating an AR application.

3.4.1 AR Development Platforms [4]

When developing an AR application, most of the time game engines are used in the process. The two most popular engines are “Unreal Engine” (UE) and “Unity”. Both platforms are offering the same functionality and their structure is built similar. They both offer support for level design, real-time rendering, multiplayer applications, AI, user interfaces, physic-simulations, global lightning effects, animation and more. Also, on both platforms there are a lot of free pre-built applications and assets (Unity: Asset Store; Unreal: Marketplace).

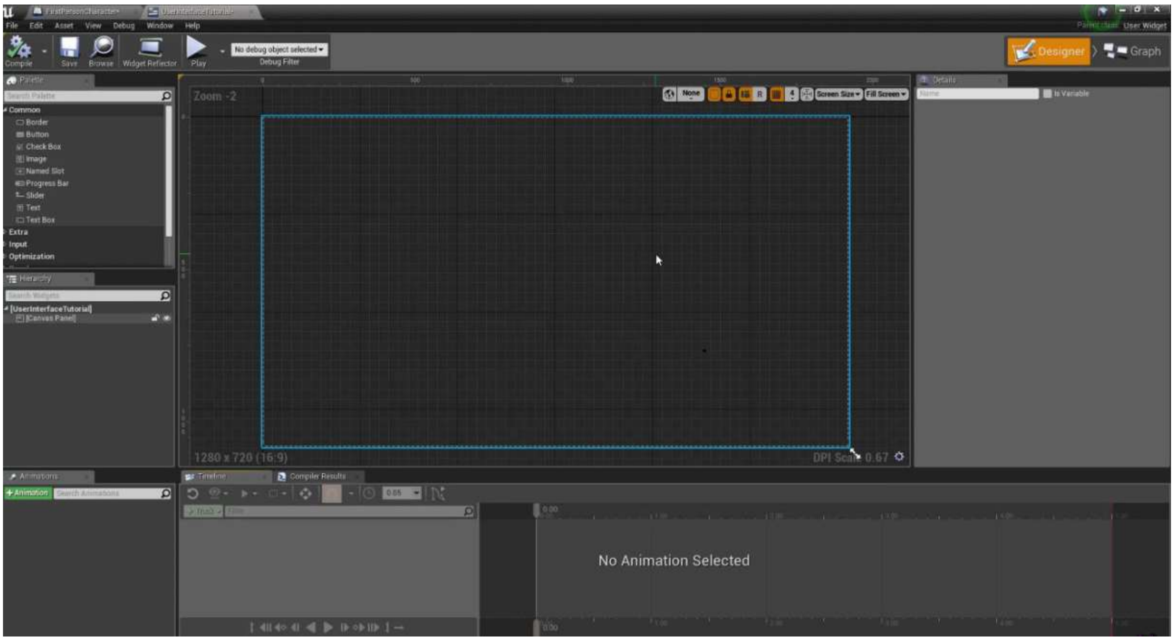


Figure 32: Unreal Engine User Interface

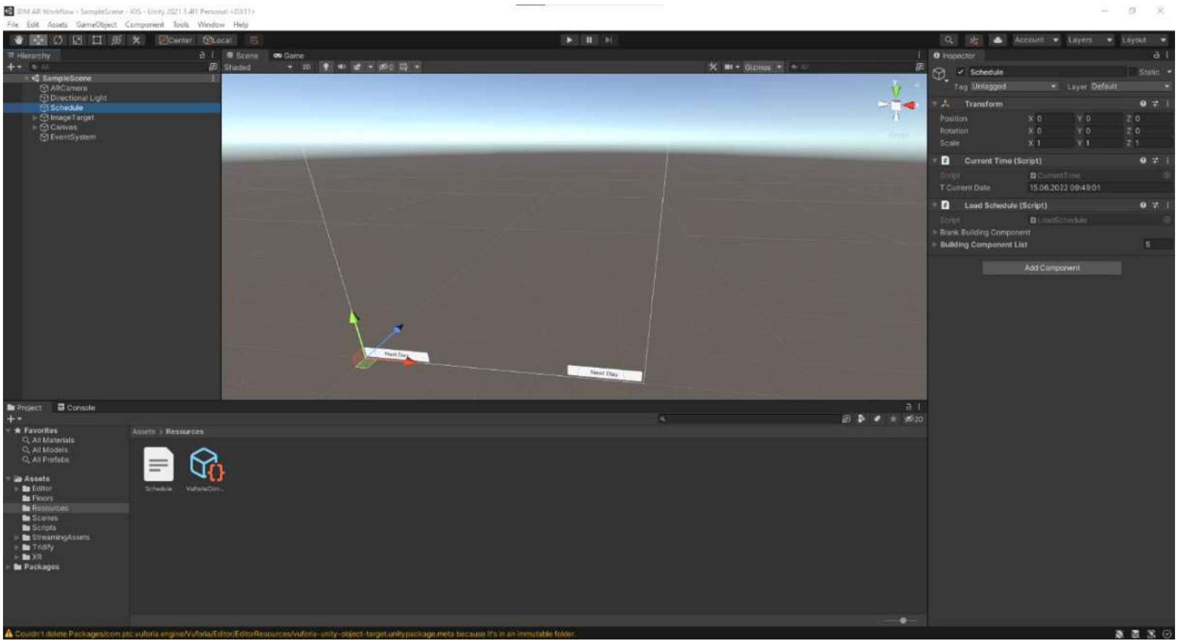


Figure 33: Unity User Interface

The differences between both applications are only seen when looking deeper into them. Unity is supporting programming languages like JavaScript and C# and visual programming is becoming more popular in the recent updates. UE has used visual programming for a longer time and supports C++ and Python. In the past Unity was always a little faster in implementing new VR- and AR-hardware then UE. The community of Unity is more active and bigger than UE´s, which can be a key factor for beginners. When choosing the development platform soft factors (pricing, hardware-support, community, etc.) and knowhow of the developers or their environment are the key factors.

In the following Unity will be further discussed and used for the project study.

3.4.2 Unity

Unity is a popular game engine which supports a wide range of AR-hardware and -tools. It is also very beginner friendly and has a big community. Also, a lot of learning material is available over their website "unity.com".

Figure 34: Unity Logo (source: unitiy.com)

Unity works with the “entity-component-model”. An object in the scene is an exact individual which stands in relationship to other objects. Objects in unity are called “GameObjects”, “Prefabs” or “Scripts”. Scripts hold code in C#. A GameObject is a collection of components like 3D-models, sounds, textures or also Scripts. A Prefab is a GameObject, which is specifically made to exchanges GameObjects between applications. In the creation of Scripts, mechanisms from software-development like inheritance can be used. Scripts can then be added to GameObjects via the UI and so their functionality can be extended.

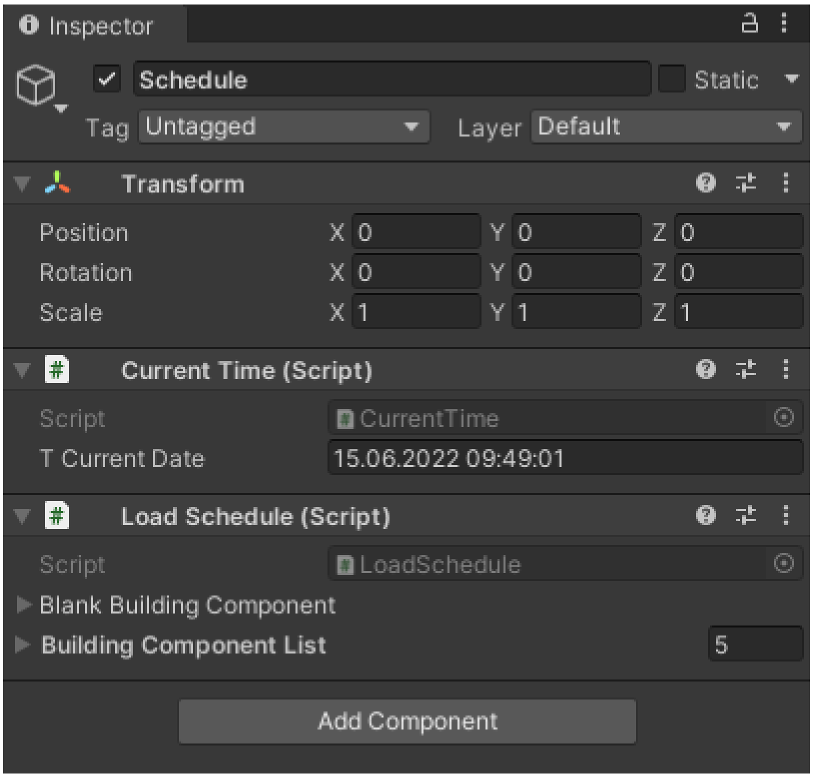


Figure 35: A GameObject in Unity with two added Scripts

The “Scene” in unity represents a level of an application and is pictured in the “Hierarchy”. The GameObjects in the Hierarchy can be arranged as the application design needs it. To manipulate sole GameObjects, Unity offers the “Inspector”-panel. It is shown in the figure above. In the Inspector components can be added to GameObjects and different Components can be altered. To get a preview of the application, Unity uses the “Game-View”, this can be accessed via the play button in the UI. When the Game-View is active, changes to the objects are not saved and lost when leaving it.

When starting a project in Unity there are different templets for different use-cases. A lot of software for the connection to other applications or specific hardware is preinstalled when choosing the right template. This makes the start with Unity very easy, but experienced users might install only the necessary addons for their project.

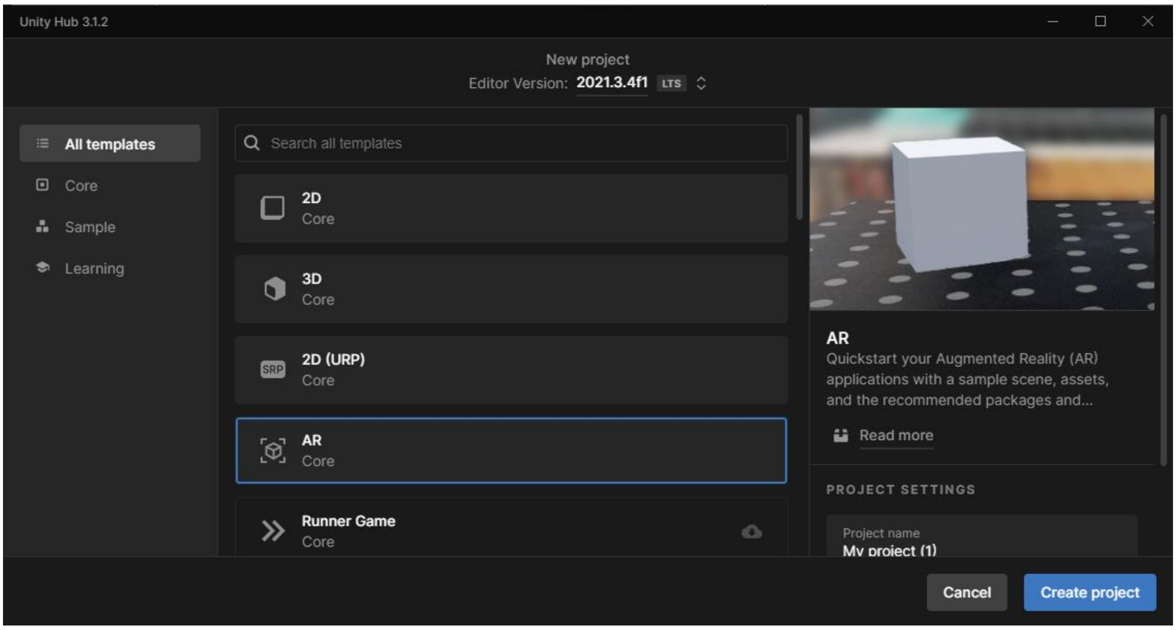


Figure 36: The creation of a new project and the different templates

3.4.3 AR-Devices [4]

Modern AR-devices are an assembly of a complex composition of different hardware. The invention of smartphones and their further development was the first big step to affordable, end-user AR-devices. Historically the “iPhone 3GS” must be mentioned, as it was the first iPhone with a magnetometer and therefore a digital compass. This led to a huge growth in AR-applications and innovation in this sector. Later the uprising of tablet-computers with a bigger screen like the “Samsung Galaxy Tab”-series or the “iPad” gave new possibilities.

The development of HMD also came a long way since Sutherlands “Head-Mounted Display System”. With more powerful hardware and sensors, the immersion of AR gets better and new use-cases can be developed.

In the following a few devices will be introduced to give an overview of the current state of AR-devices.

HMD and Smartglasses [4]

HMD and Smartglasses help to get a better immersion because the Field of View (FOV) can be much higher than when using a handheld device. Also, the user has the advantage of using a HMD or Smartglasses hands-free.

HMD and Smartglasses are normally different in their way of implementing AR. HMD base on a VST-AR approach, while Smartglasses normally have an OST-AR approach. The differences are discussed in 3.2.2.

First Smartglasses will be discussed and then other VST-HMDs will be presented.

<u>Smartglasses – OST-displays</u>

There are different function schemes of Smartglasses. The main difference is how the transport light from the display and light from the real world to the eye.

- Wave guide optics
- Prismatic glasses
- Mirror glasses
- Retinal glasses

Wave guide optics [4]

The principle is the same as with glass fiber. Light gets transported inside a glass-body and reflects on its boundaries. The key is how light ins transported in (coupled-in) and how it leaves the glass-body (coupled-out). For that, special optical elements have to be used which only lead the light into one way and others which lead it out in the direction of the user’s eye. Special elements for entering the glass-body are not needed if the light (from the display, which projects the virtual objects) is led in from the side of it.

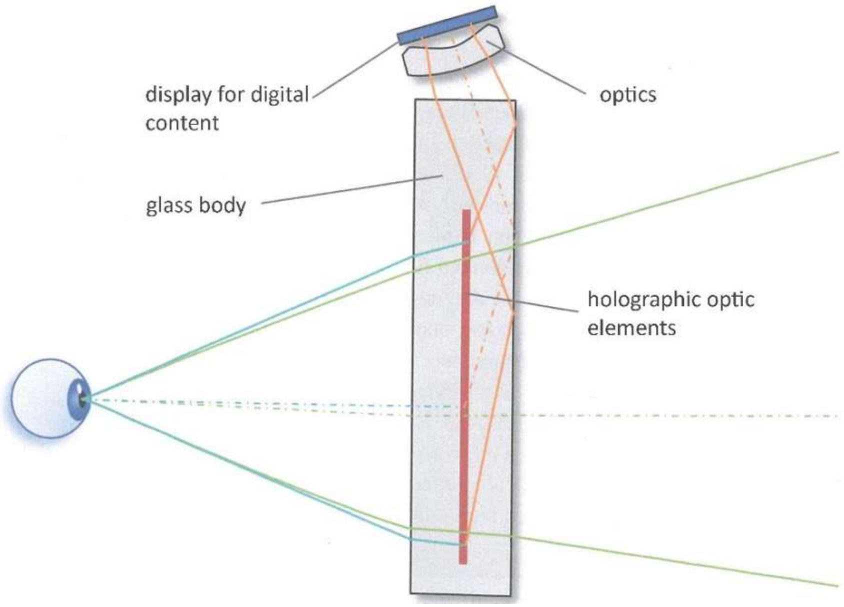


Figure 37: Schematic functionality of an OST-display based on light conductors, in this case a holographic optic element (Dörner et al., 2019)

With this technique, flat looking OST-HMDs, which resemble normal glasses can be achieved. To get a full-colored display three layers of glass-bodies have to be added together because its only possible to transport one color of the RGB spectrum at the same time.

Prismatic glasses [4]

Prismatic glasses have a relatively good light output for OST-devices with a compact construction. They also are used as VST displays, but to achieve a see-through effect, the prism gets completed to a glass body with parallel surfaces to let the light pass through.

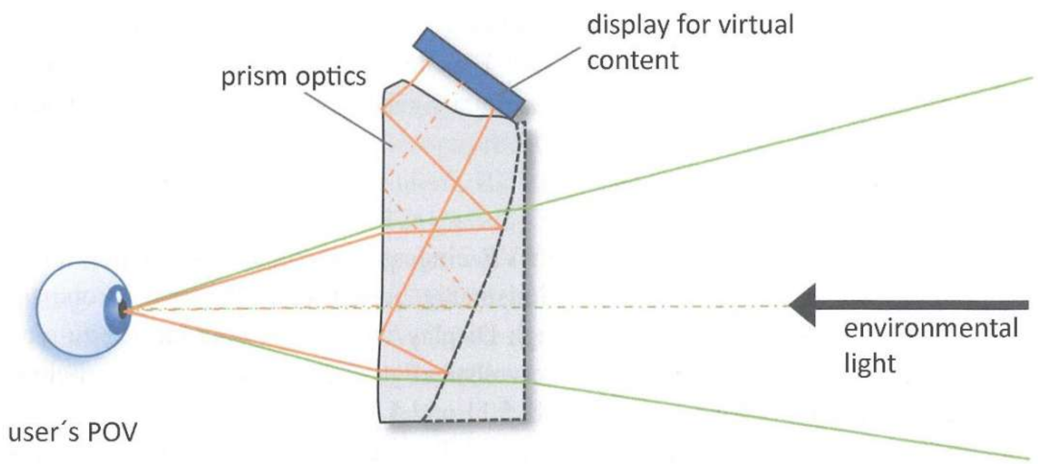


Figure 38: Schematic build of prismatic OST-glasses, consisting of two glass bodies. (Dörner et al., 2019)

The most known device with a construction method like this is the “Google Glass”.

Mirror glasses [4]

Mirror glasses use a semi-transparent mirror which reflects the virtual content onto another semi-transparent mirror. This content is displayed on a LCD- or OLED-display and gets optical enhanced to let the users eye focus on this content. Because the virtual and real content both pass the mirrors, the light is dampened before reaching the eye.

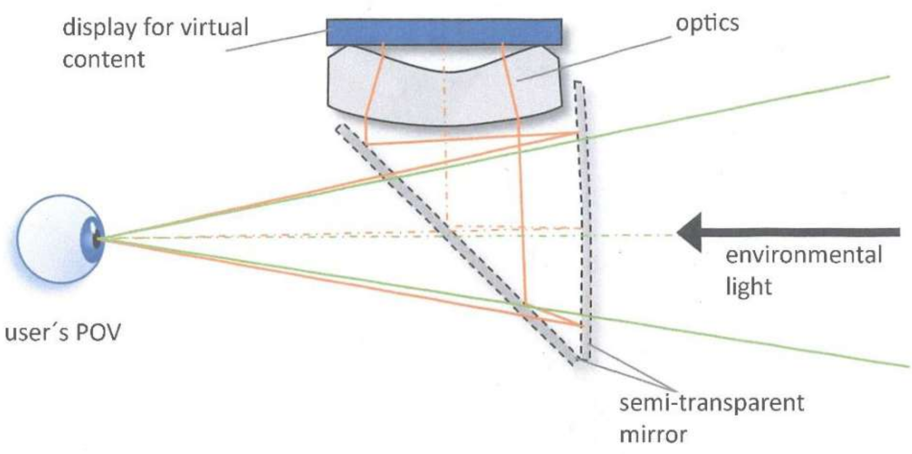


Figure 39: Schematic build of a mirror-based OST-display. (Dörner et al., 2019)

Retinal glasses [4]

With retinal HMD a display is non-existent, because the virtual content is projected directly onto the user´s retina. This method has two major advantages. First the removal of optical equipment and a very big field of view with a very small construction method. To project the virtual content directly onto the retina, modulated laser light gets reflected over a prism or a semi-transparent mirror. Multiple patents are held by apple and google for such glasses.

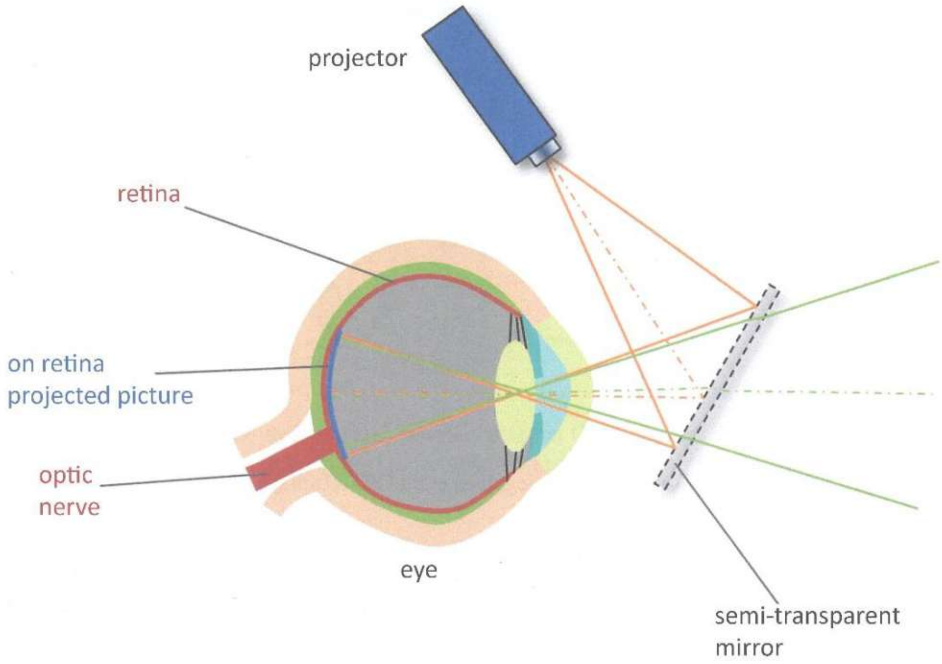


Figure 40: Schematic functionality of a retinal virtual OST-display. (Dörner et al., 2019)

<u>VST-displays</u>

When looking at VST-displays it can be said that most of them are built completely closed. This means that the user is at first disconnected from the outside, like when using a VR-headset. In difference to VR a video stream of the real world is added and the user has the feel of seeing through the glasses. For that VST-HMDs have one or two cameras built in. Since the human eye only sees the information projected on the screen, everything is at the same focus level. Also, the recording is at a lower quality, than looking directly onto the real objects.

Normally it is not possible to place the camera directly in the line of sight. Because of this, it is important to do a correction of the perspective. If that is not done, the user does have difficulties to assess size ratios and distances.

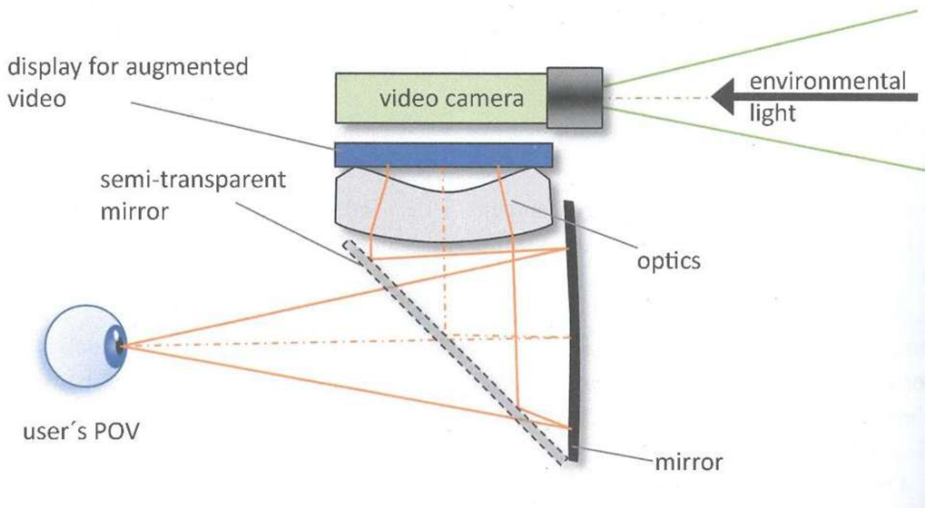


Figure 41: Schematic build of a mirror-based VST-display. (Dörner et al., 2019)

HMDs which direct the light, which normally would be in line of sight of the eye, to the camera or place the camera directly in the line of sight, do not have this problem.

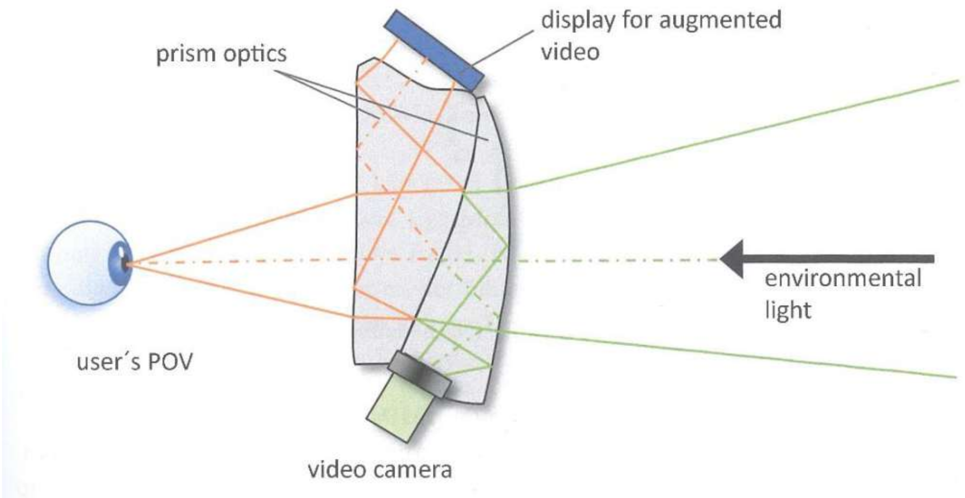


Figure 42: Schematic build of a prism-based VST-display with camera in line of site. (Dörner et al., 2019)

<u>Other differences of HMDs</u>

Apart of OST and VST-method, HMDs can differ in other categories too. In the following there is an overview of these.

Monocular and Binocular HMDs

Another way where HMDs can differ between each other is if they are monocular or binocular.

Monocular HMDs have one display and optical equipment for one eye and normally the second eye stays free. Binocular HMDs have the equipment for both eyes separately. With this a stereoscopic perception is possible. There are also binocular HMDs with one display, but two separate optical equipment sets. Here a stereoscopic perception is not possible. An example for that is a smartphone-based HMDs like “Google Cardboard”.

Open and Closed HMDs

While closed HMDs restrict the visual field of the user to the field of view of the device, open HMDs do not restrict the visual field outside of the display.

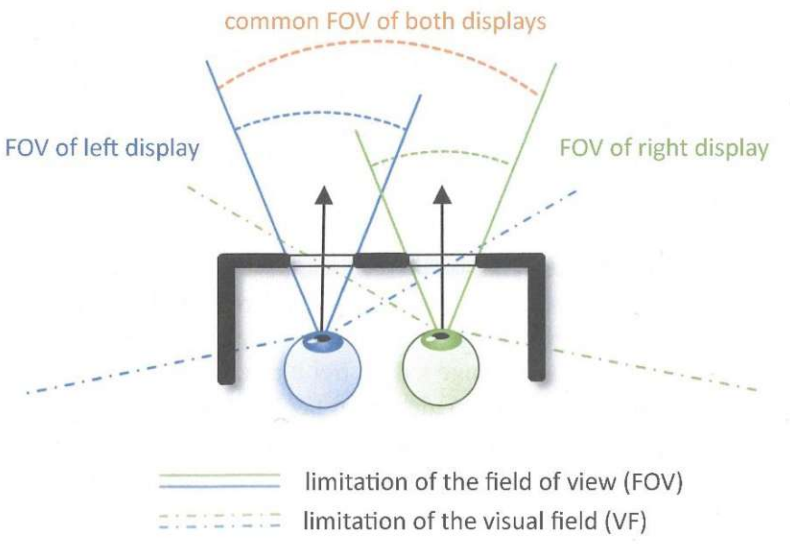


Figure 43: FOV of a binocular HMD, with closed building method in comparison to the user´s VF (Dörner et al., 2019)

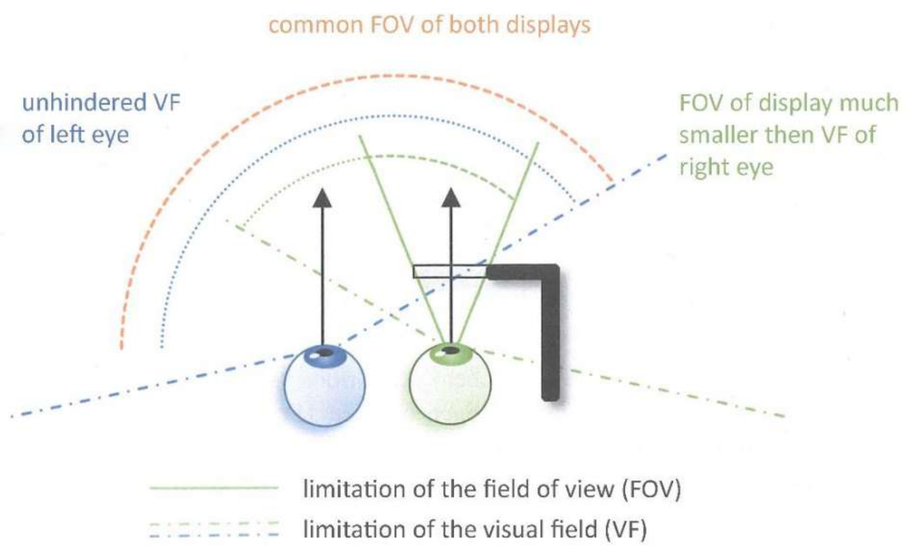


Figure 44: FOV of a monocular display on the right side with closed building method. (Dörner et al., 2019)

As shown in the figures above, the field of view is affected by the building method of the HMD. This means that the intersection between the field of view, which is the area where stereoscopic perception is possible can be quite limited. When this area becomes very small, it can lead to a tunnel view and motion sickness. Also, the peripheric seeing is hindered and the perception of surrounding objects can be restricted immensely. For this reason, it is very important that the device is chosen wisely for the respective use.

Monocular HMDs allow most of the time that one eye has at least unhindered sight of the environment.

Open HMDs give the user unhindered sight of the environment. Because of this fact, they are used in military use cases and work environments.

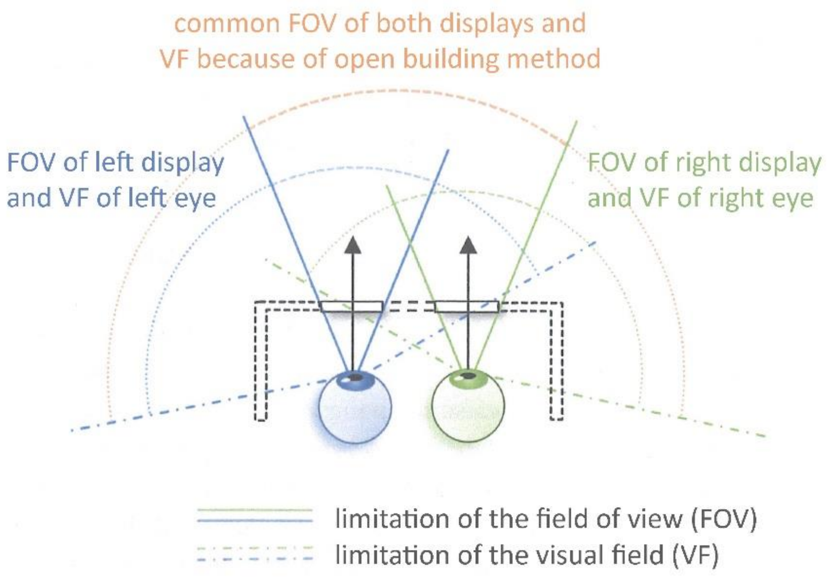


Figure 45: FOV of a binocular HMD with open building method. (Dörner et al., 2019)

They give unhindered peripheric sight, but the virtual content can only be shown in the field of view of the HMD. This can lead to cut off edges as shown in the following figure.

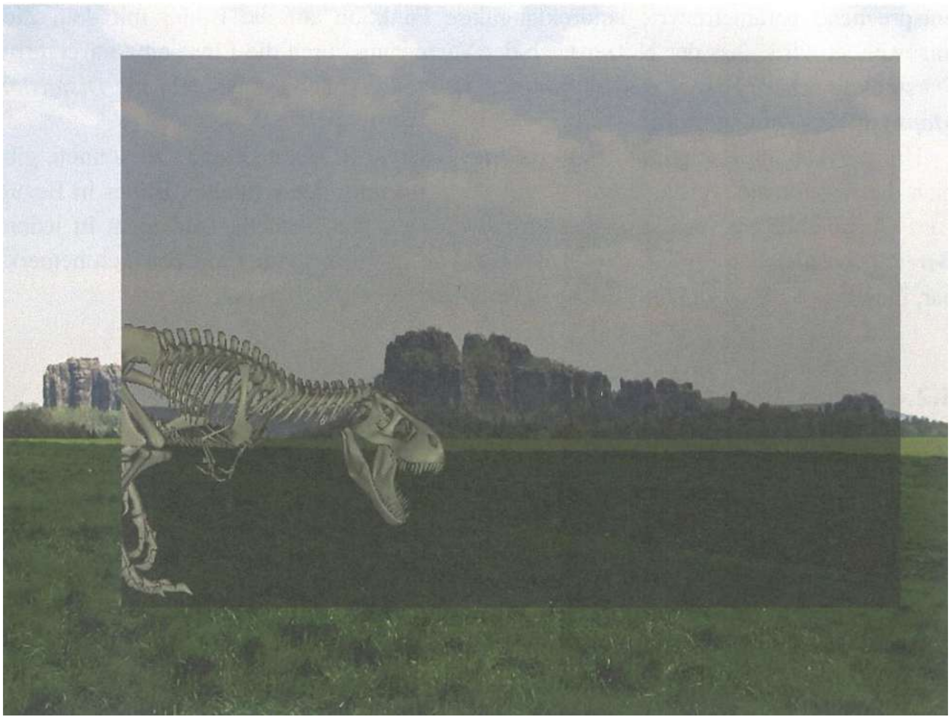


Figure 46: Problem with displaying virtual content on the edge of the user´s visual field. (Dörner et al., 2019)

Handheld- and Smartdevices

Because of the accessibility of AR-frameworks like ARCore for Android and ARKit for iOS, handheld and Smartdevices are one of the most popular devices for AR-development. They normally have a camera on the backside and other sensors too. They serve as recording- and optical-tracking-devices. [4]

As a VST-device the augmentation happens in the right perspective for the camera position, but not for the user´s position. This problem results as the camera´s field of view is always the same but not the user´s. That results in a lesser immersion than with HMDs.[4]

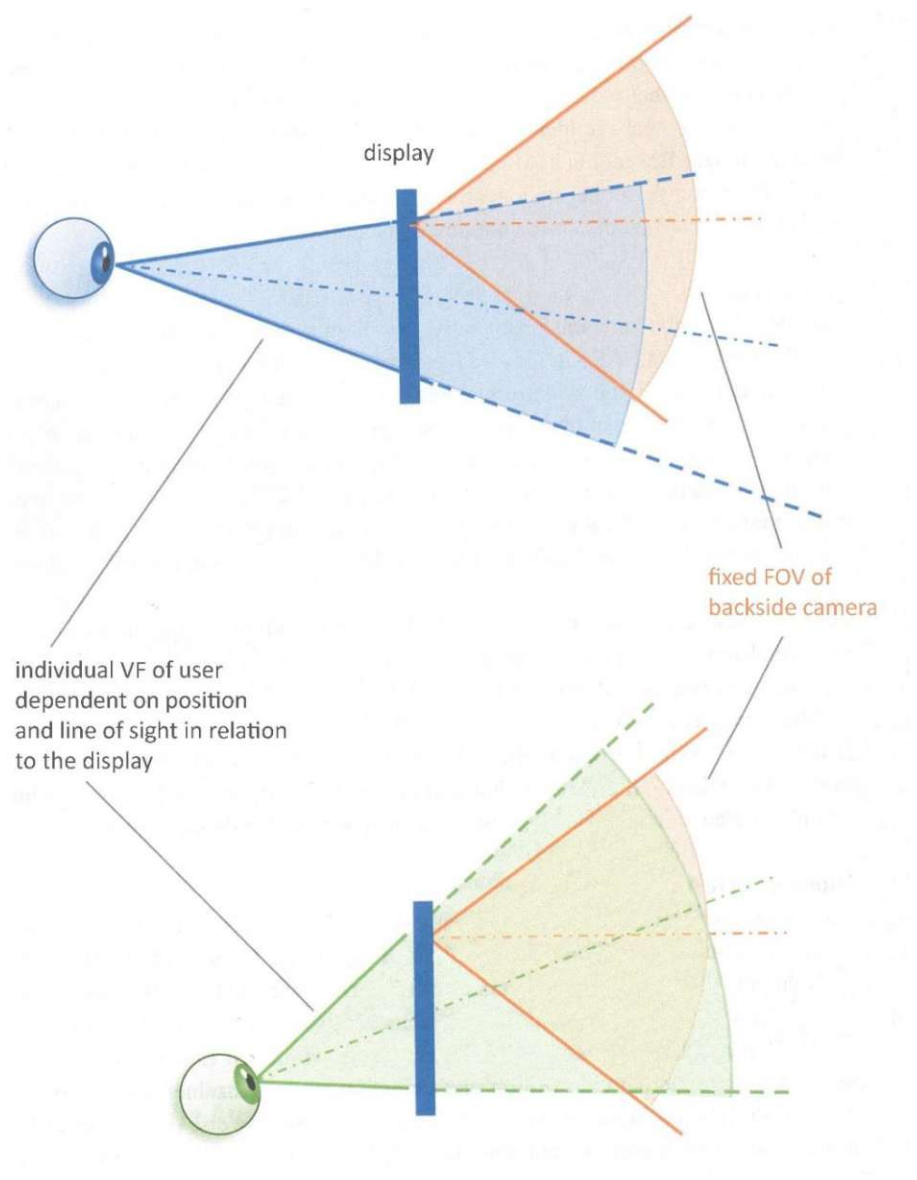


Figure 47: Difference of the field of view in comparison to the user´s position with handheld AR-devices. (Dörner et al., 2019)

Handheld devices are normally smartphones and tablets. Generally speaking, the hardware is equivalent over the different producers and are very wide spread in the end-customer-sector. Also, the end-customer has practically always a handheld device with him, which gives a huge platform for applications and their regular use.

In the following a few handheld devices of the two biggest companies (market share) will be introduced and compared. The principle of typical handheld devices is that of a VST-AR-approach, described earlier. So, the necessary hardware components which will be compared are the following:

- Camera
- Display
- Additional Sensors

The specs of the listed products are gathered from the respective websites of the companies.

First two smartphones will be introduced and then two tablet computers.

Apple – iPhone

Apple was the first company which developed a smartphone, which got wide spread at the end-customer sector. The first model launched in 2007 and the third generation of the phone was the first one with a magnometer built in, which made it a digital compass. There are now 38 different models released and in the following the high-end model iPhone 14 Pro Max will be presented.

Figure 48: Apple iPhone 14 pro max (source: www.apple.com)

- **Camera**

  Multiple Rare Cameras (48MP + 12.0MP + 12.0MP + 12.0 MP)
- **Display**

  Size: 6,7" – Resolution: 2796x1290 (approx. 460ppi)
- **Sensors**

  Accelerometer, Barometer, Gyro Sensor, Magnometer, Light Sensor, Proximity Sensor, LiDAR Sensor

Samsung – Galaxy

The Samsung Galaxy series is the flagship series of Samsung. Thy have multiple phones from entry-level models to high end models. There are foldable models, which have the advantage that the display size is can be nearly doubled if wanted. Here we look at the high-end model Samsung Galaxy S22 Ultra.

Figure 49: Samsung Galaxy S22 Series. (source: www.samsung.com)

- **Camera**

  Multiple Rare Cameras (108MP + 10MP + 12MP + 10MP)
- **Display**

  Size: 6,8" – Resolution: 3088x1440 (approx. 500ppi)
- **Sensors**

  Accelerometer, Barometer, Gyro Sensor, Geomagnetic Sensor, Hall Sensor, Light Sensor, Proximity Sensor

Apple – iPad

The first tablet introduced by apple was the iPad in 2010. Since then more different versions like iPad mini, iPad Air and the flagship iPad Pro where developed. At first the iPad had the same operating system then the iPhone (iOS), but today it has its own iPadOS. The main difference is the bigger display and better processors. In the following the model from 2022 of the iPad Pro 12,9" will be presented.

Figure 50: iPad Pro (source: www.apple.com)

- **Camera**

  Multiple Rare Cameras (10MP + 10MP)
- **Display**

  Size: 12,9" – Resolution: 2732x2048 (approx. 264ppi)
- **Sensors**

  LiDAR Sensor, Gyro Sensor, Accelerometer, Barometer, Light Sensor

Samsung – Galaxy Tab

The Samsung Galaxy Tab series got presented in 2010 and is running on an Android operating system. There were over 25 different versions released since then. In the following the Samsung Galaxy Tab S8 Ultra will be presented.

Figure 51: Samsung Galaxy Tab S8 Ultra (source: www.samsung.com)

- **Camera**

  Multiple Rare Cameras (13MP + 6MP)

- **Display**

  Size: 14,6" – Resolution: 2960x1848 (approx. 239ppi)

- **Sensors**

  Gyro Sensor, Accelerometer, Barometer, Light Sensor, Hall Sensor, Geomagnetic Sensor

When looking at the specs of the products above it can be said that smartphones have in comparison to tablets, better cameras and more sensors which can be used for AR.

Most of the sensors of both manufacturers are on the same level, but the apple products have both built in a LiDAR sensor, which can lead to better visual tracking.

When comparing the two smartphones it can be said that Samsung has a lead on display resolution and camera resolution. But the missing LiDAR sensor might give Apple a lead overall.

4.0 Project Study

In the following project study, some of the methods above will be used and tested to create an AR-application for mobile phones, which will implement a BIM-model and a building schedule to create a step-by-step augmentation of the building process.

This BIM-model is exported from a BIM-platform as IFC-file (see 2.2.2) and then is connected to a simple building-schedule. The schedule is implemented as CSV-file (Comma Separated Values), exported from Excel or other spreadsheet-programs. CSVs are text-documents where the data is separated via commas or semicolons and it is a common format for data exchange.

By combining these two elements an enhanced building model with additional information from the CSV is created and can then be augmented. An application which provides a step by step, three-dimensional guide on the building process should be the result.

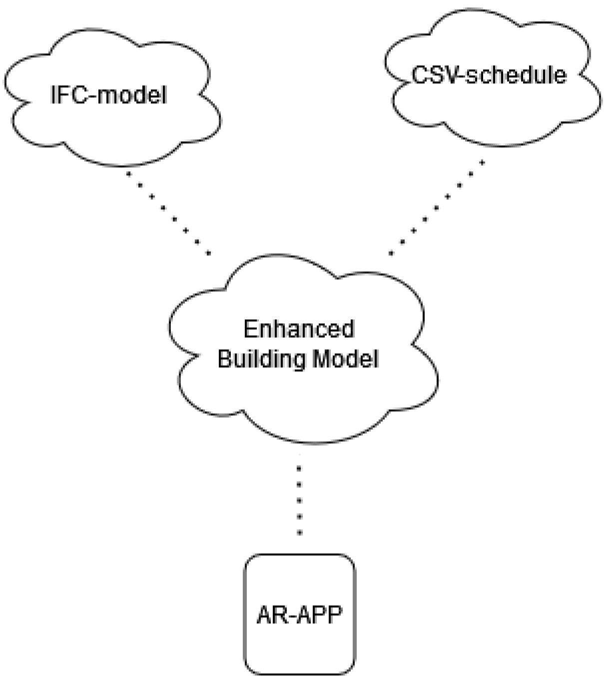


Figure 52: Basic Goal of the Study

In the subsequent chapter the basic workflow will be developed and then two projects will be introduced which will act as test-projects for the usage of the application. The workflow will be adjusted if needed and the application will be tested on stability and feasibility.

4.1 Test 0 – Workflow Development

In the following chapter the basic workflow in this study, will be introduced and thereafter every step will be annotated. For developing this workflow, basic structures like walls or columns were used and exported to test the interaction of the different file-formats with the corresponding programs. After some general tests this basic workflow was developed:

1. Design a BIM-model in an Architecture-Software
2. Export model IFC-file
3. Create a simple schedule in Excel and add via CSV to the IFC-file and export enhanced building model
4. Import enhanced building model to AR-Development-Software (game engine)
5. Add current date to the application to show currently important parts of the 3D-model only (in the game engine)
6. Enable Augmentation with marker-based approach in the game engine
7. Create a basic UI for targeted device in game engine
8. Building AR-application for chosen device
9. Augmentation on handheld device

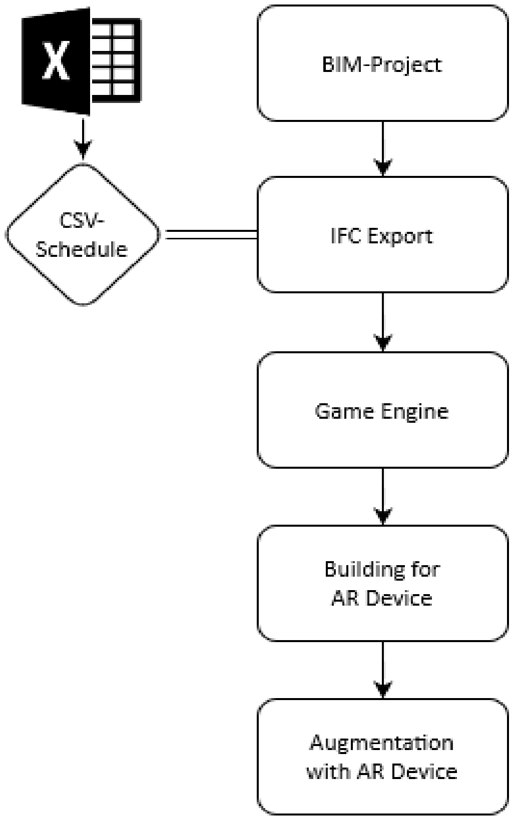


Figure 53: Basic Workflow

4.1.1 BIM Project

In the first instance, a BIM-model must be created and the chosen software for that is “Archicad”.

Archicad is a BIM-software developed by the Hungarian company Graphisoft and it is used by architects, designers, and builders to create and manage building information throughout the design, construction, and maintenance phases of a project.[17]

Also, other BIM-software, like “Revit” by Autodesk, would be a viable option and is used on a broad spectrum in AEC-companies.

Within the process of designing the BIM-model, all building elements are created, categorized, and labeled with their assigned Element-IDs.

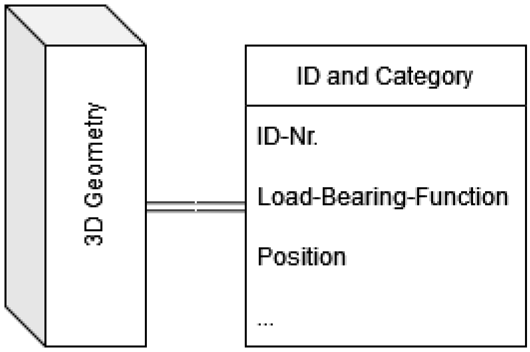


Figure 54: Schematic depiction of the classification of a building-component in Archicad

4.1.2 IFC Export

The export of the BIM-model as IFC-file is directly conducted in Archicad. For the export the current state of the art, therefore the IFC4-standard is used.

By exporting the BIM-model as an IFC-file, the information about geometry and other ascribed information is exported in form of so called IFC-property-sets. These appear as text-files and an extract from the IFC-file of the "Building Blocks"-Project, described later on, is shown below. The blue marked section mirrors the description of the column "Stütze-001". Because of the codification of the IFC-file the name of the column appears as "St\X2\00FC\X0\tze-001".

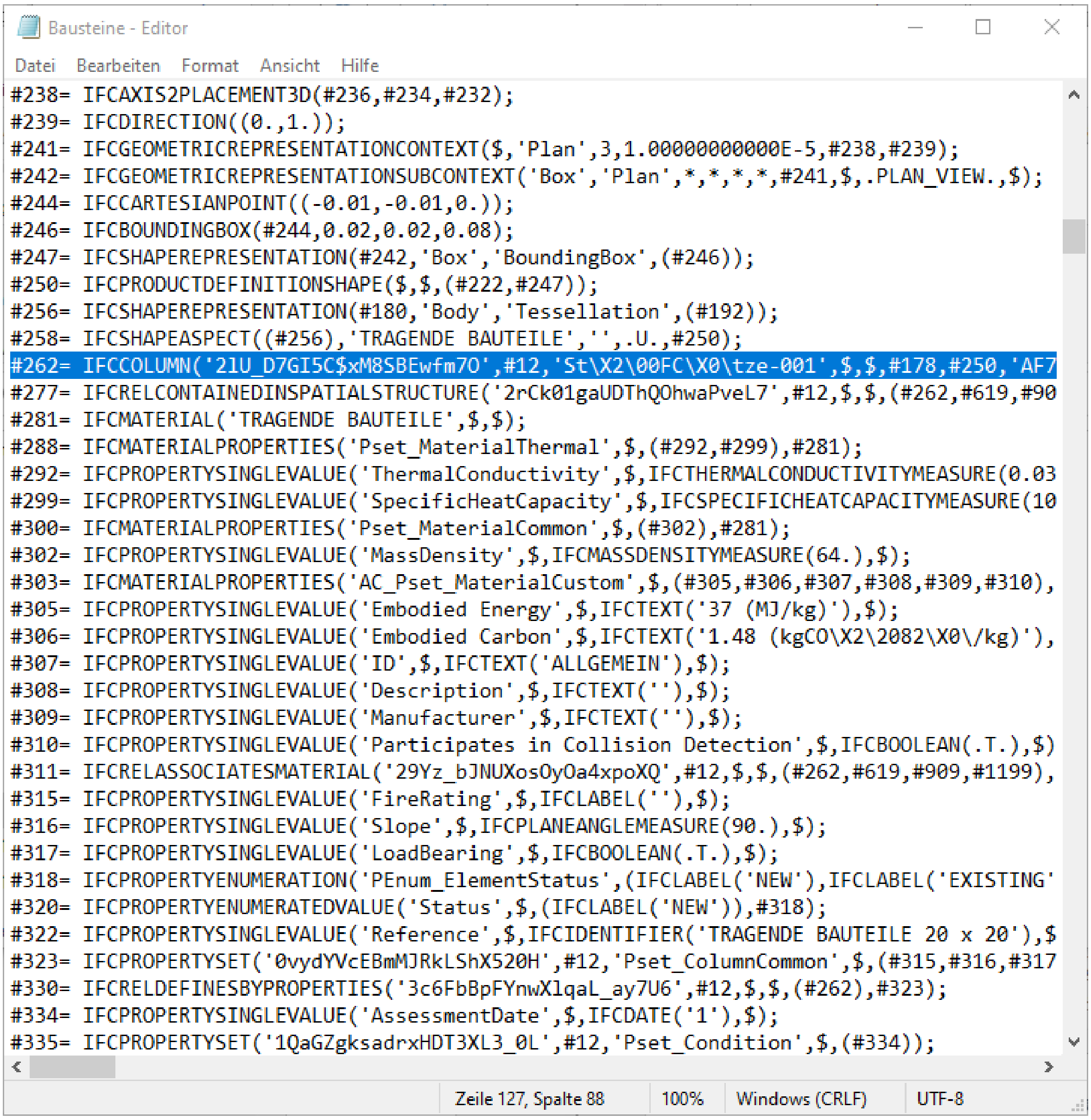


Figure 55: Part of the "Building Blocks"-IFC-file in a text editor

4.1.3 Adding a simple schedule to the IFC

The goal of the application is an easy to understand, three-dimensional presentation of a building-model, created by the combination of a BIM-model and the desired schedule of its construction. The first attempt to add the schedule to the IFC file was to combine the schedule and the IFC-file outside of the game engine “Unity” and only afterwards, import the enhanced file to the engine to create the AR-application.

In this case, the programs ASTA-Powerproject and Navisworks were tested to combine schedule and file. The import of the IFC was possible with both programs and the creation of a schedule and combination with the IFC was also simple. Sadly, there was no way to export the IFC with the added information. There is a recently put forward Plug-In for

Navisworks that should solve this issue, but after testing it, the additional information about the schedule was still lost with the export. The result was the same IFC as imported into Navisworks.

Another alternative to Navisworks or ASTA-Powerproject would be the software “Vico” from Trimble (https://vicooffice.dk), which is a modular project and risk management tool. One of the modules of this program is for 4D-management (3D+time) and provides simulations like the programs above, but the export of 4D-data is not described on their website and unfortunately there was no way to test the software.

After testing these different software-prospects, the original plan was altered and the combination of schedule and IFC was successfully conducted in Unity later in the workflow.

4.1.4 Import IFC to AR-Development-Platform

The chosen AR-Development-Platform is Unity.

Unity is a cross-platform game engine which started in 2005 and started as a desktop-only-engine and now supports a variety of desktop, mobile, console and virtual reality platforms.

At first, a Plug-in from the company “Tridify” was used to perform the importing of the IFC file into Unity. Unfortunately, this Plug-in is no longer available as of May 2023. Still, the workflow with Tridify is explained below since it might be available again soon. Afterwards, an updated workflow using an available and functioning paid Plug-in from the asset-store is provided.

Workflow with Tridify

Unity offers a great variety of plug-ins which are used for different purposes. One is provided by "Tridify". With the Tridify-plugin, it is possible to import all BIM-metadata which is provided by the export from Archicad.

Therefore, the IFC-file must be uploaded on the Tridify-website and the Plug-in must be installed and connected with Unity. The Plug-in allows to import the IFC-model into Unity with all its property-sets.

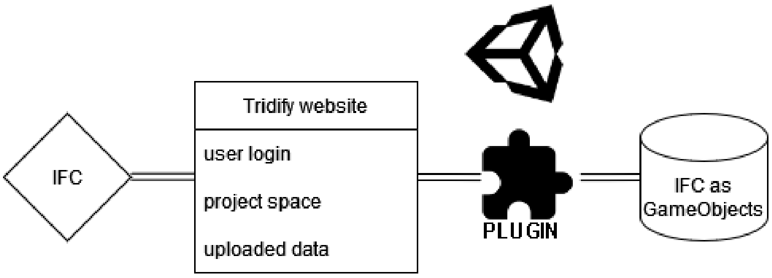


Figure 56: Workflow with Tridify

During testing an important part was, that the used Unity-version supports the “Used API Compatibility Level” “ .NET 4.x”, in order for the plugin to work.

The import of the IFC-file into Unity works via the Interface provided by the plugin. All imported data appears as GameObjects in Unity. These GameObjects hold scripts with the data from the IFC-property sets. This data cannot be altered, but new information can be added and recalled from the GameObject.

Workflow with IFC-importer from the asset-store

Since Tridify was no longer available, another tool was tested to import the IFC file into Unity, which can be downloaded from the Unity Asset Store.

The following instruction can also be found on the asset´s page and in its readme:

1. Download the plugin from the Unity Asset Store
2. Download IfcConvert.exe version for your operating system from https://ifcopenshell.org/downloads.html

3. Unzip the file you downloaded and copy the executable file “IfcConvert.exe” (Windows) or “IfcConvert” (Linux & Mac) to the IfcImporter folder in your project folder

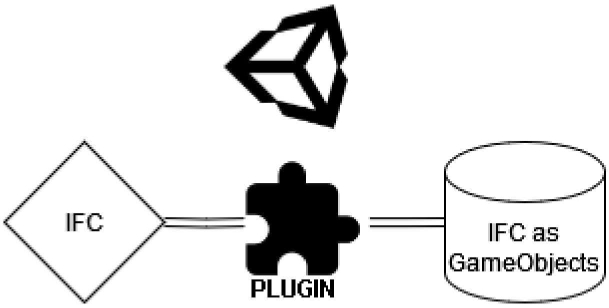


Figure 57: Workflow with IFC Importer

After installing the IFC Importer, it is possible to simply drag and drop the IFC file into Unity. The conversion to a Prefab then automatically starts. The prefab contains all 3D- and IFC meta-data. It also adds so called MeshRenderer and a MeshCollider to the objects.

A MeshRenderer is needed to render objects in Unity. Without a MeshRenderer the elements would appear invisible and therefore they would not be useable.

A MeshCollider tracks collision during runtime in Unity. This will not be needed in this case, but it is worth to mention it for further research.

4.1.5 Add a simple schedule in Unity

A simple schedule, including the Element-ID of each object being built and the start and desired end date of their construction, is created in Excel and then exported as a CSV file as mentioned in the above section of the thesis. In this project study, the smallest time skip will be a day, this is due to simplification and should be changed when building a ready to use application.

An important finding within that process was that the exported file must be saved with the UTF-8 codification, when using letters like "Ä" and "Ü" in the Element-IDs, which is common in the Austrian Archicad version.

The import of the CSV to Unity is achieved via drag and drop into the asset-folder. This CSV-file can be assigned to a GameObject with the help of code (in this case a C#-script named LoadSchedule). This script gets bound to a new created, empty GameObject in Unity. This GameObject got named “Schedule” and stores the data from the CSV file.

From “Schedule” the timestamps can be distributed to the imported IFC-data via another C#-script (AddSchedule). Although to make the script work, it is important that it only tries to add information to objects with Element-IDs mentioned in the schedule. Otherwise, an error occurs and the script stops working mid execution.

For that the script is written in a way, that it only distributes data to GameObjects tagged as “BimData”. This custom tag is added to all imported GameObjects from the IFC-file.

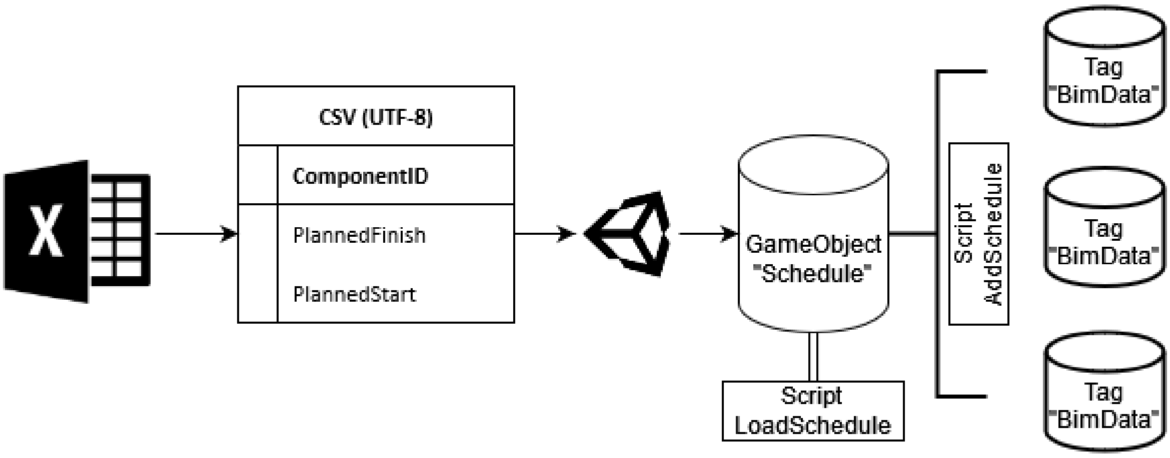


Figure 58: schematic distribution of CSV-data to GameObjects from IFC-import

4.1.6 Add time to application to show currently important parts of the 3D-model only

A time-dimension must be added to the application, in order to only show the parts of the 3D-model that should currently be worked on. For that, the function DateTime is used, used from the "System" namespace. With this function it is possible to calculate with dates. For example, it is possible to compare two specific dates and tell which one is before the other.

Unfortunately, the Unity inspector (a feature that shows detailed information about the currently selected GameObject), does not support DateTime. Hence, the DateTime will only be shown as a string in a specific format (dd.MM.yyy HH:mm). To ensure the functioning of this, the same format must be used in the CSV-file.

With that, the string can be parsed to DateTime format and more operations like <, >, =, etc. are possible.

First it was tested to deactivate the GameObjects, which should not be shown, but it was not possible to access them once they were deactivated. To solve this problem, GameObjects would not be deactivated, but the MeshRenderer component of the object, so they will not be rendered. So, all GameObjects can still be accessed, even when they are currently not rendered.

To get a current timestamp, the “Schedule”-GameObject gets added a script (CurrentTime). This script also holds functions to alter the current timestamp. This enables to compare the current timestamp with the dates from the schedule and alter the MeshRenderer component of the GameObjects.

In following depiction, the MeshRenderer component of the GameObjects with the tag “BimData” are activated when their PlannedStart<=CurrentDate and deactivated when their PlannedStart>CurrentDate.

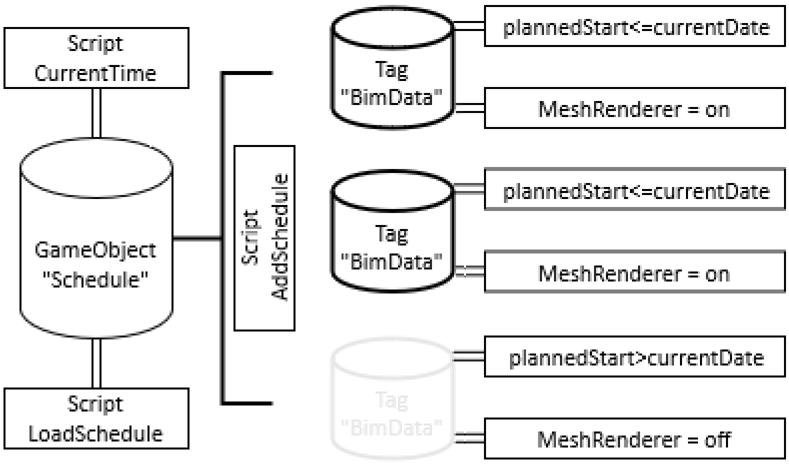


Figure 59: schematic depiction of the basic functionality

With this functionality only elements that are already built or currently under construction are shown.

4.1.7 Augmentation with marker

To start the augmentation in Unity, a Plug-in from the company "Vuforia" must be implemented. A description of the implementation follows.

Firstly, the latest version of GIT, a control system that tracks changes in files, must be installed. The Vuforia Plug-in must be added to Unity and the Unity Package for Vuforia must be downloaded and added as well. Next, a registration on the "Vuforia"-website is necessary to activate a license. From the so called "Target Manager", databases can be added that feature different kinds of markers or targets. A logo was created which will be prospectively used as trackable marker.

Figure 60: Marker used as ImageTarget for augmentation. 10x10cm

Vuforia describes some guidelines that should be followed when creating such markers – these can be found on their website.

The created logo is uploaded on the Vuforia "Target Manager" and can be accessed with the Vuforia Plug-in in Unity. The Vuforia Plug-in gives access to some special features. The two most important ones are the ARCamera and the ImageTarget - both are used for augmentation in this project study.

The first step to start working with the created logo is to add an ARCamera to the hierarchy. It is then possible to access the Vuforia Engine configuration where a license key must be added, which is generated over the website.

In the next step, an "ImageTarget" gets added, it is important to choose "From Database" as its type. The database where the created marker is held must then be downloaded from the Vuforia website and implemented into Unity to be accessible for the ImageTarget. Databases can hold several targets, in this case, it only is the created logo.

Important to mention is that all the GameObjects, which hold the BIM-data, must be under the ImageTarget in the hierarchy and become so called children of the ImageTarget. Otherwise tracking from the marker is not initiated.

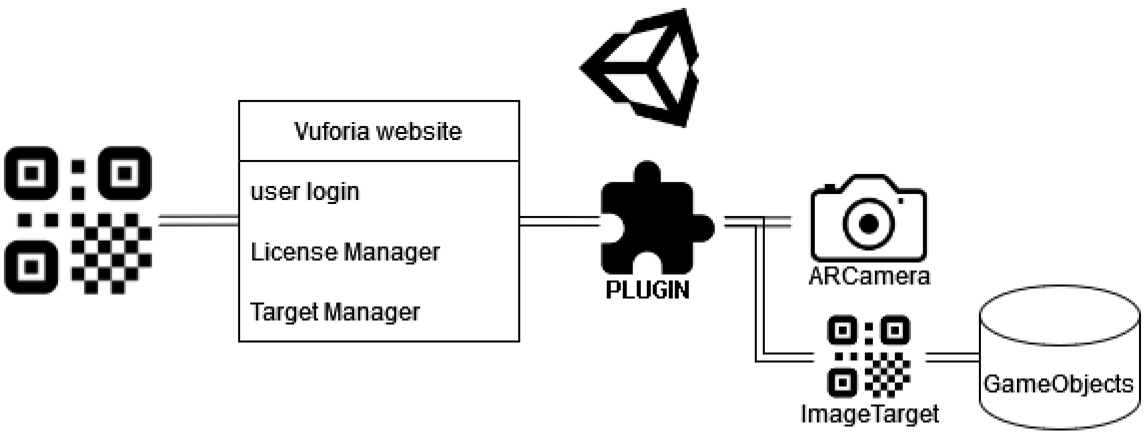


Figure 61: Workflow for Augmentation with Vuforia in game engine

With these settings, augmentation is now possible.

4.1.8 Basic UI for Handheld Device

To skip through different days and make the application interactive, a basic UI (User Interface) must be built in Unity. As mentioned above the smallest time skip in this project study will be a whole day.

The UI consists of two buttons to skip between days and a text that shows the selected time. For this a "Canvas", a "EventSystem" and the two button-GameObjects must be created. The Canvas is referencing the display of the chosen device, the EventSystem allows events, like clicking a button, to happen. The buttons get a link to the "Schedule"-GameObject to alter the time shown there. This altered time is also shown in the text in the middle.

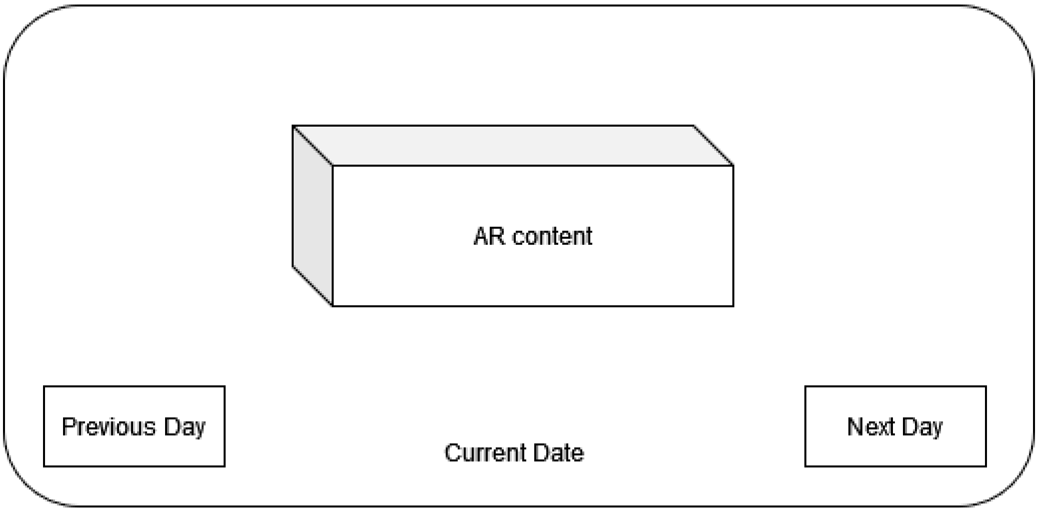


Figure 62: Basic UI for the AR-application

4.1.9 Building AR-Application for IOS

For this project study the chosen device for augmentation is an iPhone 14 Pro. This implies that the application must be built for IOS. To build for IOS, it is necessary to export a Xcode-project from Unity. Xcode is an integrated development environment (IDE) for all of Apple's platforms like iOS, iPadOS, tvOS, watchOS and macOS. It provides tools to design, develop, and publish applications. It is the only official tool for creating and publishing apps on the Apple App Store. Therefore, it must be used to publish for the iPhone 14 Pro. After developing the application in Unity, it can be directly exported for Xcode.

The exported Xcode-project must then be opened on an OS device. From there it can be published on the iOS device and tested.

4.1.10 Improved Workflow

After going through these steps and learning that the combination of the IFC and the

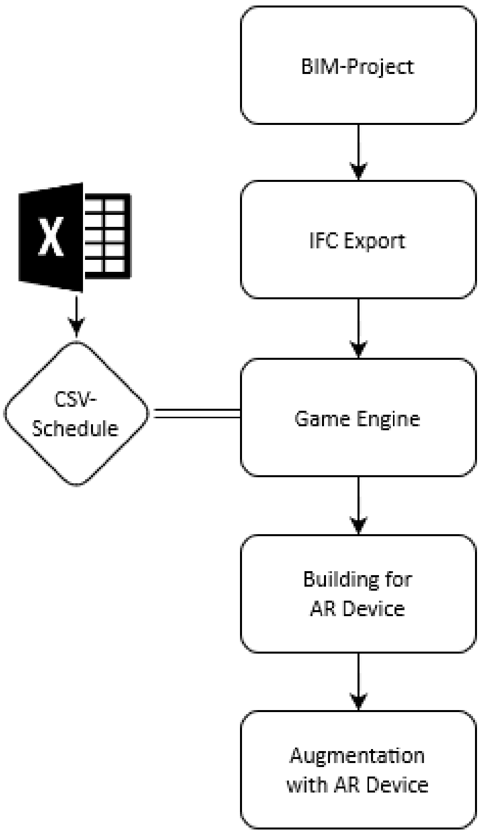


Figure 63: Altered Workflow

schedule is not that easily possible outside of Unity, the final workflow now looks like this:

4.1.11 Development Improvements

To provide a better workflow in the process of creating the application a custom GUI window in Unity can be created. With this GUI, the application does not always have to be tested in “Game-View” to validate small changes made to the code. This saves time, because the application does not need to load alle scripts and render all objects. In the end, this GUI, called “CustomEditorWindow” in this case, holds all scripts, which can be called by clicking a button.

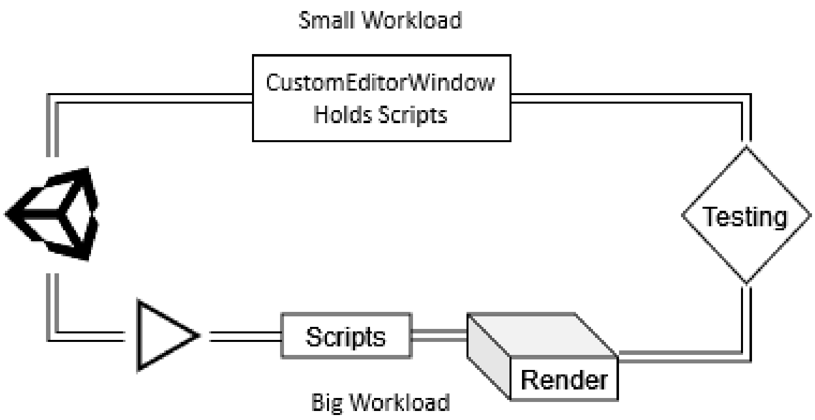


Figure 64: basic functionality of CustomEditorWindow

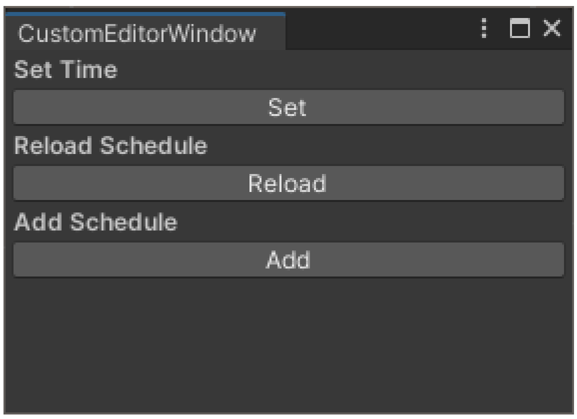


Figure 65: CustomEditorWindow in Unity

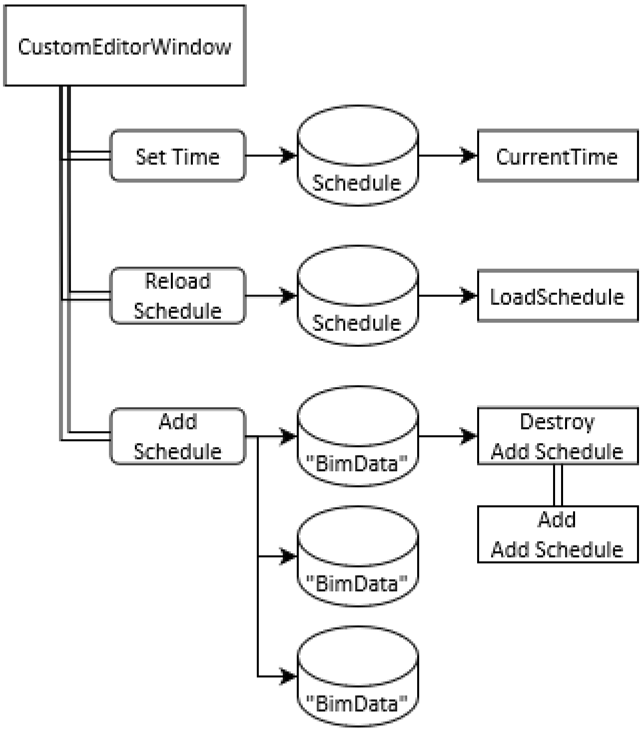


Figure 66: expanded functionality of CustomEditorWindow

4.2 Test 1 – Project "Building Blocks"

In the following the first project will be described and then used as test subject for the basic workflow above.

4.2.1 Project Description

The first project discussed in this study is a small-scale model with dimensions of 10x10x10cm, which is made of five individual elements. This model will be presented as a table-top-augmentation, meaning it has no relation to its surroundings and can therefore be presented on every table top. The model will be used to examine the development and test the workflow from the original BIM-model to an enhanced, augmented building model.

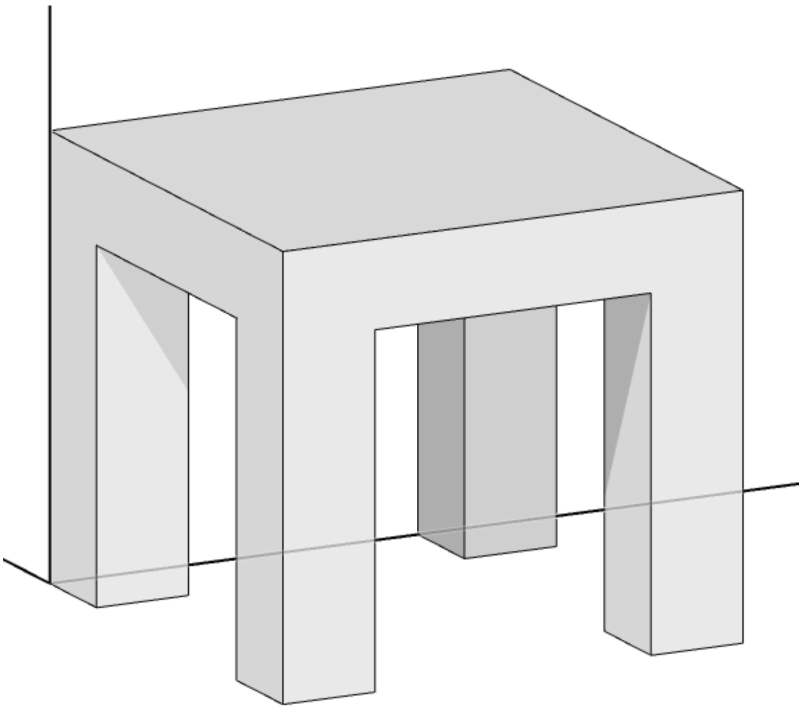


Figure 67: Isometric of the Project "Building Blocks"

The five elements of this structure are four columns and one slab. As, this is a double-symmetric structure, it is very important to be able to differentiate all elements from each other. This happens over the meta-data in every object´s settings. This means that every object gets a unique "Element-ID", which ensures the distinction. These "Element-IDs" are shown in the following drawings.

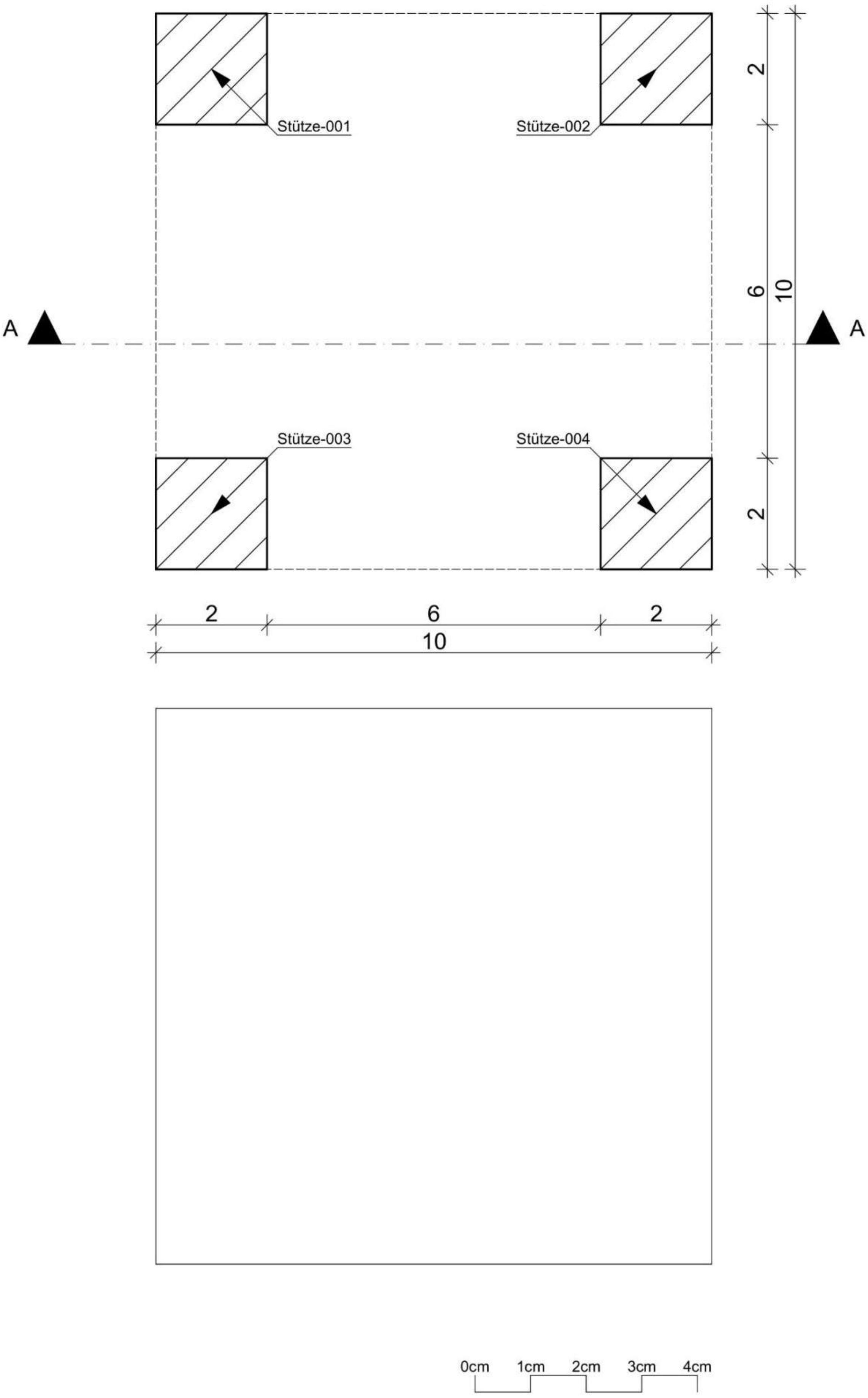


Figure 68: Floorplan (Top) and Top-view (bottom) M 1:1

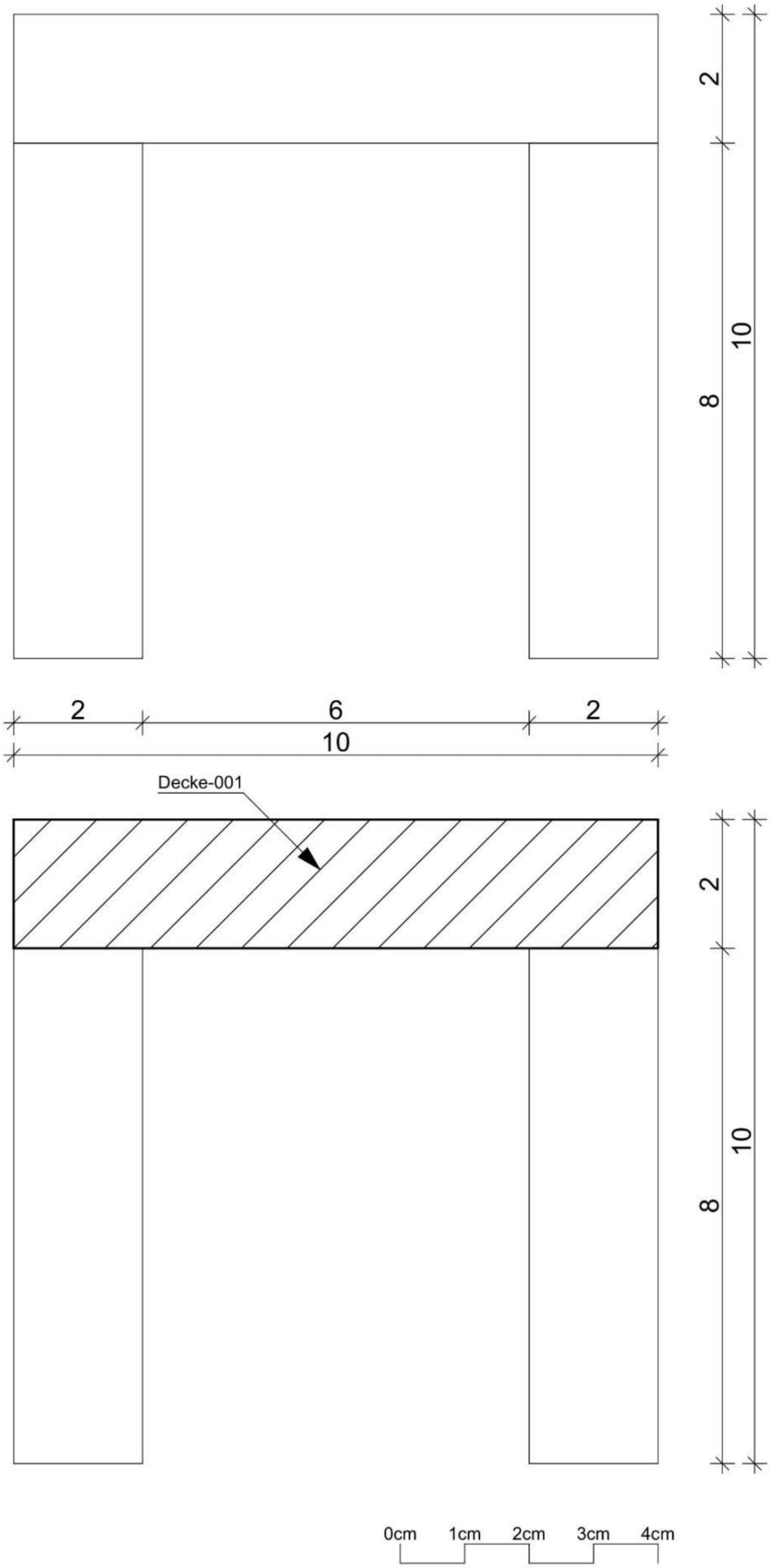


Figure 69: Elevation of all sides(top) (double symmetrical structure) and section A-A (bottom) M 1:1

Schedule

The schedule of the project is created with Excel and is then exported as CSV-file. In this project every building-component can be distinguished with the help of their Element-ID, which was assigned in the BIM-model.

The schedule determines a “PlannedStart” and a “PlannedFinish” for every object. These are the dates when the building of the object should be started and finished. In this project study there is always a skip of a whole day between objects, due to simplification. This data should later be distributed to each element of the structure to create the enhanced building model. The Element-ID ensures that this distribution is correct.

Element-ID	PlannedStart	PlannedFinish
Stütze-001	14.05.2023 08:00	14.05.2023 17:00
Stütze-002	15.05.2023 08:00	15.05.2023 17:00
Stütze-003	16.05.2023 08:00	16.05.2023 17:00
Stütze-004	17.05.2023 08:00	17.05.2023 17:00
Decke-001	18.05.2023 08:00	18.05.2023 17:00

Figure 70: The simple schedule in Excel for the project "Building Blocks"

In the following picture the Excel table is exported as CSV-file and opened in a text-editor program. All rows are separated in new horizontal lines and columns are separated via a semicolon in this case.

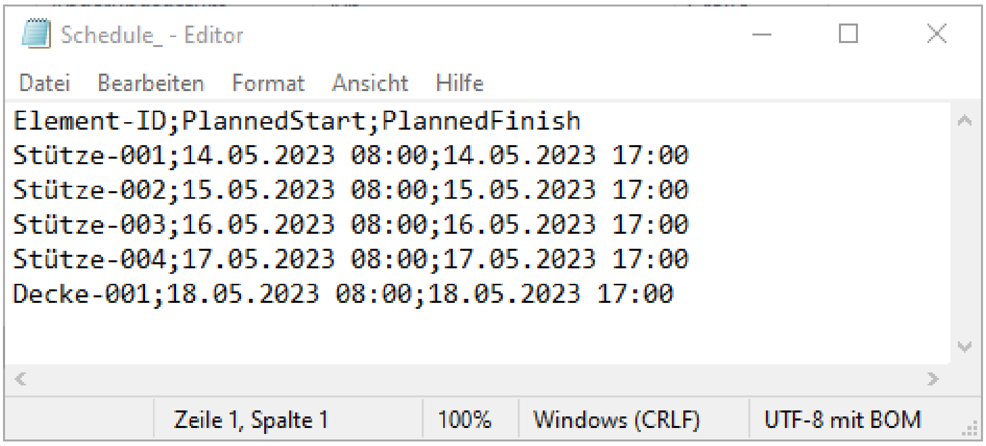


Figure 71: The exported CSV-schedule of the project "Building Blocks" in a text-editor

4.2.2 Workflow-Test for the Project "Building Blocks"

The acquired workflow will now be tested by augmenting the project "Building Bricks" as a table-top augmentation. This means it can be tested everywhere and has no relation to its surroundings, by just placing the marker on a flat surface.

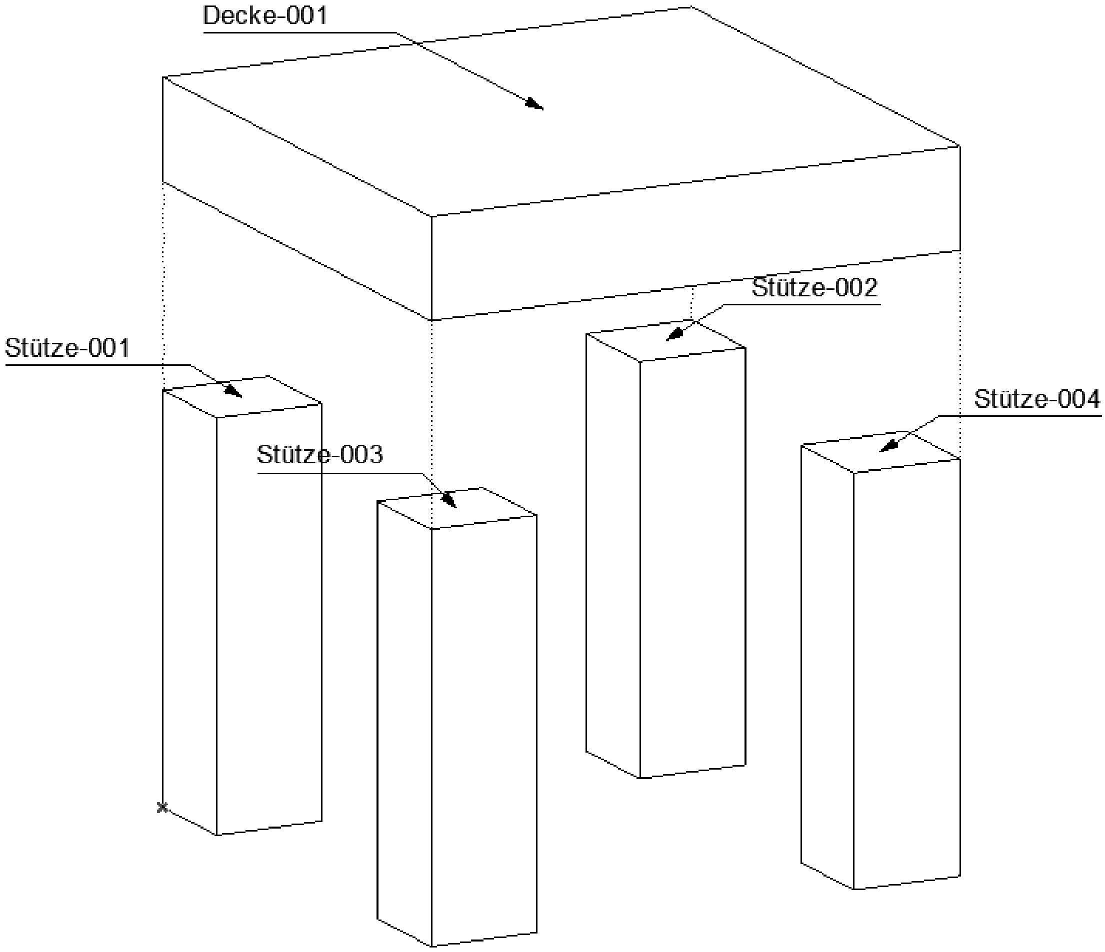


Figure 72: Exploded Axon Diagram of the project "Building Blocks"

BIM Project

The design-phase of the "Building Blocks"-project in Archicad is simple and straight forward. The structure consists of four columns and a slab.

IFC Export

Before exporting the model as an IFC-file, all five elements receive a unique Element-ID which is done via the IFC-properties. The columns will be called *Stütze-001*, *Stütze-002*, *Stütze-003*, *Stütze-004* and the slab is named *Decke-001*.

The current standard of IFC files "IFC4" is used to prevent any complications in the further process.

Import IFC to AR-Development-Platform

With the help of the bought asset, IFC-Importer, the import is a simple drag and drop task. After dropping the IFC into Unity the importer converts it into a Unity prefab. This prefab contains all metadata from the IFC and additional Unity-specific elements, which are described in 4.2.4. This prefab can be placed in the scene to take the next steps.

Adding a simple schedule to the IFC

This step also has nearly no complications. At first an Excel 2013 version was used and the export as UTF-8 was not that easy, but with an update to an Excel 2019 version a dedicated UTF-8 export is available without diving into more options.

The scripts work as intended and all information stored in the "Schedule"-GameObject is distributed to the respective building-components, as seen in the example of "Decke-001" below.

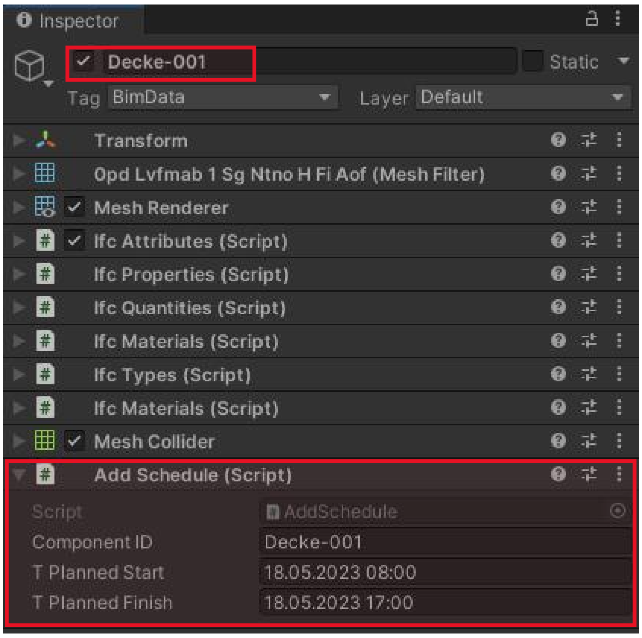


Figure 73: Added information from the schedule seen in the Inspector in Unity for the GameObject "Decke-001"

Add time to the application to show currently important parts of the 3D-model only

This process works as intended, but to make the it clearer what step of the building-process is next, a new feature was added. All parts which are new, will be rendered in red. This makes it easier to identify what step is next and what building-components are part of it. For this the PlannedStart- and PlannedFinish-component of the schedule are compared to the current date. If the PlannedFinish>=CurrentDate and PlannedStart<=CurrentDate the object is then rendered in red. The other cases are described in the following figure.

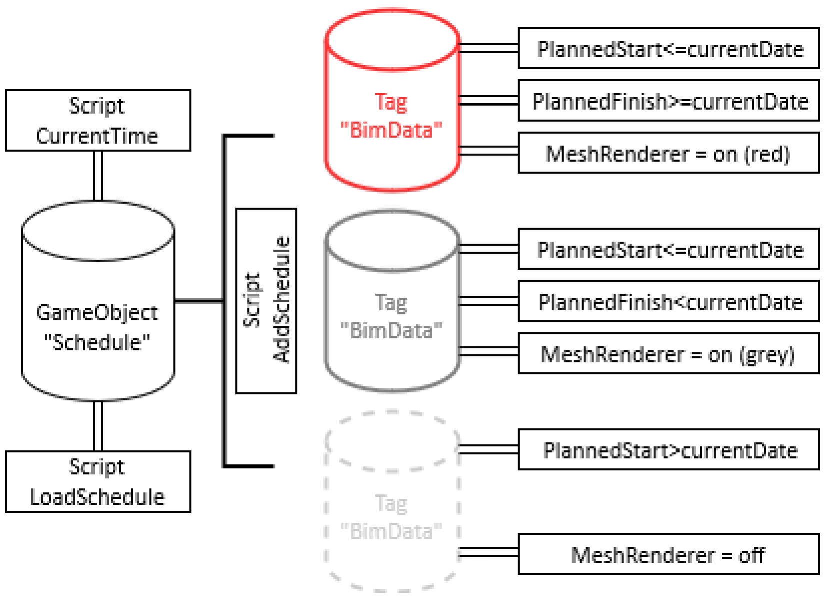


Figure 74: Updated scripts to show important objects more clearly

To give a better understanding about this functionality and the sequence of events, the schedule, and a series of drawings for the project "Building Blocks" are depicted below.

Element-ID	PlannedStart	PlannedFinish
Stütze-001	14.05.2023 08:00	14.05.2023 17:00
Stütze-002	15.05.2023 08:00	15.05.2023 17:00
Stütze-003	16.05.2023 08:00	16.05.2023 17:00
Stütze-004	17.05.2023 08:00	17.05.2023 17:00
Decke-001	18.05.2023 08:00	18.05.2023 17:00

Figure 75: Schedule of the project "Building Blocks"

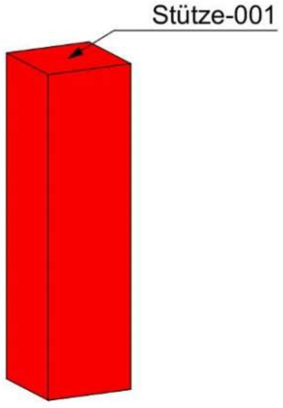


Figure 76: Step 1 of the process for the project "Building Blocks"

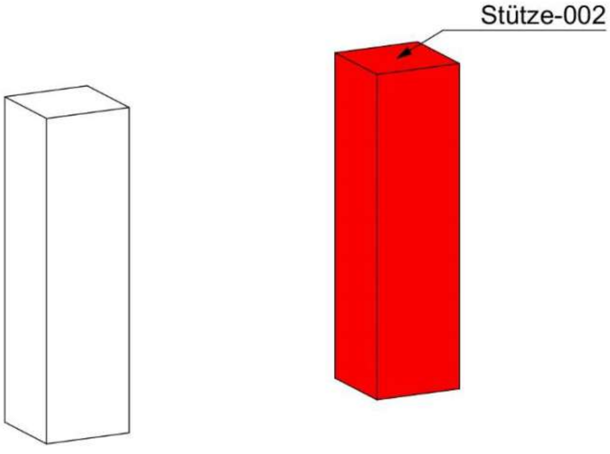


Figure 77: Step 2 of the process for the project "Building Blocks"

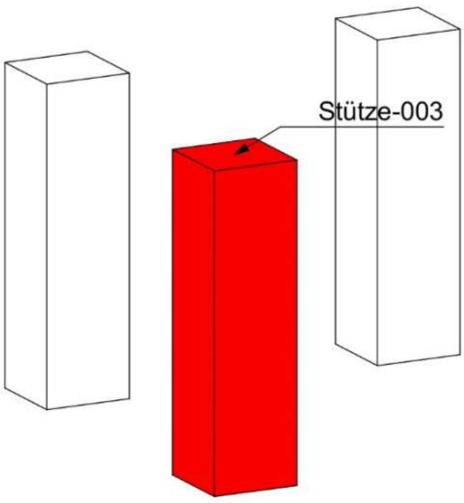


Figure 78: Step 3 of the process for the project "Building Blocks"

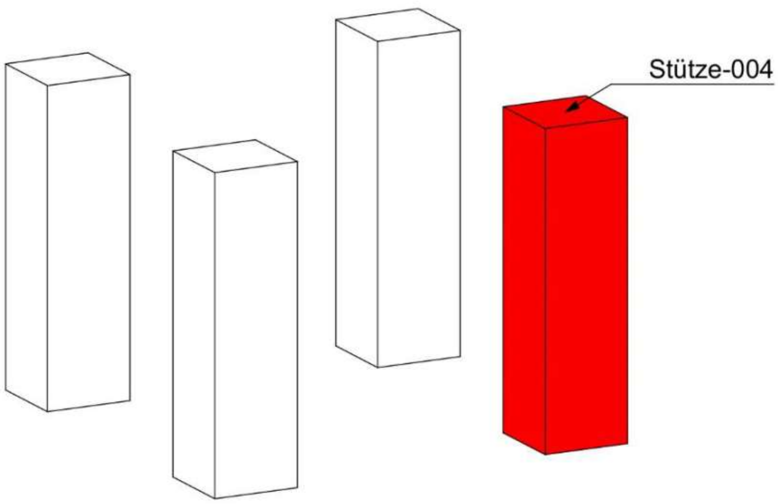


Figure 79: Step 4 of the process for the project "Building Blocks"

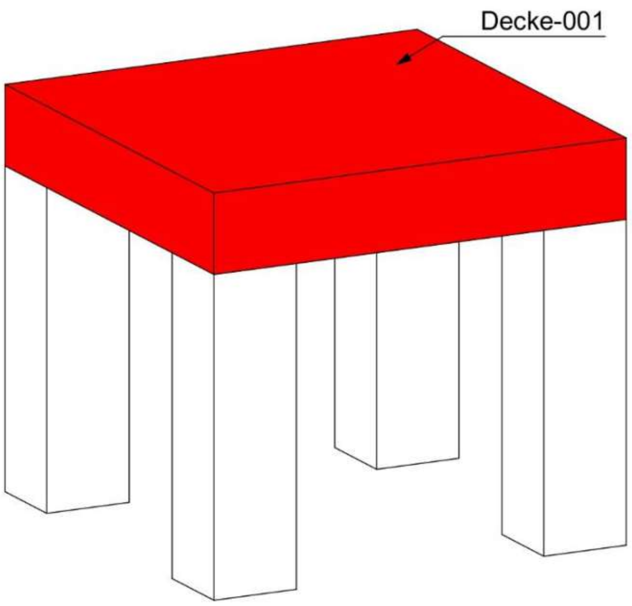


Figure 80: Step 5 of the process for the project "Building Blocks"

Augmentation with marker

As described above, Vuforia was added in Unity and the Vuforia-specific GameObjects like the ARCamera and ImageTarget are utilized.

Even though augmentation will be initialized with a single marker in this scenario, the tracking is not solely dependable on this marker. This is due to Vuforia´s technology called "Vuforia Fusion".

Fusion solves the problem of fragmentation in AR-enabling technologies. It utilizes cameras, sensors, chipsets, and software frameworks like ARKit or ARCore.[17]

In this case it utilizes the sensors and camera of an iPhone 14 Pro and ARKit, Apple´s AR-framework. With these techniques like SLAM or VSLAM are used to stabilize the AR objects after initial positioning through the marker.

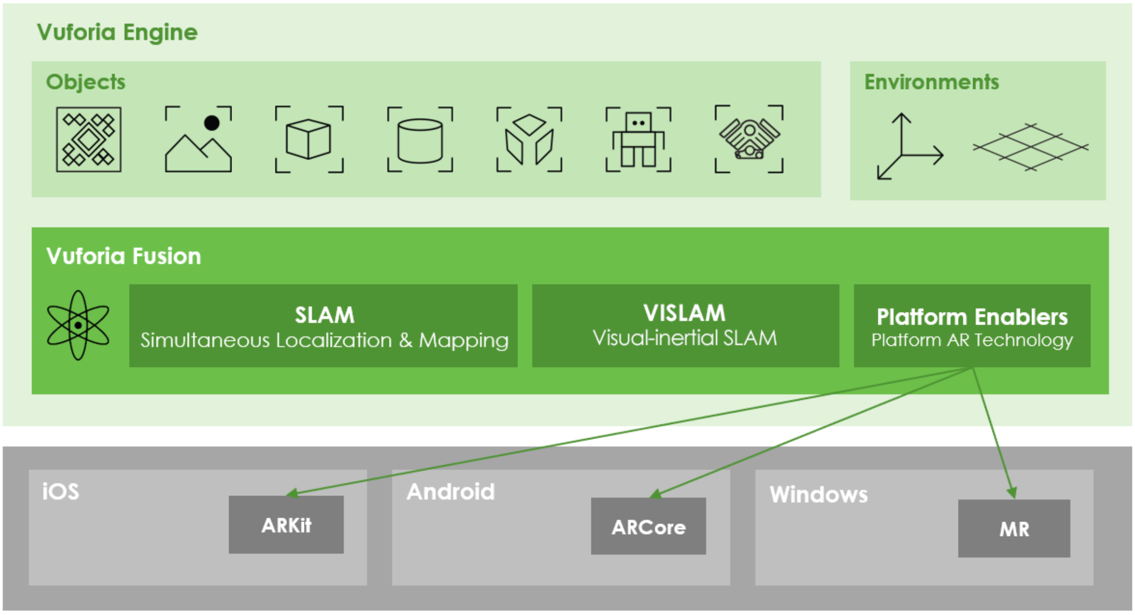


Figure 81: Vuforia Fusion Overview (source: www.vuforia.com)

A detailed explanation of Vuforia Fusion is found on their website.

Building AR application for IOS

The process of building the AR application for IOS functioned as described above, therefore the file was exported as a Xcode-project from Unity and was then opened on an OS device.

Augmentation of small project, as table-top-augmentation

After the application is published on the device it can be tested. Therefore, the model of "Building Bricks" was used, which is 10x10x10cm on its outer boarders.

After scanning the ImageTarget in reality, the augmented model is shown at the same position, which was predefined in Unity. Initial tracking is therefore functioning, but the whole project is shown in red. This is due to a programming error.

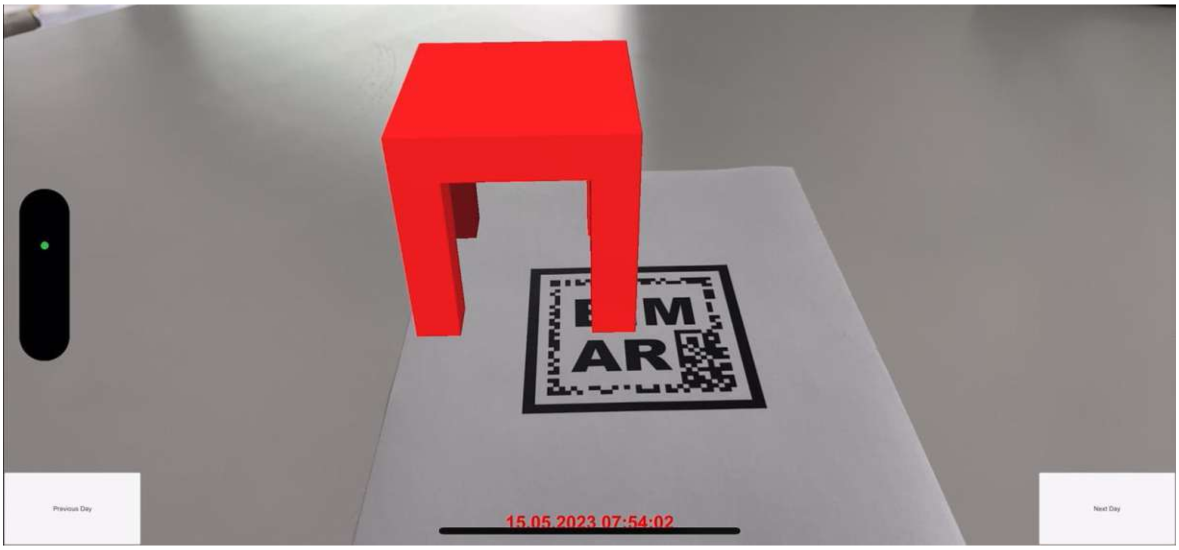


Figure 82: Test1 - initialized model shows in all red

After clicking the UI-button "NextDay" the date shifts to next day as intended and the building bricks are shown like they are supposed to.

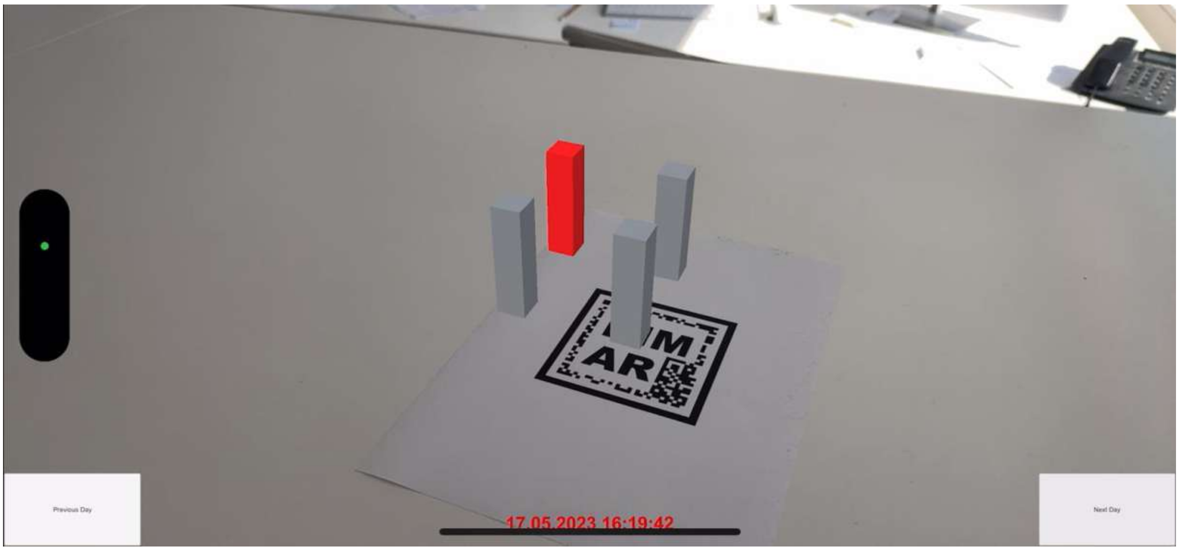


Figure 83: Test1 - when skipping through steps, the color is as intended

When moving the camera away from the marker, augmented content is still correctly depicted, as shown in the following figure. This is an indicator that Vuforia Fusion is working and the tracking is not only based on the marker.

Figure 84: Test1 - when the marker is not visible in the video stream, the model still is stable and working as intended

Results of the first test

All in all, this first test showed a promising result. The augmentation of the 10x10x10cm big project initialized as expected at the position predefined in Unity and its size seemed to be depicted correctly. The DateTime function, together with the UI controls allowed to go back and forth, to see the changes in the model at different timestamps. This shows that the information of the schedule was allocated to the GameObjects of the IFC-import, otherwise the model would not be able to interact along the displayed time. Only the initial visualization of the whole model in red was not intended and needs to be fixed.

Vuforia Fusion seems to work properly, this is indicated by the model not disappearing, when the marker is covered in the video stream (see Figure 113)

To test the application further, the bigger project "Garden Shed" will now be approached.

4.3 Test 2 – Project "Garden Shed"

In the following the second project will be described and then used as test subject.

4.3.1 Project Description

The second project used in this project study is a garden shed designed for a residential building in Wels, Upper Austria.

Due to delays, parts of the structure are already built and seen in the following pictures. Unfortunately, the site was not photographed before the construction started. But this also has a positive impact for the testing-phase, since deviations between augmented parts and already built parts are seen promptly.

Figure 85: Location of the project "Garden Shed"

Building site and Requirements for Design

The building site is an open space behind the driveway of the building. The main entrance of the building is also located along this driveway.

In this chapter pictures of the site will be shown and beforehand a map of the camera-positions is included for better understanding. The pictures are marked with the letter P+NUMBER (e.g., P1, P2, P3, …). A small symbol for orientation and the index is shown on the map.

Figure 86: Plan for orientation of Photos. M 1:200

Figure 87: P1 - Entrance to the driveway

Figure 88: P2 - Main entrance for residents, located in the driveway

The shed will be used to provide covered space for bikes of the residents, garden supplies and garbage containers. When designing the project, it was important that as much space as possible between existing building and new built shed remains. Two cars should still be able to park and drive in and out of the property.

Figure 89: P3 - Space between building and shed. On the left already built parts of the shed are seen.

Because of a change in level of the terrain between driveway and the lower located garden, the shed must be supported by two cantilevers to provide the needed space between building and shed.

Figure 90: P4 - Change in level of the terrain – two cantilevers support the structure.

Figure 91: P5 - change in terrain. The picture is taken from basement-level.

As already seen in the photographs above, it is important to mention that there is a big tree after the fall in terrain, which must remain and is only trimmed under the shed.

Therefore, the floor and roof of the shed will have an outlet for this tree.

Figure 92: P6 - Substructure for the grating in the area of the tree.

Another desired use of the shed is, that it serves as privacy screen to the private garden below. Therefore, and because of the needed storage-space, the shed will cover the whole side of the driveway.

Figure 93: P7 - Sight from driveway down into the garden.

This proposition will require some form of fire protection on the side of the shed, because it is on the boarder to the neighboring property.

Figure 94: P8 - Sight down the boarder to the neighboring property.

Figure 95: Site plan of the project "Garden Shed" M1:200

Plans and Design Process

After analyzing the site, and surveying it, the design-process began.

With the challenges described above, the design was developed to match the requirements. The shed was moved partly over the edge of the driveway to ensure parking was still possible. This requires two cantilevers, which support the structure and conduct the bearing loads into the concrete retaining wall. The maximum length of these cantilevers is 1,23m. They also had to be supported by bracings to ensure enough static stiffness. Together with two screw-foundations, in axis B, this wall forms the foundation of the shed.

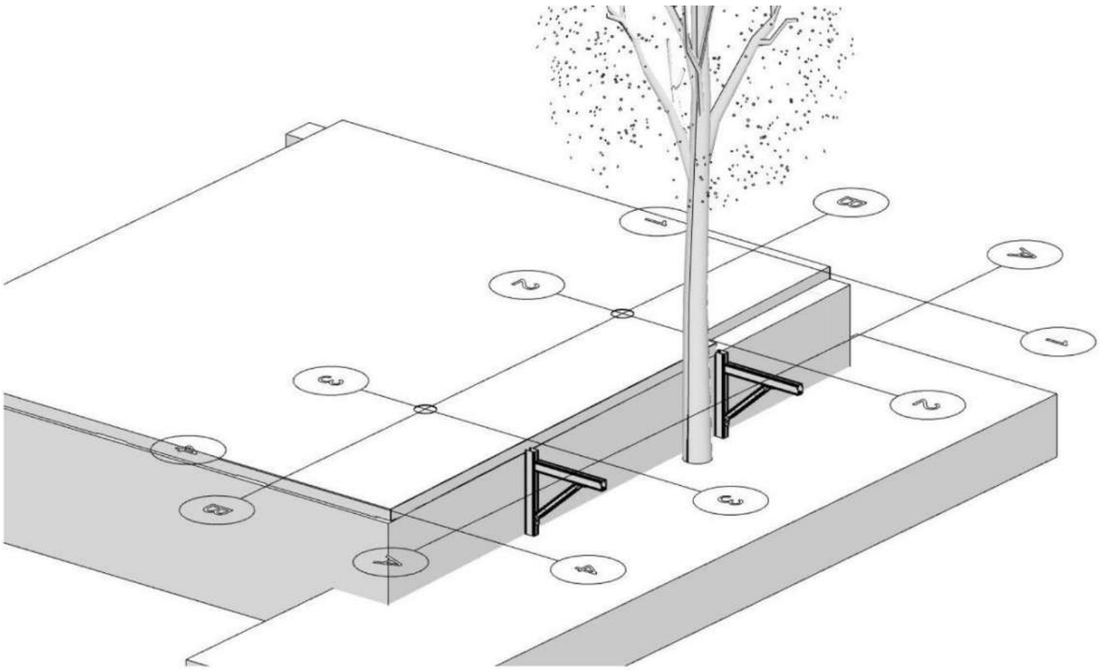


Figure 96: Depiction of the foundation for the shed

The main structure is built along two horizontal (A and B) and 4 vertical axes (1-4). Only one beam ("Träger-014" +" Träger-015"), which serves as support for the grating on the floor level, is not orthogonal to the axes, due to the already existent, inclined retaining wall.

Because of limitations in the production of the chosen building material, single elements cannot be longer than 6,00m. Therefore, some beams are coupled with a special connection. Later, the way of construction will be discussed in further detail.

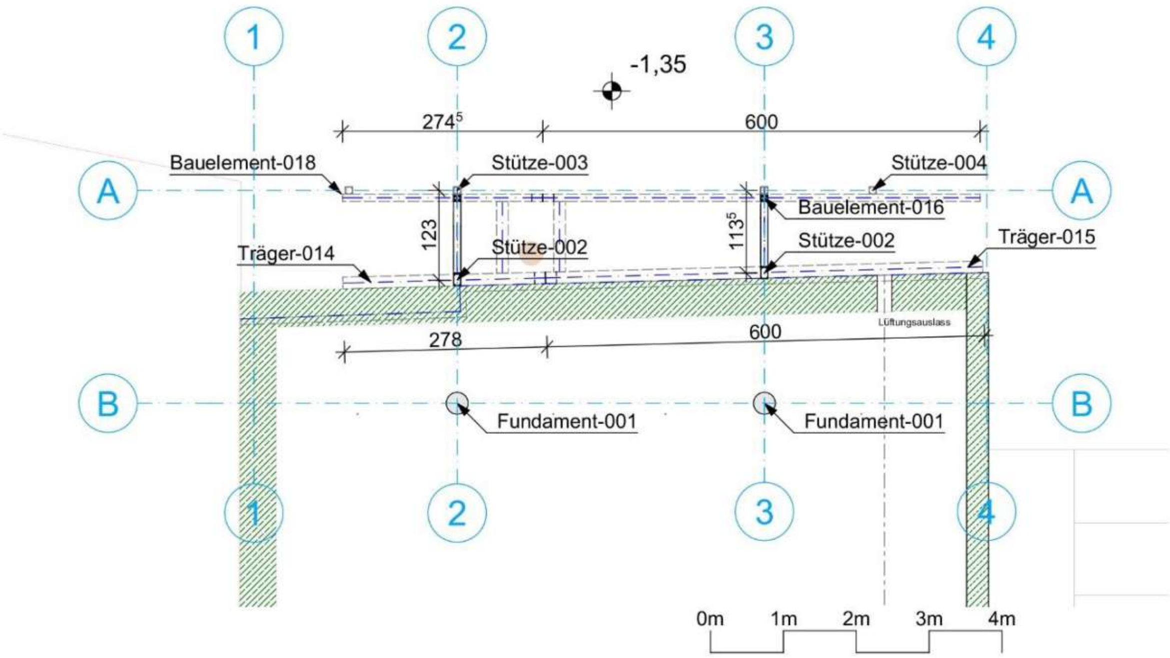


Figure 97: Floorplan Basement. M1:100

As can be seen in the plan above, and in differentiation to the model before, some elements of this project have the same Element-ID. As an example, both screw-foundations or some columns have the same Element-ID. This is done for an easier handling of data and to display certain processes at the same time, as they do not need a strict order.

Four columns (2x Stütze-001 and 2x Stütze-003 – see figure 71) bear the load of the whole structure, these are supported by bracings to minimize deflection of the beams above. These columns are connected to the cantilevers and the screw-foundations.

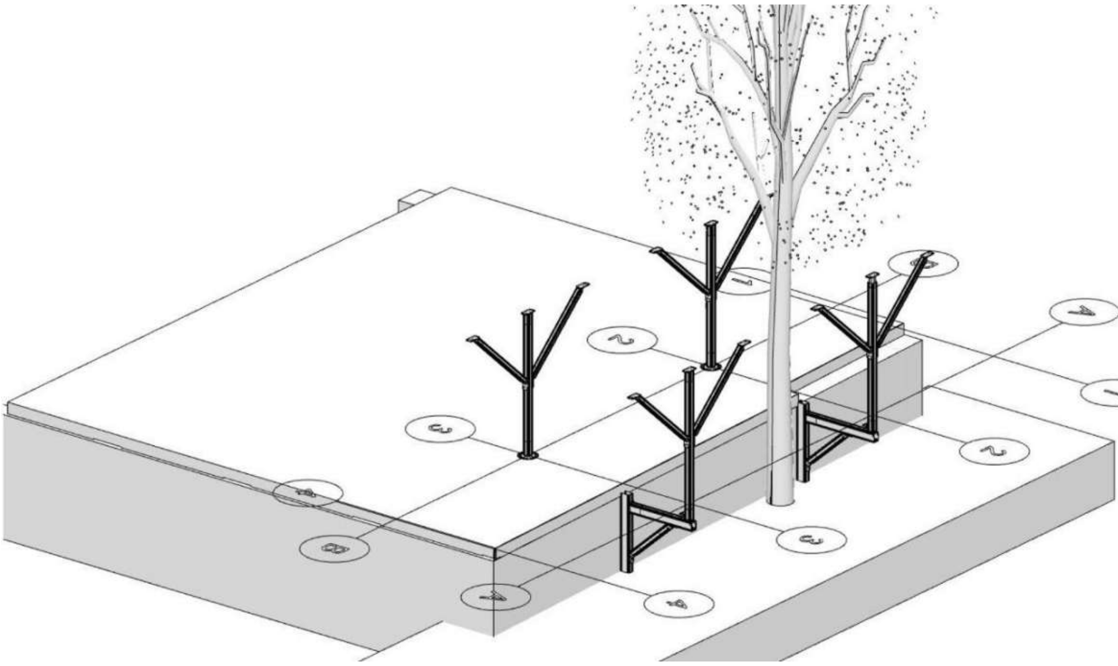


Figure 98: Depiction of the four main columns with their bracings

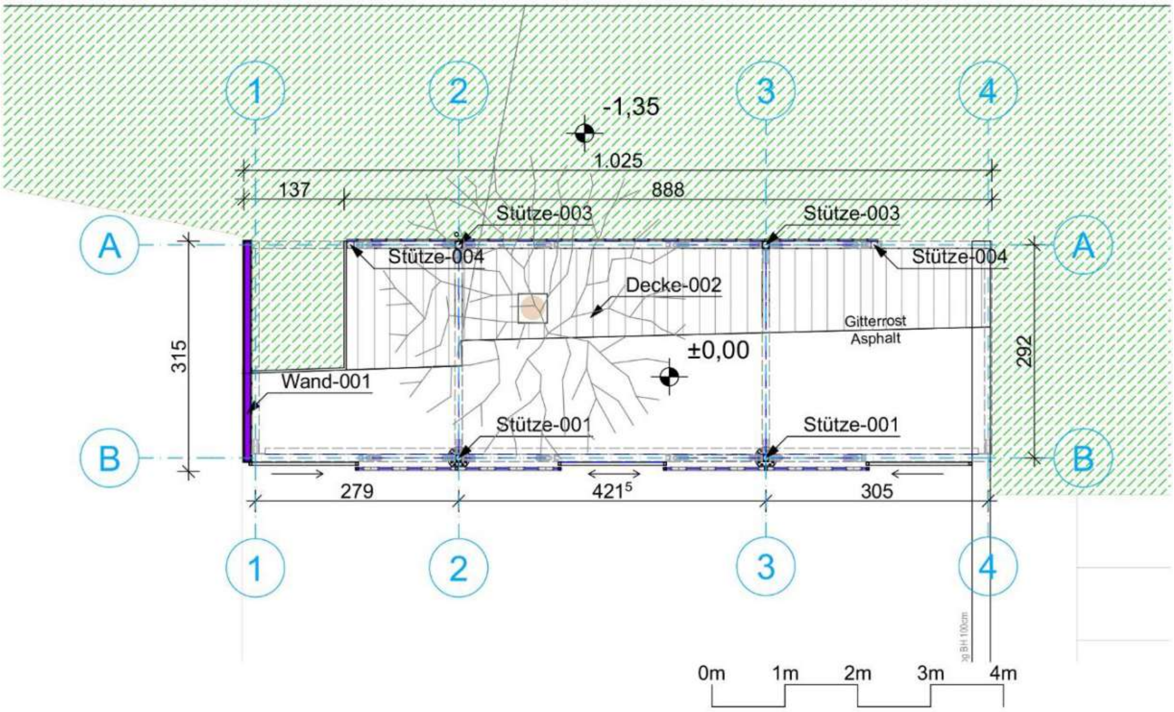


Figure 99: Floorplan of the shed. M1:100

Two extra columns (Stütze-004) in the axis A-A are utilized as tension rods, which support the beams on the floor-level to minimize their deflection.

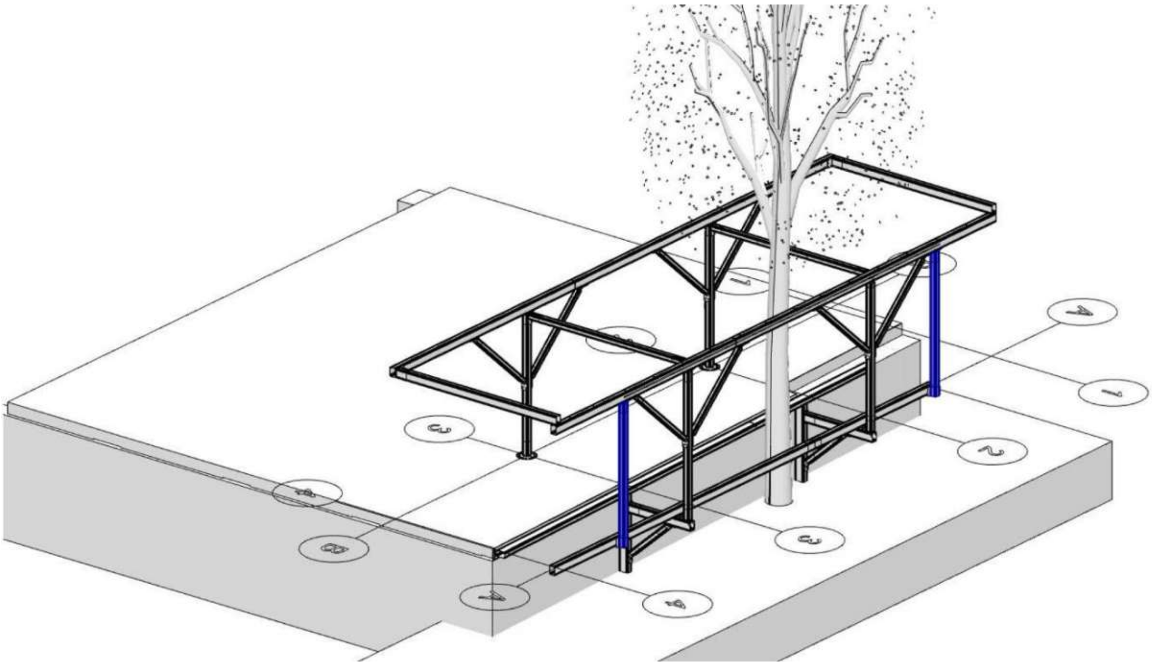


Figure 100: Depiction of the basic structure. Blue highlighted tension rods support the beam below

The flooring of the shed is made from a grating (Decke-002) and offers a seamless transition from the asphalt of the driveway. This grating is supported by the two beams (Träger-013 and Träger-014 – see figure 73) in the floor-level.

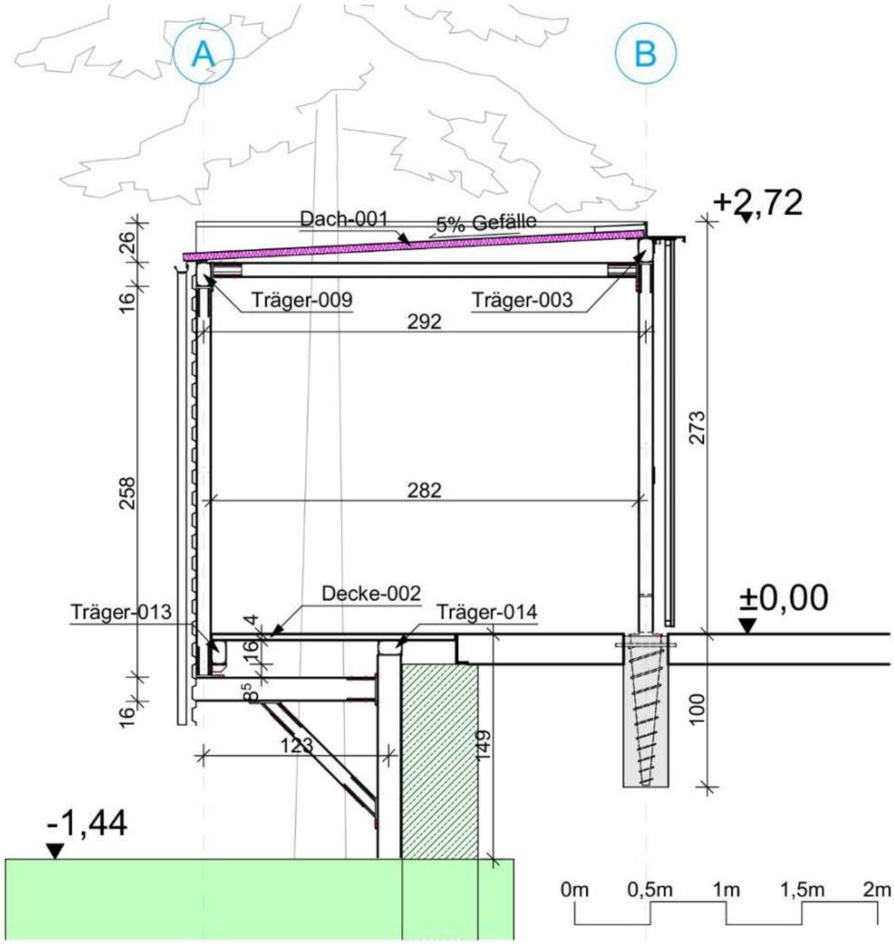


Figure 101: Section through axis 2-2. M1:50

The outer dimensions of the shed are 10,25x3,15 meters and the height are 2,73 meters. In the axis 1, a fire-protection-panel is utilized to ensure safety codes are met. The roof is made from sandwich-panels which are slightly inclined (5%) in the direction of the garden.

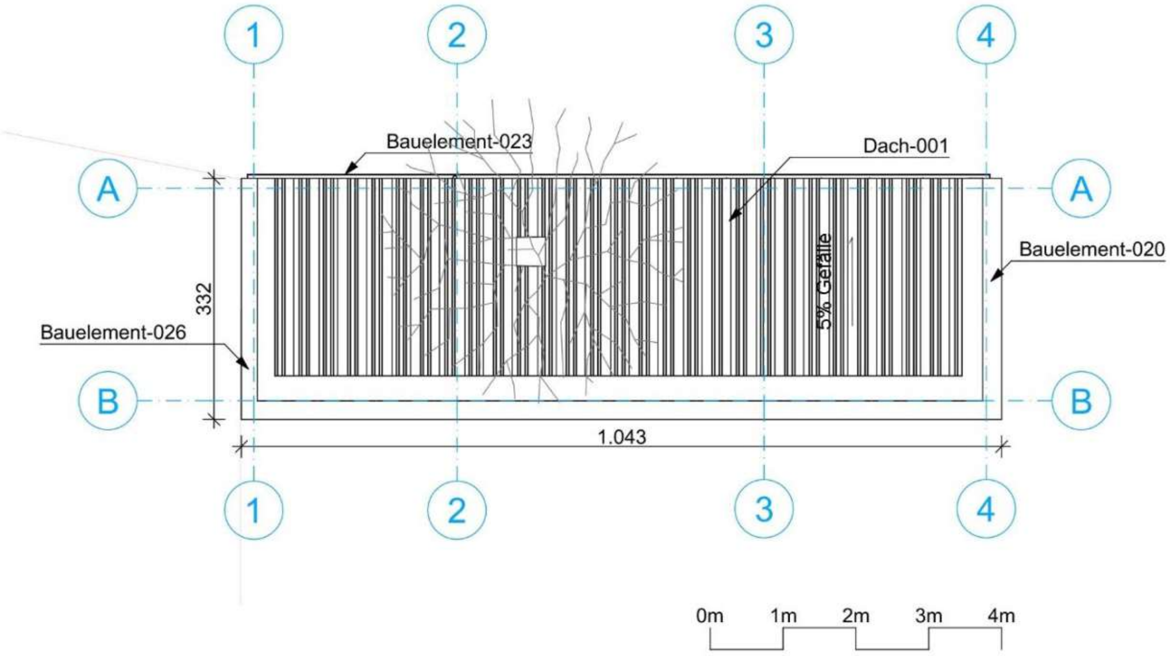


Figure 102: Topview of the project. M1:100

Towards the garden, profiled sheeting is used and towards the driveway three lockable sliding-doors and two fixed parts made of expanded metal are chosen as paneling. These elements will be prefabricated and then assembled on site. All coating of crucial parts like attic and verges is made by a tinsmith onsite, as is the gutter and the rain pipe.

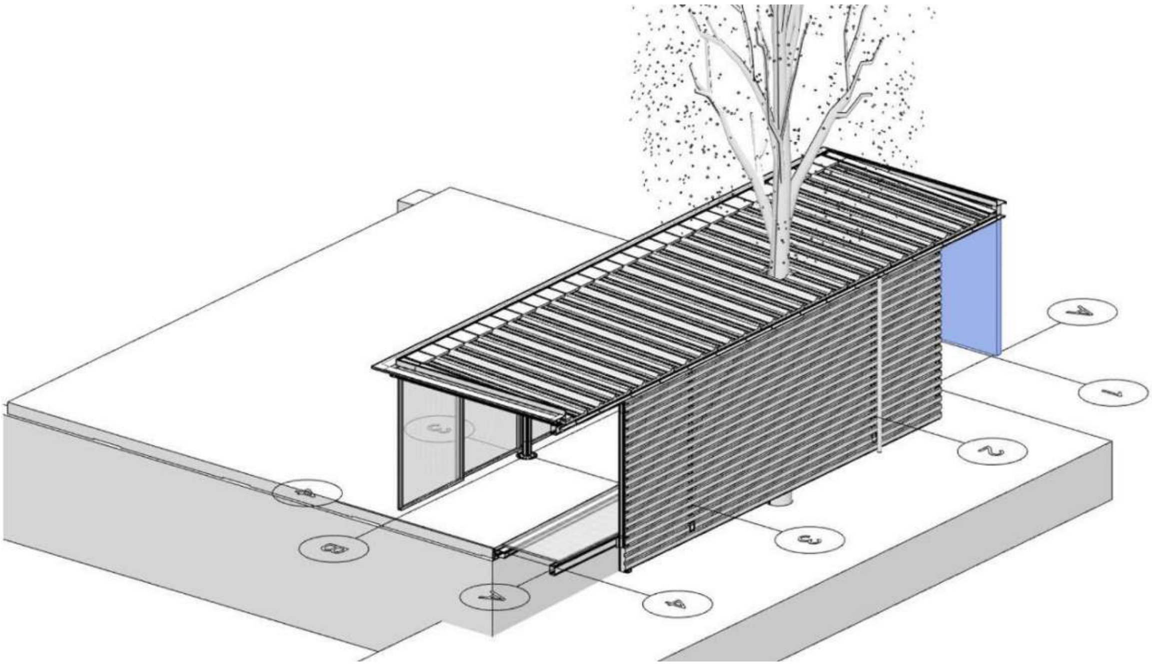


Figure 103: Whole shed with all coverings. The fire-protection-panel is highlighted in blue

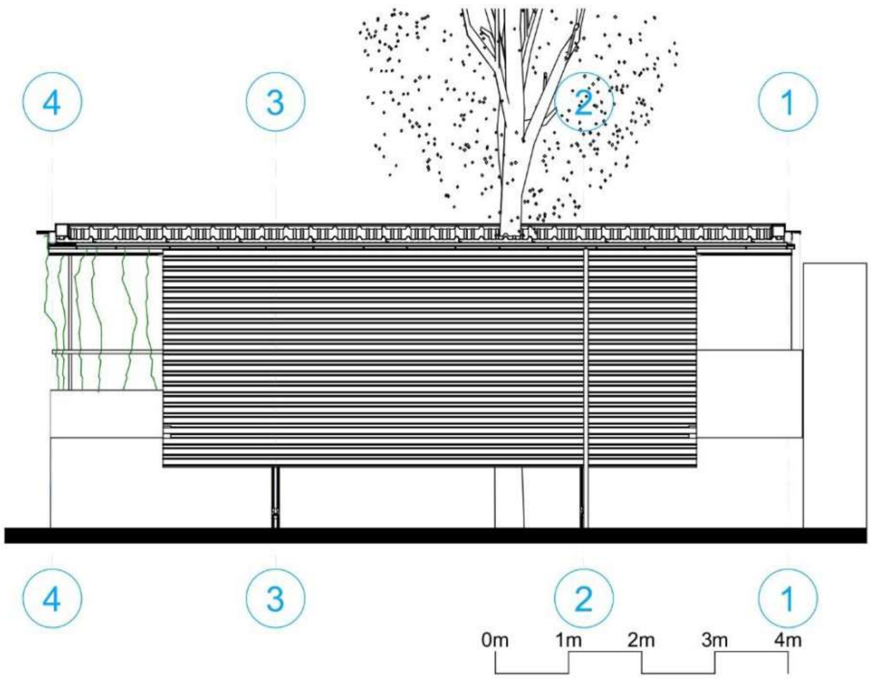


Figure 104: Elevation East M1:100. Profiled sheeting is used as paneling.

To the south of the driveway, the guardrail will be provided by plant containers in the future. Climbing plants should cover the side of the shed.

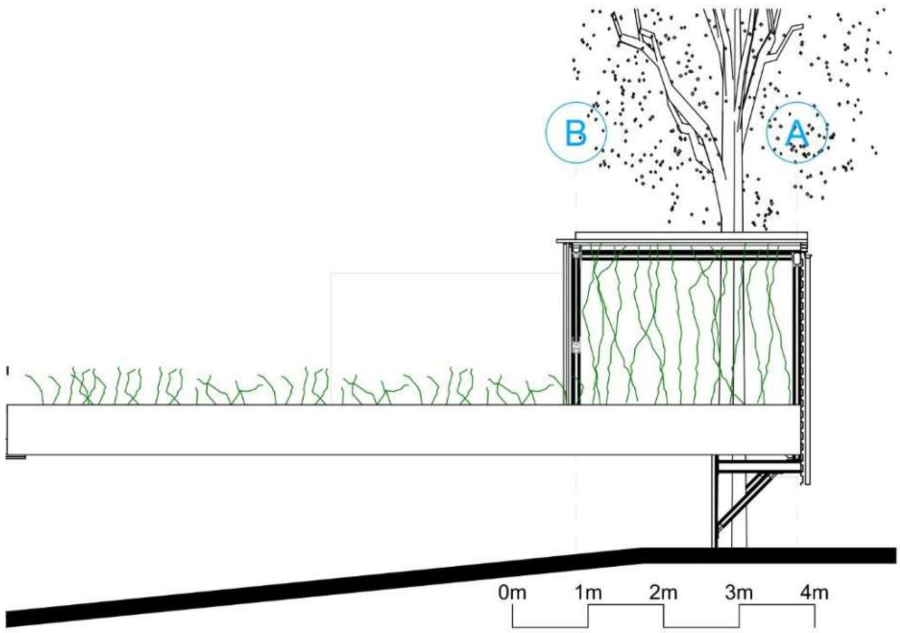


Figure 105: Elevation South M1:100. Plant containers will be the guardrail to this side in the future

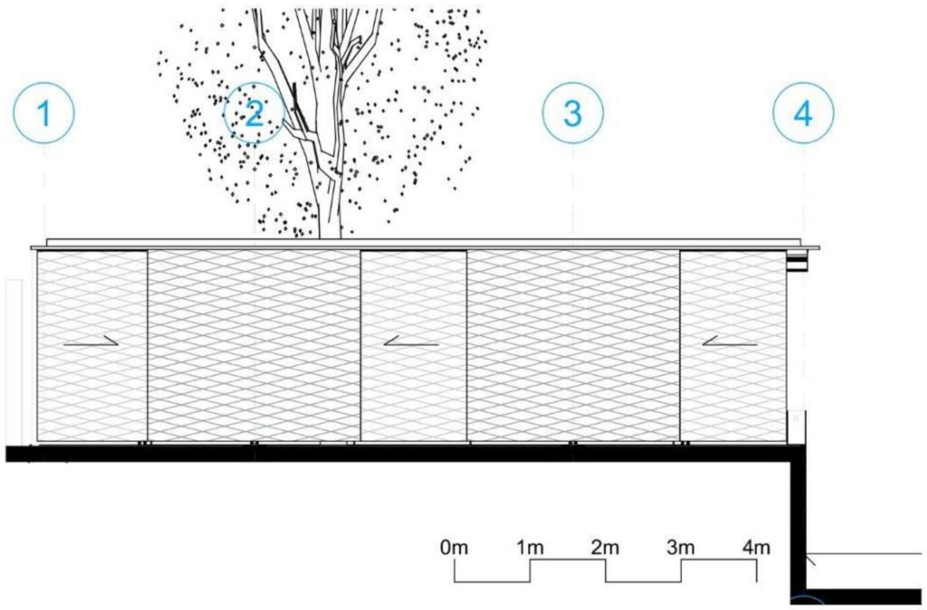


Figure 106: Elevation West M1:100. The Paneling is made of expanded metal. Three sliding panels provide access to the shed

Construction method

The primary structure of the shed is built with modular steel-elements, which are screwed together with a patented system from the company "Sikla", called "siFramo". This system allows a fast assembly and a high grade of prefabrication. All parts were ordered and produced by using the IFC-file of this project.

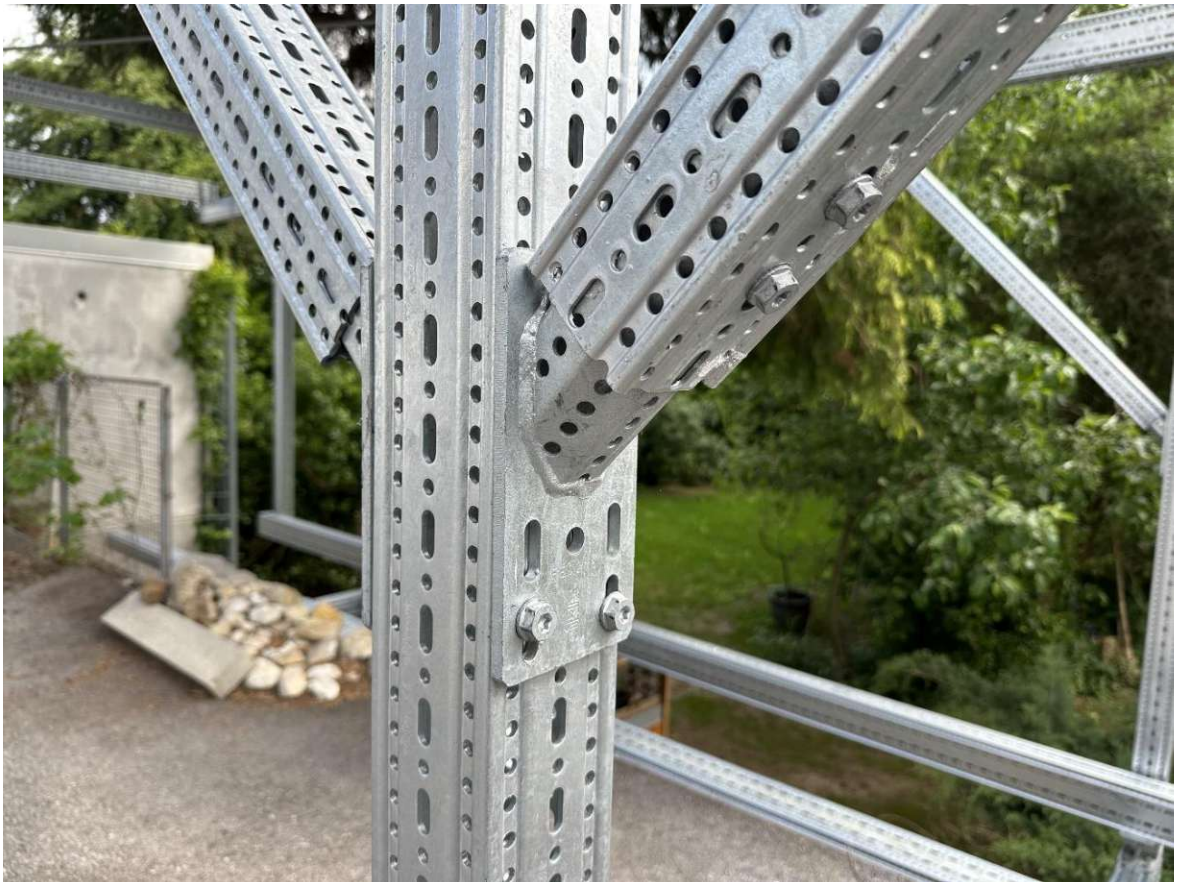


Figure 107: Photograph of a connection made with prefabricated steel-elements from the company Sikla

As mentioned above, there are some limitations with the length of single objects. The standard length of the beams and columns is 6m and there is one type of screw which is used in this system. Connections are made mostly with off the shelf products, only 1 connection for the beams in axes 1 and 4 are custom made.

The standard profiles for all parts of this project range from 80x80mm for bracings, over 100x100mm for columns and some beams, to 160x100mm for the main beams and cantilevers.

All profiles are perforated with a series of holes and long holes. Also, the connectors have a fitting series of long holes and holes. This ensures a seamless connection with high flexibility.

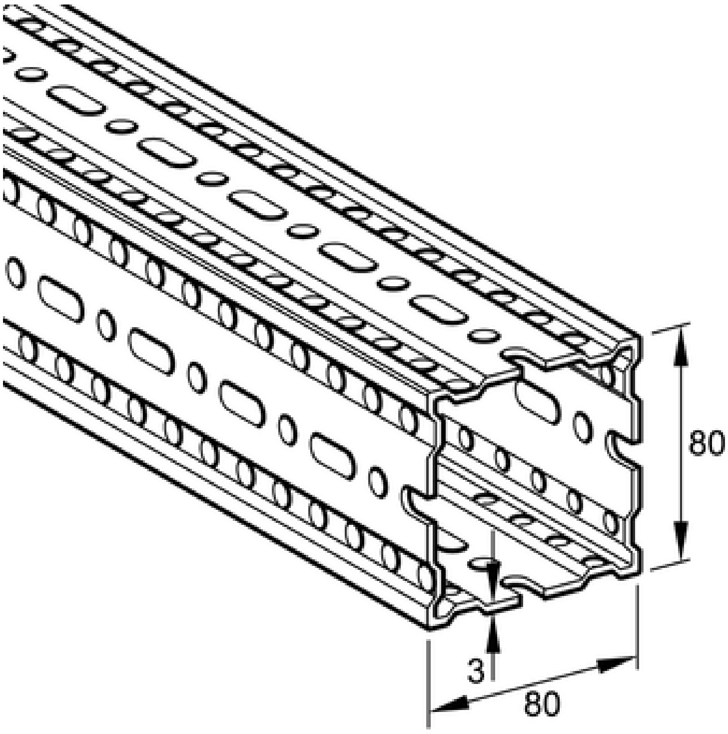


Figure 108: beam profile TP F 80 - used as bracings in this project. (source: sikla.at)

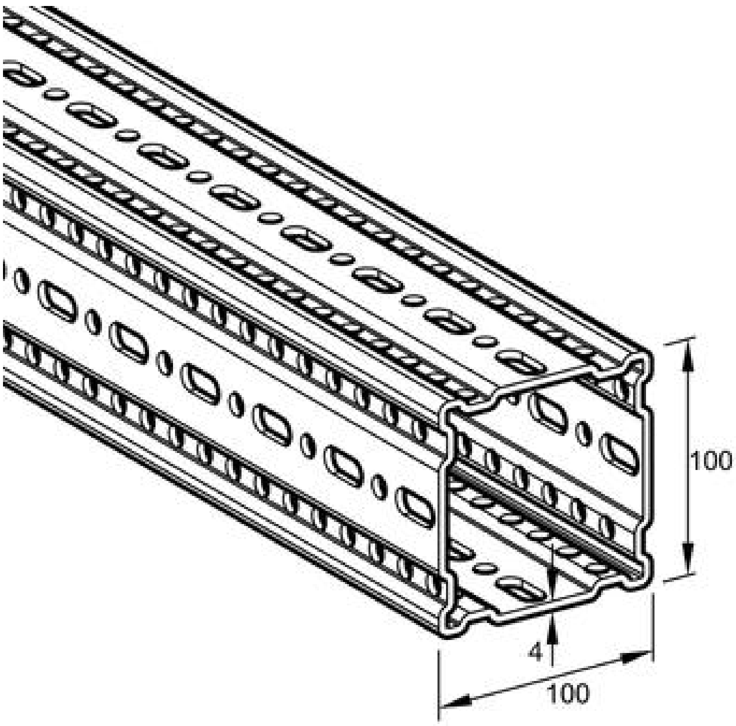


Figure 109: beam profile TP F 100 - used as columns and beams in this project (source: sikla.at)

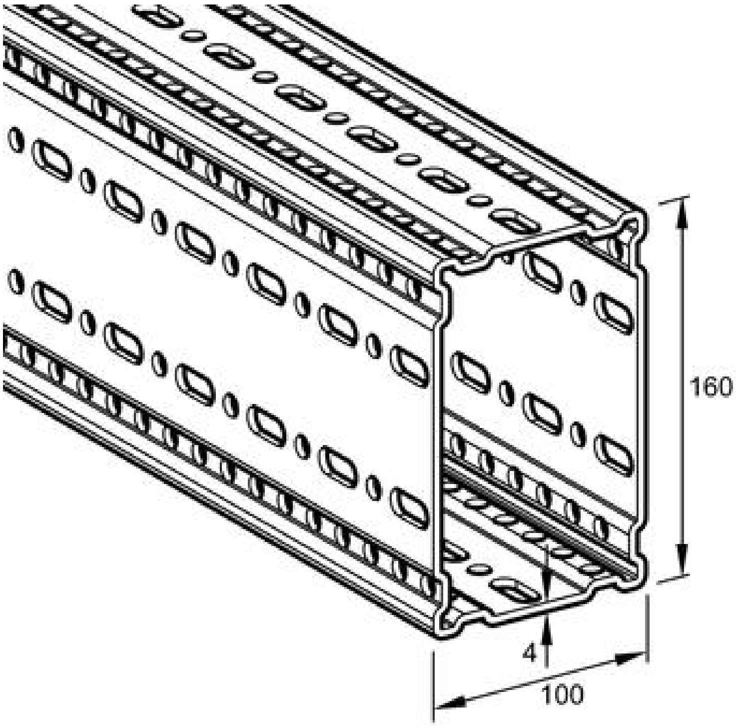


Figure 110: beam profile TP F 100_160 - used for main beams and cantilevers in this project (source: sikla.at)

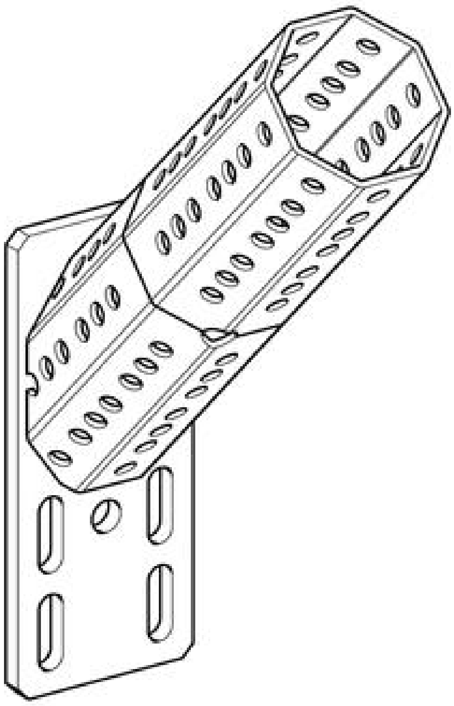


Figure 111: example of a standard connection used for the connection of bracings to beams and columns. STA_F_80_45 (source: sikla.at)

Schedule

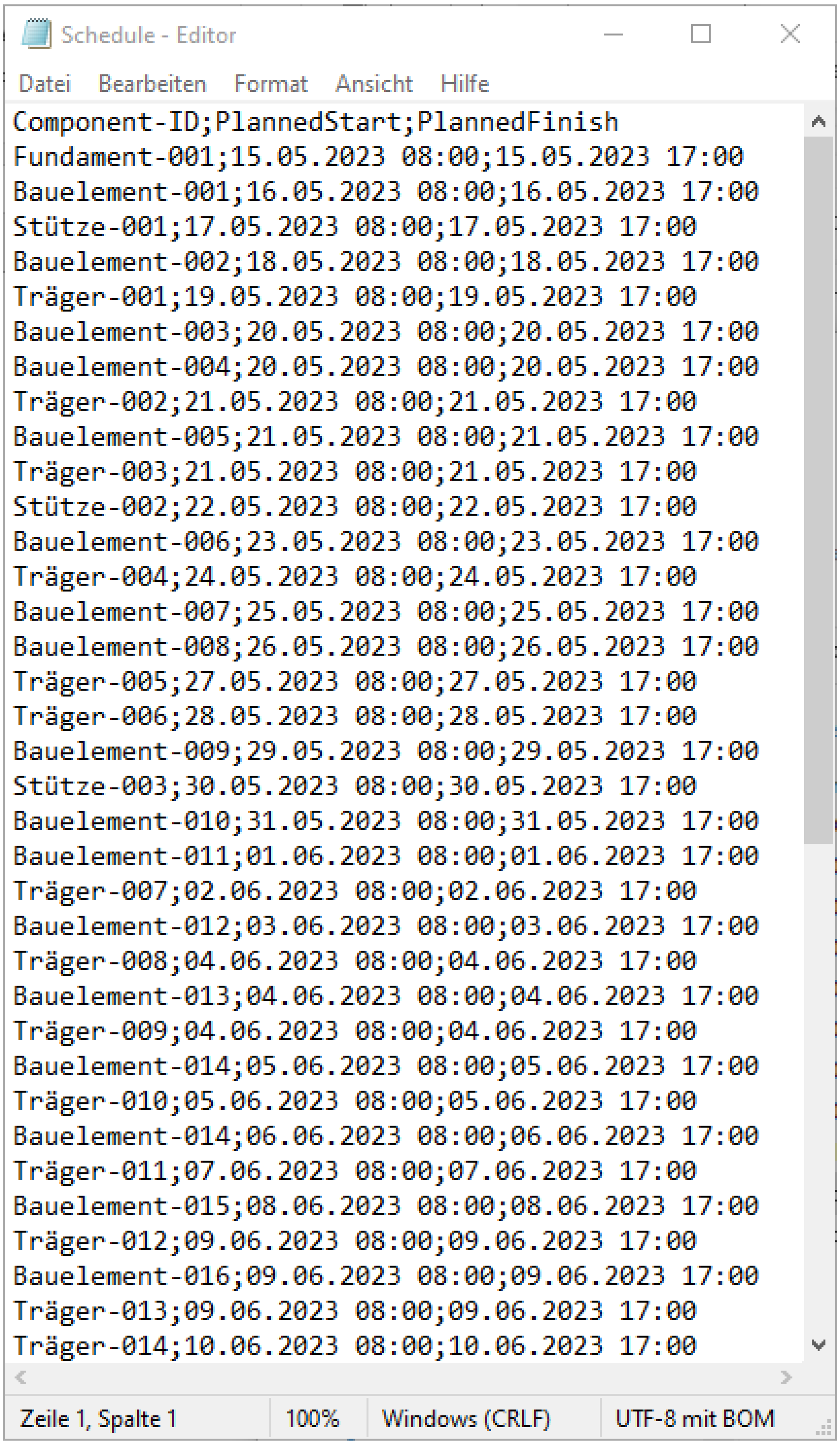
The schedule for this project was also created in Excel and was then exported as CSV-file for further usage. Since some objects are built simultaneously and have the same geometry, it was chosen to give them the same Element-ID.

As mentioned in the first project, the schedule determines a “PlannedStart” and a “PlannedFinish” for every object. Again, whole days between steps are skipped to provide a better explanation of the process.

Some Elements have different Element-IDs, but are assembled in one step, due to the building process. Therefore, they have the same timestamps in the schedule to ensure the practicability of the assembly.

Component-ID	PlannedStart	PlannedFinish
Fundament-001	15.05.2023 08:00	15.05.2023 17:00
Bauelement-001	16.05.2023 08:00	16.05.2023 17:00
Stütze-001	17.05.2023 08:00	17.05.2023 17:00
Bauelement-002	18.05.2023 08:00	18.05.2023 17:00
Träger-001	19.05.2023 08:00	19.05.2023 17:00
Bauelement-003	20.05.2023 08:00	20.05.2023 17:00
Bauelement-004	20.05.2023 08:00	20.05.2023 17:00
Träger-002	21.05.2023 08:00	21.05.2023 17:00
Bauelement-005	21.05.2023 08:00	21.05.2023 17:00
Träger-003	21.05.2023 08:00	21.05.2023 17:00
Stütze-002	22.05.2023 08:00	22.05.2023 17:00
Bauelement-006	23.05.2023 08:00	23.05.2023 17:00
Träger-004	24.05.2023 08:00	24.05.2023 17:00
Bauelement-007	25.05.2023 08:00	25.05.2023 17:00
Bauelement-008	26.05.2023 08:00	26.05.2023 17:00
Träger-005	27.05.2023 08:00	27.05.2023 17:00
Träger-006	28.05.2023 08:00	28.05.2023 17:00
Bauelement-009	29.05.2023 08:00	29.05.2023 17:00
Stütze-003	30.05.2023 08:00	30.05.2023 17:00
Bauelement-010	31.05.2023 08:00	31.05.2023 17:00
Bauelement-011	01.06.2023 08:00	01.06.2023 17:00
Träger-007	02.06.2023 08:00	02.06.2023 17:00
Bauelement-012	03.06.2023 08:00	03.06.2023 17:00
Träger-008	04.06.2023 08:00	04.06.2023 17:00
Bauelement-013	04.06.2023 08:00	04.06.2023 17:00
Träger-009	04.06.2023 08:00	04.06.2023 17:00
Bauelement-014	05.06.2023 08:00	05.06.2023 17:00
Träger-010	05.06.2023 08:00	05.06.2023 17:00
Bauelement-014	06.06.2023 08:00	06.06.2023 17:00
Träger-011	07.06.2023 08:00	07.06.2023 17:00
Bauelement-015	08.06.2023 08:00	08.06.2023 17:00
Träger-012	09.06.2023 08:00	09.06.2023 17:00
Bauelement-016	09.06.2023 08:00	09.06.2023 17:00
Träger-013	09.06.2023 08:00	09.06.2023 17:00
Träger-014	10.06.2023 08:00	10.06.2023 17:00

Figure 112: Excerpt of the schedule in Excel for the project "Garden Shed"


Schedule - Editor
Datei Bearbeiten Format Ansicht Hilfe

```
Component-ID;PlannedStart;PlannedFinish
Fundament-001;15.05.2023 08:00;15.05.2023 17:00
Bauelement-001;16.05.2023 08:00;16.05.2023 17:00
Stütze-001;17.05.2023 08:00;17.05.2023 17:00
Bauelement-002;18.05.2023 08:00;18.05.2023 17:00
Träger-001;19.05.2023 08:00;19.05.2023 17:00
Bauelement-003;20.05.2023 08:00;20.05.2023 17:00
Bauelement-004;20.05.2023 08:00;20.05.2023 17:00
Träger-002;21.05.2023 08:00;21.05.2023 17:00
Bauelement-005;21.05.2023 08:00;21.05.2023 17:00
Träger-003;21.05.2023 08:00;21.05.2023 17:00
Stütze-002;22.05.2023 08:00;22.05.2023 17:00
Bauelement-006;23.05.2023 08:00;23.05.2023 17:00
Träger-004;24.05.2023 08:00;24.05.2023 17:00
Bauelement-007;25.05.2023 08:00;25.05.2023 17:00
Bauelement-008;26.05.2023 08:00;26.05.2023 17:00
Träger-005;27.05.2023 08:00;27.05.2023 17:00
Träger-006;28.05.2023 08:00;28.05.2023 17:00
Bauelement-009;29.05.2023 08:00;29.05.2023 17:00
Stütze-003;30.05.2023 08:00;30.05.2023 17:00
Bauelement-010;31.05.2023 08:00;31.05.2023 17:00
Bauelement-011;01.06.2023 08:00;01.06.2023 17:00
Träger-007;02.06.2023 08:00;02.06.2023 17:00
Bauelement-012;03.06.2023 08:00;03.06.2023 17:00
Träger-008;04.06.2023 08:00;04.06.2023 17:00
Bauelement-013;04.06.2023 08:00;04.06.2023 17:00
Träger-009;04.06.2023 08:00;04.06.2023 17:00
Bauelement-014;05.06.2023 08:00;05.06.2023 17:00
Träger-010;05.06.2023 08:00;05.06.2023 17:00
Bauelement-014;06.06.2023 08:00;06.06.2023 17:00
Träger-011;07.06.2023 08:00;07.06.2023 17:00
Bauelement-015;08.06.2023 08:00;08.06.2023 17:00
Träger-012;09.06.2023 08:00;09.06.2023 17:00
Bauelement-016;09.06.2023 08:00;09.06.2023 17:00
Träger-013;09.06.2023 08:00;09.06.2023 17:00
Träger-014;10.06.2023 08:00;10.06.2023 17:00
```

Zeile 1, Spalte 1 | 100% | Windows (CRLF) | UTF-8 mit BOM

Figure 113: Excerpt of the schedule as CSV-file in a text editor for the project "Garden Shed"

4.3.2 Workflow-Test for the Project “Garden Shed”

To test the application, in a real setting, the project described in 4.1.2 will be augmented and the workflow will be tested on feasibility.

Although it is still a relatively small project, a lot of different and more complex construction methods for the BIM-model are used and this leads to new challenges.

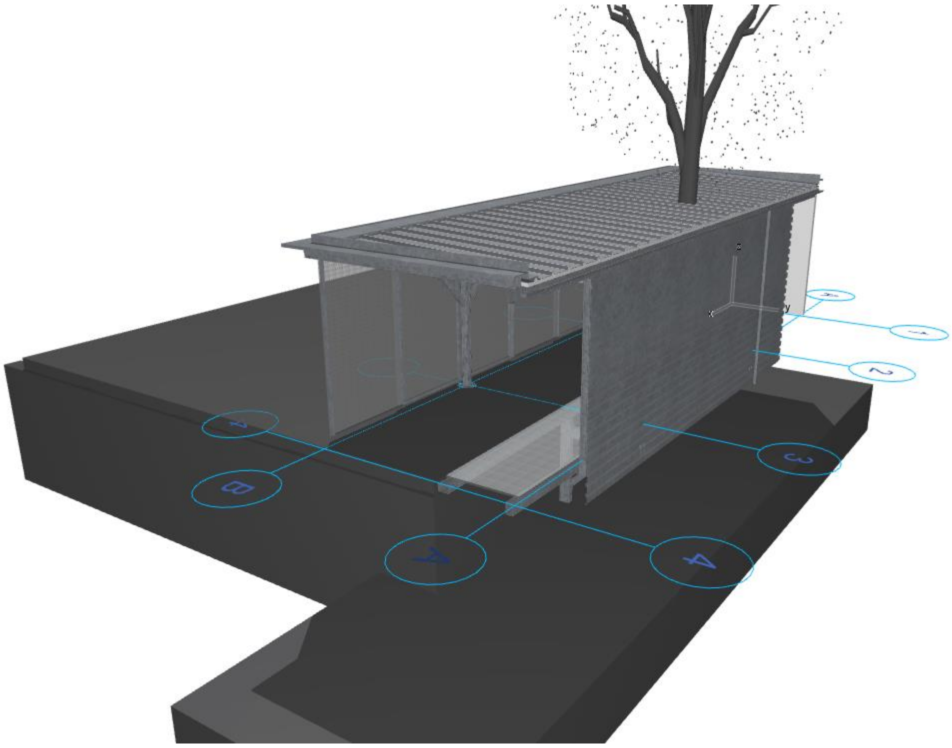


Figure 114: Test2 - Project in Archicad

BIM Project

Because this is a real site, there is already a built structure. It was important to distinct those objects from the new ones and to properly survey the site before planning started.

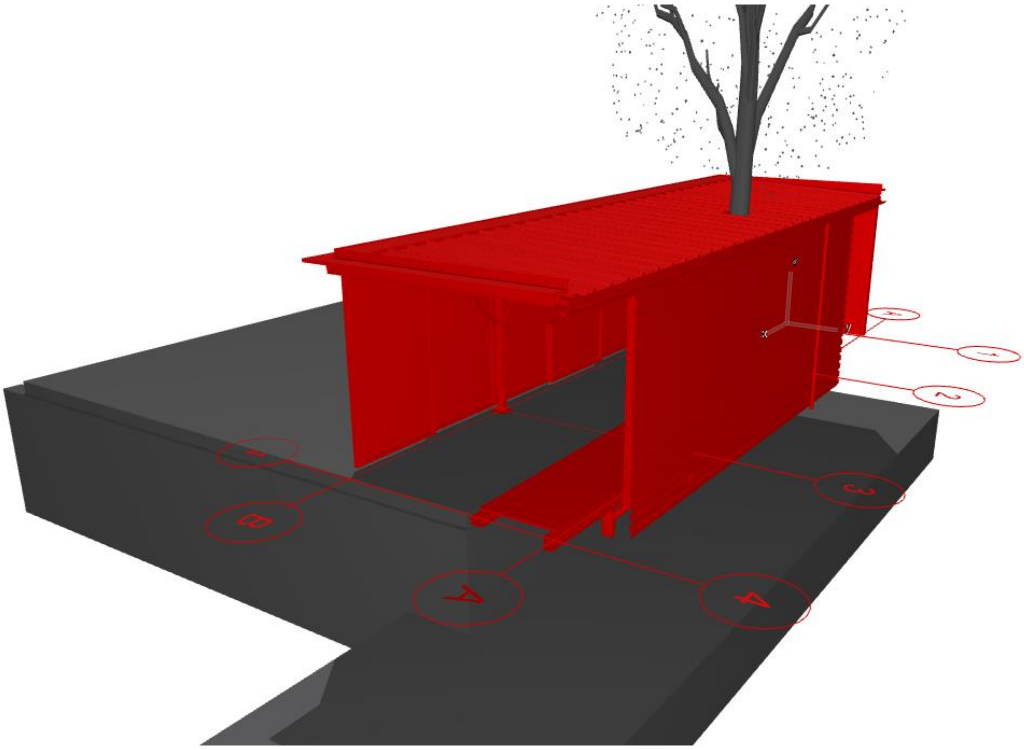


Figure 115: Test2 - Distinction between new (red) and old (grey) structure in Archicad

The survey of the site was done by Egon Grünwald during his master-thesis. A point-cloud was created with a laser scanner from the company Leica.

Figure 116: Example-picture of a laser scanner from the company Leica (source: www.leica-geosystems.com)

To survey the whole site, the laser scanner was positioned at four different locations on the driveway.

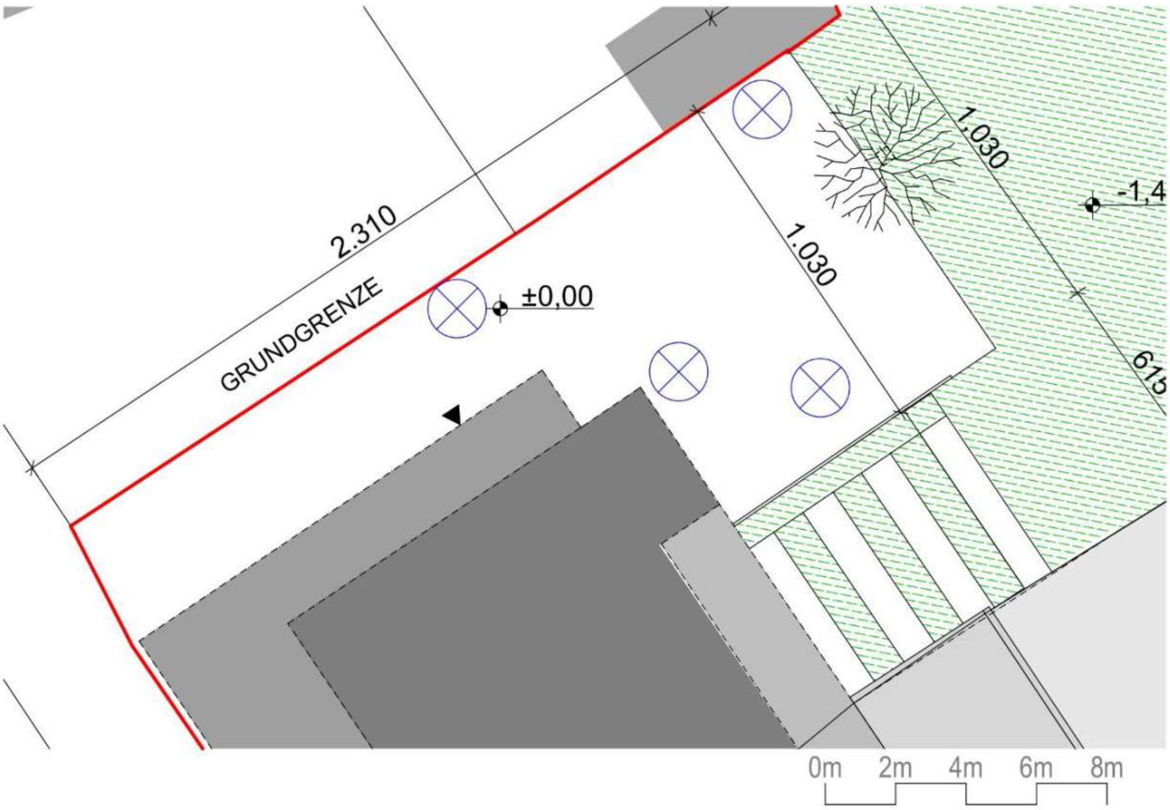


Figure 117: Positioning of the laser scanner onsite M1:200

The data was then processed via Leica´s software. This process was done by Egon Grünwald, and then the data was imported into Archicad via the cloud point import option.

Figure 118: Picture of the pointcloud in Archicad.

Figure 119: Picture of the point cloud in Archicad

Unfortunately, some parts of the surrounding area were not scanned properly, because of obstructions like the tree, which was not cut out at this point. Therefore, some measurements were taken with a tape measure to ensure the complete the survey of the area.

This project is also a little more complicated to design in Archicad. There are multiple objects which are designed with custom profiles and objects which consist of more than one material.

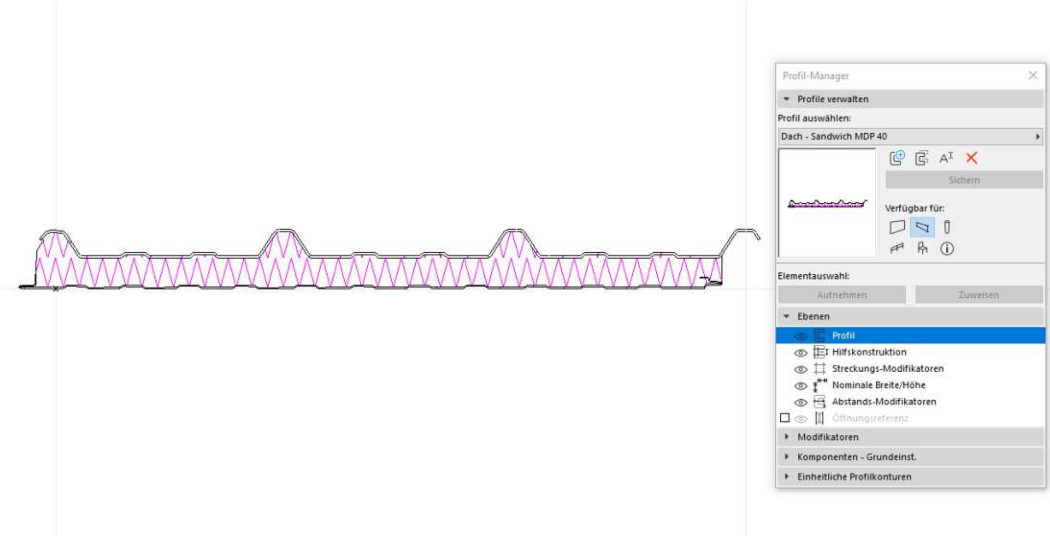


Figure 120: Test2 - complex profiles in Archicad with more than one material

It was also important to properly model the structure in its parts, so that different steps of construction can be visualized later. This means that a complex profile, which could also be drawn as one profile, must be split in its parts to separate every assembly-step properly. In the following figure, the whole detail is structured as one profile, and then the red-marked part is separated to ensure a more detailed process can be visualized.

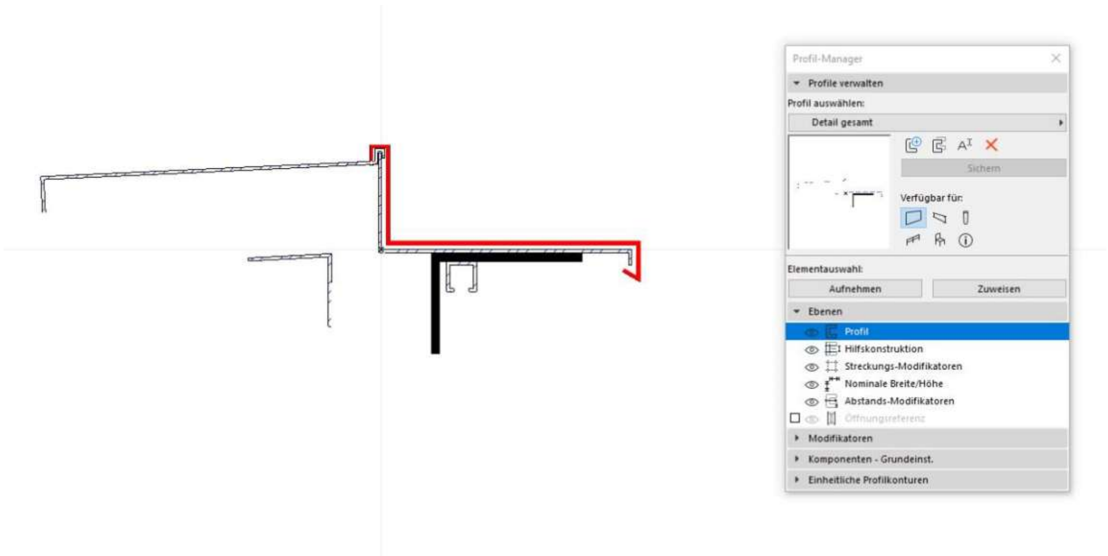


Figure 121: Test2 - Whole detailed construction would be difficult to split in Unity. The red marked part is separated and drawn individually

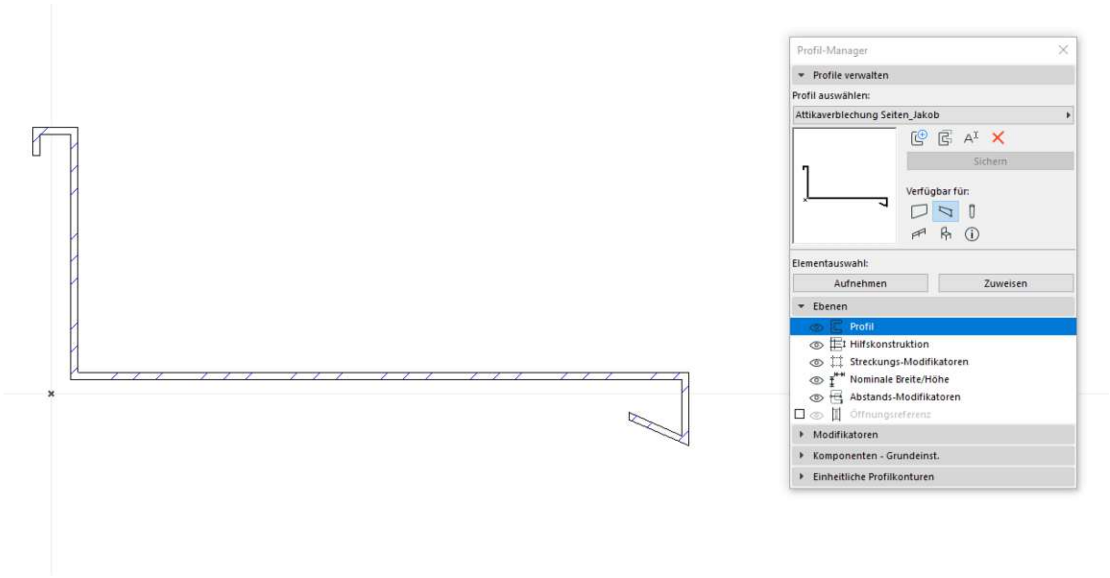


Figure 122: Test2 - the parts must be drawn separately to be identified easier late in the process

This must happen for every part of the profile shown in “Figure 120”. After that it is still important to give the objects the correct IDs to properly connect a building schedule later in the process.

IFC Export

There was no change needed in this step of the project.

Import IFC to AR-Development-Platform

Here first major changes must be implemented as a result of the more complex building components. The complex profiles which consist of more than one part, are imported different than simple objects.

As in contrast to “Figure 74”, the main GameObject of such a geometry does not contain geometry-data or a MeshRenderer. These objects are imported with children in the prefab, which hold the geometry-data and MeshRenderer. Because of this instance, the code must be altered to properly function. This is achieved with the function in Unity called “GetComponentsInChildren”. This function can call up the MeshRenderer of all children.

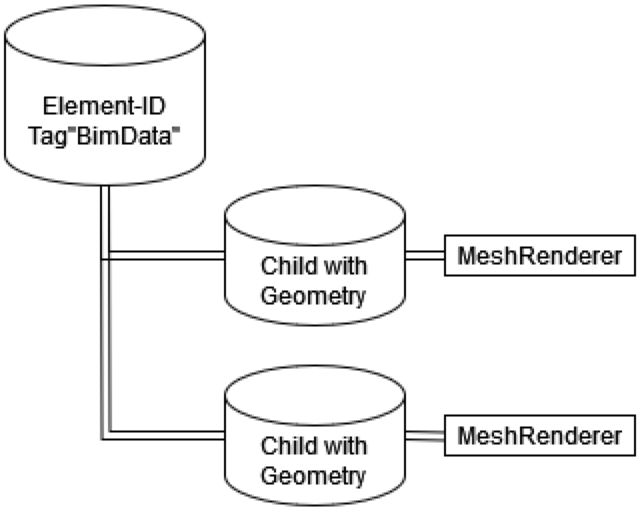


Figure 123: Main GameObject does not contain geometry, but children contain the MeshRenderer

Another point which must be implemented is occlusion. Occlusion means that parts of the 3d model should not be rendered if they are concealed behind real objects. If this is not properly implemented, depth-perception is not easily processed by the user and misleading augmentation is the result. In the following figure occlusion is illustrated. The red marked object (left) must be modeled in 3D and is given a special shader in Unity to cover the augmented object (red apple). The picture in the middle demonstrates augmentation without occlusion. The apple seems to hover in front of the real object and its augmented shadow is misleading. In the picture on the right, occlusion is implemented and only the parts of the apple, which are not covered by the 3D-model of the real object, are rendered.

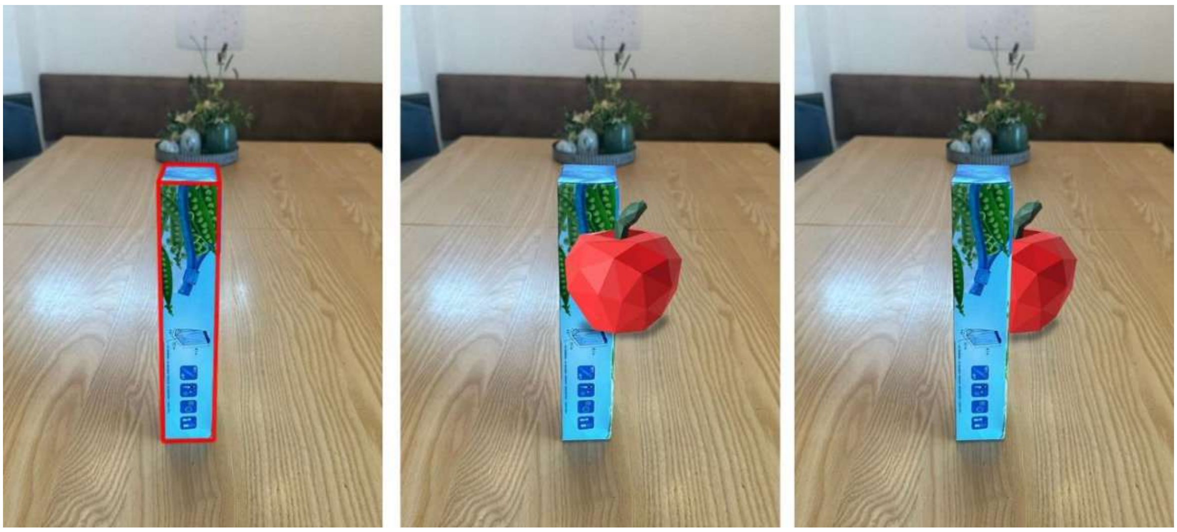


Figure 124: Occlusion - The red marked object (left) must be modeld in 3D. Augmentation with Oclusion (middle). Augmentation with occlusion (right)

In this project-study the level difference between the garden and the driveway should conceal some of the augmented structure. Also, the foundation should be under the ground and not be seen in the rendered augmentation. If these parts are rendered it is very confusing to understand the augmentation properly, like it is shown in the following figure.

Figure 125: Augmentation without occlusion. Parts which should be behind other objects are rendered on top of them.

For this the terrain is also modeled and a so called “Depth Mask Shader” is added to the objects in Unity. This is a feature of Vuforia and allows the correct representation of the model.

Figure 126: Augmentation with occlusion. Elements are hidden behind the 3d-modeled terrain with “Depth Mask Shader”.

Also, the tag “BimData” should be distributed via a script to the main GameObjects, because the number of GameObjects gets higher and is not easily handled.

Add a simple schedule to the IFC

Because this project is more complex the project in first Test, the schedule also got more complex, just because of the increased number of objects. This is seen and described in 4.3.1.

The export and the assignment in Unity still are the same and work as intended.

Add time to the application to show currently important parts of the 3D-model only

This step stays the same, only the problem with the whole model showing in red at the start is fixed and implemented.

For better understanding of the building-process, the following figures provide an overview about the first steps of the assembly. The schedule for the first steps of the project “Garden Shed” along with drawings of the process are depicted below.

Component-ID	PlannedStart	PlannedFinish
Fundament-001	15.05.2023 08:00	15.05.2023 17:00
Bauelement-001	16.05.2023 08:00	16.05.2023 17:00
Stütze-001	17.05.2023 08:00	17.05.2023 17:00
Bauelement-002	18.05.2023 08:00	18.05.2023 17:00
Träger-001	19.05.2023 08:00	19.05.2023 17:00
Bauelement-003	20.05.2023 08:00	20.05.2023 17:00
Bauelement-004	20.05.2023 08:00	20.05.2023 17:00
Träger-002	21.05.2023 08:00	21.05.2023 17:00
Bauelement-005	21.05.2023 08:00	21.05.2023 17:00
Träger-003	21.05.2023 08:00	21.05.2023 17:00

Figure 127: First part of the schedule for the project "Garden Shed"

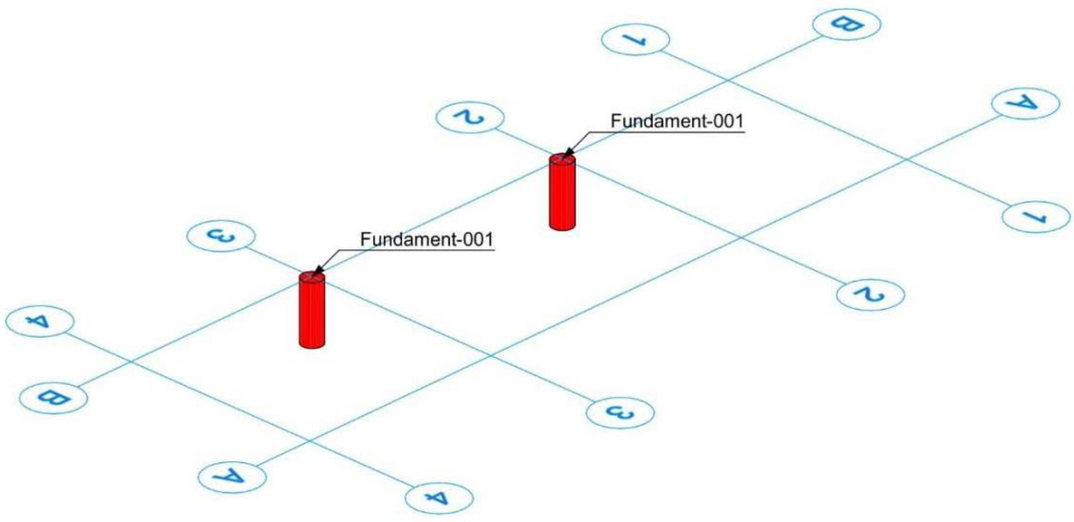


Figure 128: Step 1 - The screw-fundaments are placed in B2 and B3

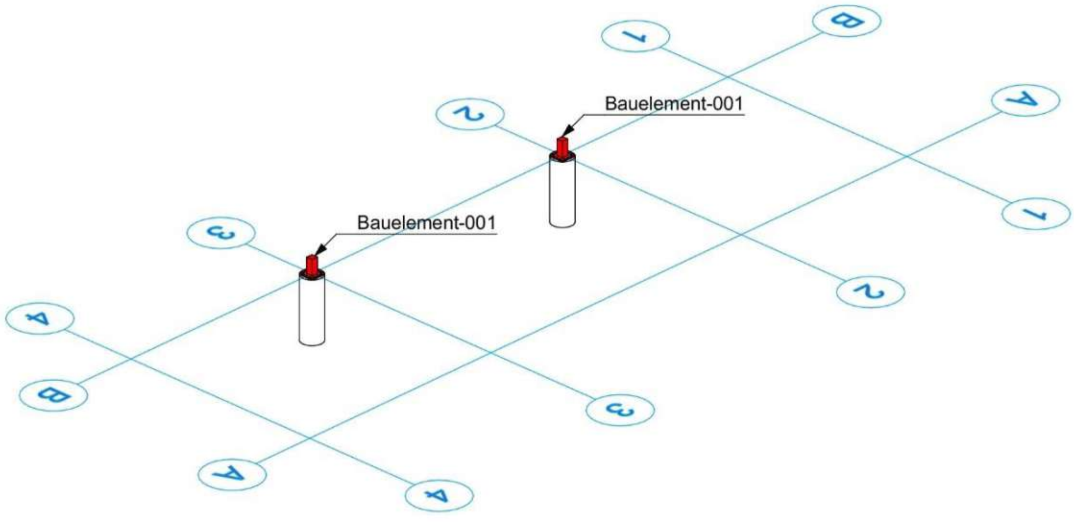


Figure 129: Step 2 - The connections for the columns are fixed to the screw-fundaments

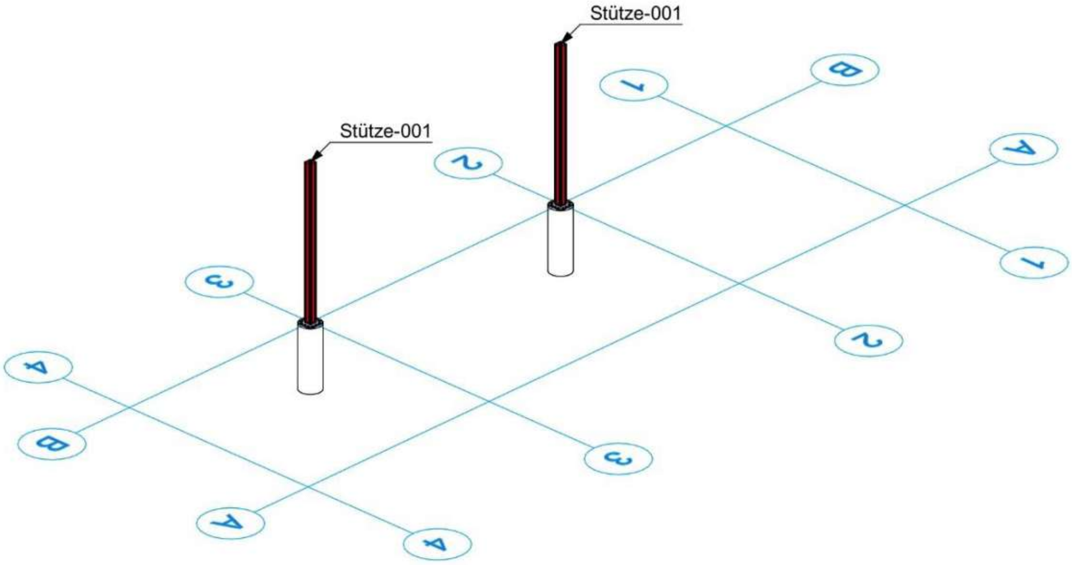


Figure 130: Step 3 - The columns "Stütze-001" are placed

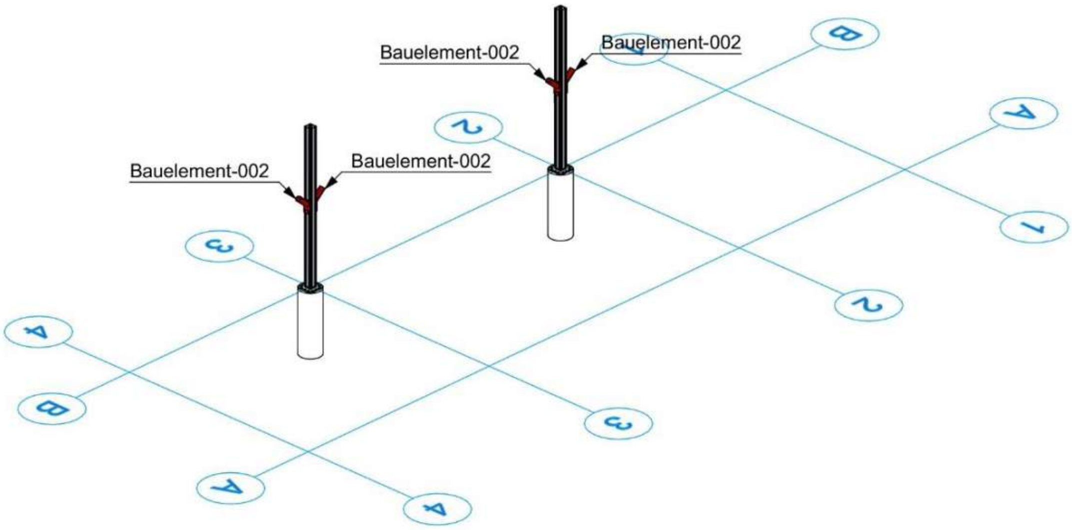


Figure 131: The connections between columns and bracings are assambled

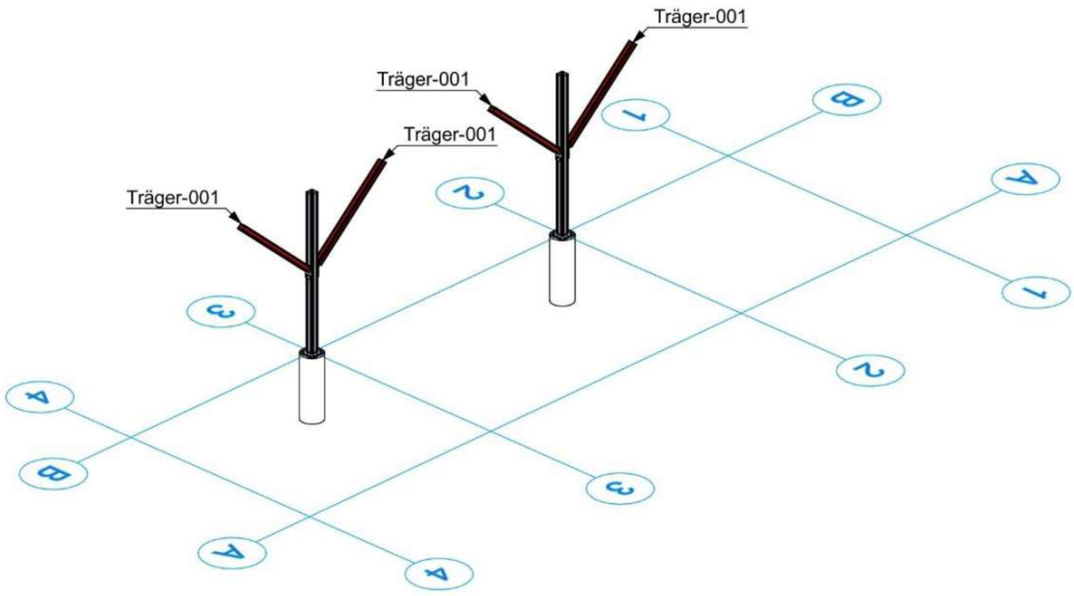


Figure 132: Step 5 - The bracings are placed on their position

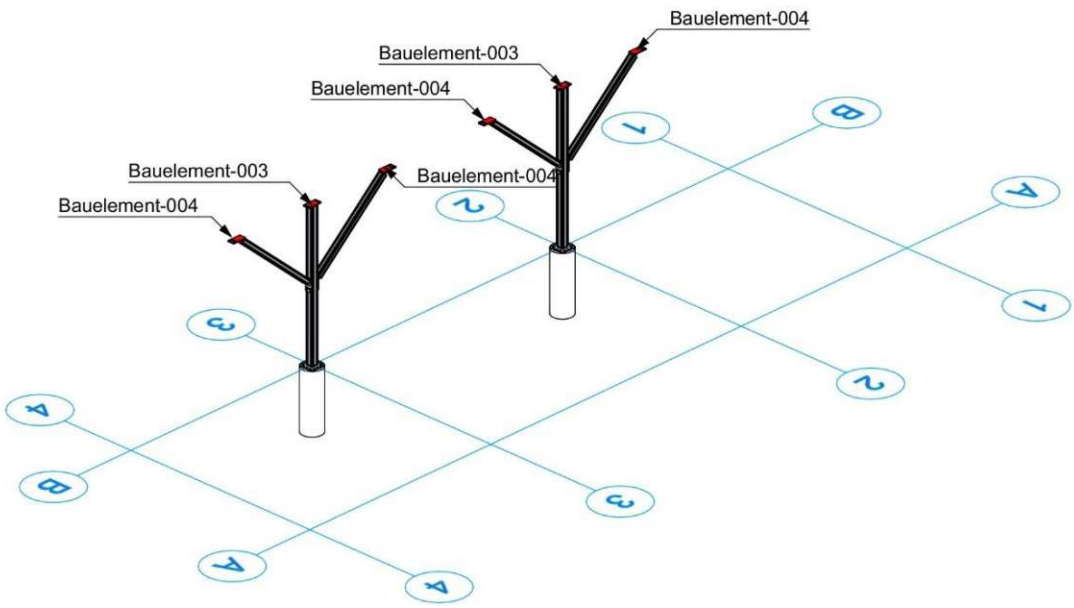


Figure 133: Step 6 - The connections between bracings, columns and the beam above are assembled and aligned

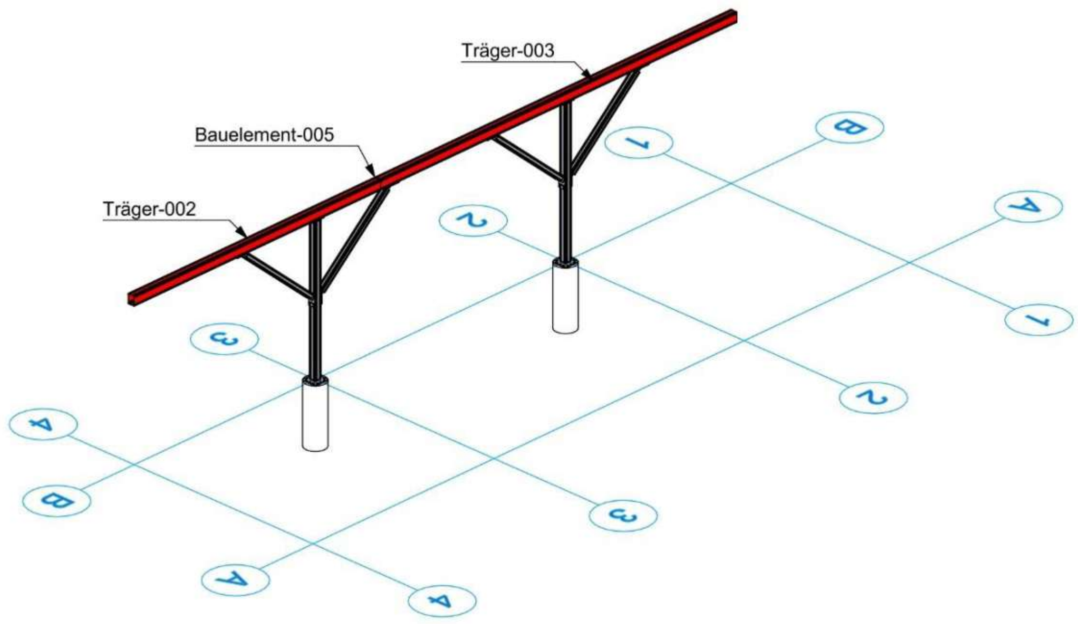


Figure 134: Step 7 - The pre-assembled beam, made of 3 elements, is placed on top of the columns and bracings

Augmentation with initial marker and Vuforia Fusion

This still is the same. The marker is just now fixed on a part of the already build structure, where it easy can be placed on the real object. Therefor the marker also must be placed at the same location in Unity, otherwise the tracking would be off.

This step is very important to be accurate, otherwise the quality and accuracy of the augmentation is directly affected.

Building AR application for IOS

Here, changes are also not needed. Only the version of the used Xcode changed, because it no longer supported the latest iOS (16.4.1).

Augmentation of the project onsite

Because of a view delays, the project already started and parts are already built. This is the perfect opportunity to test and assess the application and it´s accuracy, as deviations between 3d-model and reality are noticed immediately. In the following pictures the built structure is shown in a silver-grey and the augmented model is shown in dark-grey and red.

Figure 135: Already built parts of the structure as testing begins

The marker is fixed exactly in one meter height on the column “Stütze-001” in axis 2. Tracking starts as intended most of the time. Sometimes there is an unknown malfunction, where the model is turned by 45 degrees and the application must be restarted.

Figure 136: Error when initializing the augmentation

To initialize the correct tracking, it is important to stay focused at the marker and move slowly around it and scan the area around the marker.

Figure 137: Initial Positions tracked to get good results

After a short time, the model is stable and the marker does not have to be in line of sight of the camera. Very good results were achieved in some run-throughs.

Figure 138: After good initialization, the model has high accuracy and is stable.

Some inaccuracies between 3d model and real structure can be traced back on the inaccuracies in the model, but sometimes the tracking is also not perfect. Especially the farther away the device gets from the marker, more deviations appear. Then inaccuracies between 5 to 20cm are possible, although not the rule. These were measured with a tape measure during augmentation. In the pictures inaccuracies of the column Stütze-001 in the axis 3 are shown.

Figure 139: Inaccuracy of the structure in axis 3 when moving away from the marker. The extent is about 5cm during augmentation

Figure 140: Detailed view of the inaccuracy of the column "Stütze-001" in axis 3. The extent is about 5cm during augmentation

Also, it worth to mention that big inaccuracies above 10cm seem to only appear, when the device simultaneously is screen-recording.

Figure 141: Possible error, when moving further away from the marker. Big inaccuracies, like depicted in this picture, only appear when the device is simultaneously screen-recording

Tests were made with different weather situations and the results stayed the same overall. But there was no situation where the marker or the surrounding surfaces were overexposed to the sun.

Skipping through different stages of the project, by skipping days, is not affecting the tracking or the quality of the augmentation.

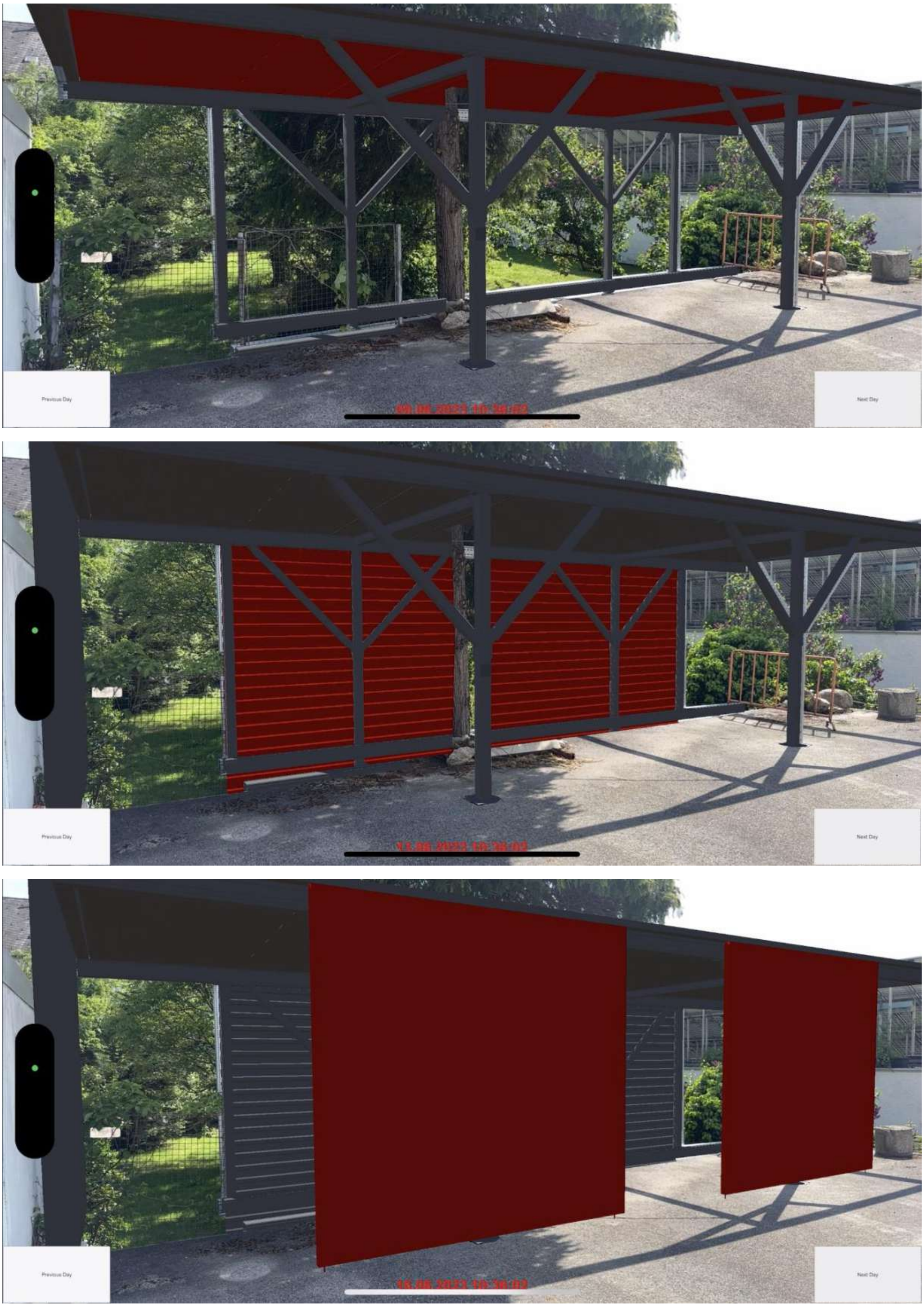


Figure 142: The depiction of three different stages of the model works as intended and creates good results

Results of the second Test

The augmentation of the project “Garden Shed” was done successfully. The correct initialization of the augmentation is more sensible then in the first test and does sometimes have errors of unknown origin as seen in Figure 136. When scanning the marker thoroughly, the results get better and the model is stable.

In some run-throughs inaccuracies appeared in the augmented model (Figures 139-141), which might be errors in tracking, but it was not possible to definitely figure that out. Therefore, more markers or other methods of tracking could lead to a better result. Vuforia fusion is working, otherwise the model would fully disappear when the marker is not in the video-stream.

The interactive depiction of different stages of the building-process is working as intended and does not affect the quality of the augmentation.

Different weather situations (clear sky with bright sun or cloudy) have occurred during testing and the results stayed the same over all in these different conditions.

Further research could test different sizes of markers, more markers or other tracking elements like object tracking to see if the results would change, but this would exceed this works scope.

Conclusion

The goal of this thesis was to show possibilities and workflows, which can help to further develop modern techniques for modern plan presentation and communication. The inquiry was if the combination of Augmented Reality with a BIM-model can lead to more coherent presentation of the intended design onsite, and a clearer understanding of the building process, which then should lead to less errors.

In the practical part of this thesis, two projects were presented and a workflow to augment these projects was developed. It was accomplished to develop a workflow that is not bound to a specific BIM-platform, by using the IFC-format. Though in this thesis Archicad was chosen as used BIM-platform. Other platforms were not tested, so it might be possible that different challenges appear, when designing and exporting the model.
Furthermore, a building schedule was added to the IFC-model to be able to get a step-by-step guide of the building process. This was achieved with a CSV-export, which also should be possible with a wide number of scheduling tools.
As AR-developing-platform Unity was chosen, since it seemed a suitable development platform for the above mentioned requirements and requires only limited coding experince.
The link between IFC-export and Unity was achieved with a plugin from the Unity Asset Store, but a second option was also tested with a plugin from the company Tridify, which unfortunately is not working at the current time, since their website is not reachable anymore. The building schedule was added as CSV in Unity and distributed to the IFC-model via code.
Augmentation is achieved with the Vuforia plugin for Unity and initial positioning is done with a marker-based approach. Since Vuforia supports different tracking technologies with Vuforia Fusion and utilizes all available sensors in the device, the model is then tracked with technologies like SLAM. This enables precise tracking even though not having the marker in sight, which is necessary in building construction.

As the first test of the workflow shows, smaller objects in rooms, without a real context, are tracked precisely and are coherent projected along the marker on top of a flat surface. The AR experience is also stable when the marker is not visible, this shows that

Vuforia Fusion is working and SLAM technology is utilized. The addition of the schedule was also a success and the step-by-step sequence of this project was augmented.

In the second test with a bigger project and on a real site, the initial augmentation is working and the augmentation is stable and coherent after slowly scanning around the marker. Since there is a real context to this project some of the digital elements should not be rendered on top of the video stream, because they are behind real objects. This implied the implementation of "occlusion". For this the real objects needed to be surveyed and built in 3D.

When the marker is not thoroughly scanned and the device is moved away too quick, errors occur and the augmentation is not exact. Sometimes errors occur when scanning the marker initially. To solve this problem additional research effort would be required. Also, when moving away from the initial marker, the accuracy of the 3D model weakens in some run-throughs drastically and the differences between augmentation and the already built structure shows that clearly.

Skipping through different stages of the BIM-model is working as intended and does not affect the augmentation negatively. This shows that the connection of schedule and IFC-model is possible with the developed workflow.

The current framework and workflow built around Unity and Vuforia shows that it is possible to further use BIM-models and add another benefit to accurate three-dimensional planning. The use-cases can reach from construction-monitoring to actual construction-methods, although the accuracy has to improve further. The benefit of seeing what must be built in three dimensions could lead to more accurate buildings and less rework. Also applying measurements on site can be done faster and it lets less room for errors, when the accuracy can be improved further.

For an actual, ready-to-use, application there must be made a lot of enhancements and testing. First, the use of more trackable objects could increase the accuracy when moving away from the initial marker. Already built structures could be tested to be used as trackable 3D-objects. Furthermore, in this work it was not tested how big scale projects with lots of data are handled. But the changes which where needed between the two shown projects, indicate that potential redesign of the workflow is likely for handling even bigger sets of data.

Also, the UI must undergo a lot of improvement and there should be a way to import or update projects directly to the application, to ensure the status of the project is up to date and changes are implemented properly.

Another big topic would also be testing with HMDs and the development for an UI for these devices. Since these considerations would exceed this work´s scope, these topics will be left for further research.

All in all, it can be said that Augmented Reality in combination with BIM could be a promising field for construction to be researched and developed further. While building-sites today are still dominated by two-dimensional plans, Augmented Reality could lead into a three-dimensional presentation for building construction and monitoring.

Acknowledgements

First of all, I want to thank my partner Christine for her great support. Countless hours of proofreading and emotional support helped me a lot with this thesis. Another pillar of support was my family, first and foremost my parents Gertrude and Manfred before and during this thesis.

Special thanks go to Heinz Plöderl for supporting me with flexible time management during the final phase of this thesis and my colleagues Kamil Dabrowski and Milica Cvijetic for proofreading the thesis.

Also, special thanks go to Peter Ferschin for valuable input and feedback.

References

[1] (Bew & Richards, 2008)

Bew, Mark; Richards, Mervyn; 2008; The UK maturity Model

[2] (Sacks et al., 2018)

Sacks, Rafael; Eastman, Charles M.; Lee, Ghang; Teichholz, Paul M.; 2018; BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers

[3] https://www.designingbuildings.co.uk/wiki/ISO_19650 (Accessed on 20.05.2023)

[4] (Dörner et al., 2019)

Dörner, Broll, Grimm, Jung; 2019; Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität

[5] (Pangilinan et al. 2019)

Pangilinan, Lukas, Mohan; Maeda; Parisi; 2019; Creatinig Augmented & Cirtual Realities: Theory & Practive for Next-Generation Spatial Computing

[6] (Amsters et al.,2019)

Amsters, Demeester, Slaets; Stevens; Lauwers; 2019; Evaluation of Low-Cost/High-Accuracy Indoor Positioning Systems

[7] (Herout et al. 2012)

Herout, Zacharias, Dubská, Havel; 2012; Fractal marker fields: No more scale limitations for fiduciary markers

[8] (Tautschnig et al., 2017)

Tautschnig, Fröch, Gächter, 2017, Österreichischer BIM-Bericht 2017, Repräsentative Umfrage in österreichischen Unternehmen zu Nutzung, Chancen und Risiken von BIM

[9] (AIA , 2007)

The American Institute of Architects; 2007; Integrated Project Delivery: A Guide

[10] (Kunz & Fischer, 2012)

Kunz, Fischer; 2012; Virtual Design and Construction: Themes, Case Studies and Implementation Suggestions

[11] (Tornincasa & Monaco, 2010)

Tornincasa, Monaco; 2010; The Future and the Evolution of CAD

[12] (Günthner & Borrmann, 2011)

Günthner, Borrmann; 2011; Digitale Baustelle – innovativer Planen, effizienter Ausführen: Werkzeuge und Methoden für das Bauen im 21. Jahrhundert

[13] (Philipp, 2020)

Phillipp; 2020; Architektur – Gezeichnet: Vom Mittelalter bis heute

[14] (Azuma, 1997)

Ronald T. Azuma, 1997, A Survey of Augmented Reality

[15] (Sutherland 1965)

Ivan E. Sutherland; 1965; The Ultimate Display

[16] (Khajavi et. Al., 2019)

Khajavi, Motlagh, Jaribion, Werner, Holmström; 2019; Digital Twin: Vision, Benefits, Boundaries, and Creation for Buildings

[17] (graphisoft.com)

www.graphisoft.com/de/archicad. (Accessed on 20.05.2023)

[18] (Vuforia.com)

www.vuforia.com (Accessed on 20.05.2023)

List of Figures