

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Hybrid GUI and LLM-Based Interaction for Exploring Dense Industrial CAD Models: A Case Study at CERN

João Guimarães Mota



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Alexandre Valle de Carvalho

Second Supervisor: Prof. Daniel Castro Silva

Supervisors at CERN: Fernando Baltasar dos Santos Pedrosa, Drew Mason-Laurence

June 29, 2026

Hybrid GUI and LLM-Based Interaction for Exploring Dense Industrial CAD Models: A Case Study at CERN

João Guimarães Mota

Mestrado em Engenharia Informática e Computação

June 29, 2026

Resumo

Grandes instalações científicas como o CERN trabalham cada vez mais para desenvolver gémeos digitais que combinam modelos 3D da sua infraestrutura com os dados de engenharia que a descrevem, apoiando tarefas como a inspeção e o planeamento da manutenção. O valor de um gémeo digital deste tipo depende da eficácia com que os utilizadores conseguem interagir com os modelos 3D densos que estão no seu cerne, mas essa interação continua a ser difícil. As técnicas convencionais são adequadas para visualizar estes modelos, mas revelam-se insuficientes quando é preciso isolar componentes em conjuntos densos, profundamente hierarquizados e frequentemente sujeitos a oclusão, em que métodos como o ray-casting se tornam pouco fiáveis e localizar um alvo é fastidioso. A informação semântica de que os engenheiros dependem, como identificadores, estado operacional e histórico de manutenção, é também mantida em sistemas separados, obrigando ao cruzamento manual entre a geometria e esses dados. Esta dissertação investiga de que forma uma abordagem de interação híbrida, que combina a interação manual convencional com a consulta conversacional em linguagem natural, pode apoiar a seleção e a filtragem espacial e temática de componentes em modelos 3D industriais densos. A abordagem proposta estende a interface manual do NVIDIA Omniverse, a plataforma que o CERN está a adotar, com um agente conversacional construído sobre um modelo de linguagem de grande dimensão. O agente permite que os componentes sejam localizados e selecionados descrevendo-os em linguagem natural, e recorre aos metadados dos sistemas de engenharia do CERN, de modo a que as seleções possam ser orientadas pelos atributos dos componentes e não apenas pela visibilidade geométrica. Foi desenvolvido um protótipo funcional como um conjunto de extensões do Omniverse Kit e aplicado num modelo da instalação do PS Booster, com suporte a raciocínio espacial para consultas sobre a posição relativa dos componentes. O protótipo foi avaliado num estudo com utilizadores, comparativo e intra-sujeitos, com membros do CERN de diferentes departamentos, que realizaram tarefas de identificação de componentes tanto com a interface manual de referência como com a interface híbrida. O estudo mediu o tempo de conclusão das tarefas, a exatidão, a carga cognitiva através do NASA-TLX e a usabilidade global através da System Usability Scale, a que se somaram questões de preferência comparativa e comentários qualitativos. Com a interface híbrida, os participantes foram mais rápidos (com o tempo mediano aproximadamente reduzido para metade na tarefa mais espacial), reportaram uma menor carga cognitiva (NASA-TLX bruto médio de 25,4 contra 12,1) e classificaram a ferramenta como mais usável (SUS médio de 68,3 contra 78,9), tendo 15 dos 16 participantes preferido dispor de ambas as modalidades em conjunto em vez de qualquer uma delas isoladamente. Estes resultados indicam que a interação conversacional, assente em dados semânticos que o CERN já mantém, constitui um complemento eficaz às ferramentas manuais convencionais, e não um substituto destas, apoiando fluxos de trabalho mais integrados para a inspeção e o planeamento de manutenção no CERN e em instalações de grande escala semelhantes.

Palavras-chave: Interação Humano-Computador, Interação 3D, Interface Conversacional, Modelos de Linguagem de Grande Dimensão, Gémeo Digital, Modelos Industriais Densos, NVIDIA Omniverse

Abstract

Large scientific facilities such as CERN are increasingly working toward digital twins that combine 3D models of their infrastructure with the engineering data that describes it, supporting tasks such as inspection and maintenance planning. The value of such a digital twin depends on how effectively users can interact with the dense 3D models at its core, yet this interaction remains difficult. Conventional techniques are well suited to viewing these models but fall short when users must isolate components in dense, deeply nested, and frequently occluded assemblies, where methods such as ray-casting become unreliable and locating a target is tedious. The semantic information engineers rely on, such as identifiers, operational status, and maintenance history, is also kept in separate systems, forcing manual cross-referencing between the geometry and that data. This dissertation investigates how a hybrid interaction approach, combining conventional manual input with conversational natural-language querying, can support the spatial and thematic selection and filtering of components within dense industrial 3D models. The proposed approach extends the manual interface of NVIDIA Omniverse, the platform CERN is adopting, with a conversational agent built on a large language model. The agent lets components be retrieved and selected by describing them in natural language, and it draws on metadata from CERN’s engineering systems so that selections can be driven by component attributes rather than by geometric visibility alone. A working prototype was developed as a set of Omniverse Kit extensions and deployed on a model of the PS Booster facility, with spatial-reasoning support for queries about the relative position of components. The prototype was evaluated in a comparative within-subjects user study with CERN members from different departments, who performed component-identification tasks under both the baseline manual interface and the hybrid interface. The study measured task completion time, accuracy, cognitive load using the NASA-TLX, and overall usability using the System Usability Scale, complemented by comparative-preference and qualitative feedback. Under the hybrid interface, participants completed tasks faster (median time roughly halved on the most spatial task), reported a lower cognitive load (mean Raw NASA-TLX 25.4 versus 12.1), and rated the tool as more usable (mean SUS 68.3 versus 78.9), and 15 of the 16 participants preferred having both modalities together rather than either alone. These results indicate that conversational interaction, grounded in the semantic data CERN already maintains, is an effective complement to conventional manual tools rather than a replacement for them, supporting more integrated workflows for inspection and maintenance planning at CERN and similar large-scale facilities.

Keywords: Human-Computer Interaction, 3D Interaction, Conversational Interface, Large Language Models, Digital Twin, Dense Industrial Models, NVIDIA Omniverse

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. They include 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

This dissertation is aligned with SDG 8 (Decent Work and Economic Growth), SDG 9 (Industry, Innovation and Infrastructure), and SDG 17 (Partnerships for the Goals). The work contributes to these objectives by improving the productivity of engineering maintenance tasks through technological innovation, advancing the way engineers interact with large-scale research infrastructure, and reflecting an international research collaboration between FEUP and CERN.

The specific Sustainable Development Goals mentioned have the following names:

SDG 8 Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 17 Strengthen the means of implementation and revitalize the global partnership for sustainable development

SDG	Target	Contribution	Performance Indicators and Metrics
8	8.2	By introducing conversational, natural-language interaction over dense 3D engineering models, the prototype raises the productivity of component identification and inspection tasks through technological upgrading.	Reduction in task completion time and perceived cognitive load; task correctness near the ceiling.
9	9.5	A prototype combining conventional manual and conversational interaction advances technological innovation in how engineers locate, inspect and query large-scale research infrastructure.	Usability gain (mean SUS from 68.3 to 78.9); number of system context-switches removed by querying geometry and metadata in one tool.
17	17.16	The work was carried out as an international research collaboration between FEUP and CERN, reflecting the role of multi-stakeholder partnerships in mobilizing and sharing knowledge, expertise and technology.	Knowledge and infrastructure sharing across institutions; prototype deployed and evaluated on CERN's PS Booster model.

Acknowledgements

I would like to begin by expressing my sincere gratitude to my supervisors, Professor Alexandre Valle and Professor Daniel Castro Silva, for their guidance, support, and valuable feedback throughout the development of this dissertation. Their insight and availability were essential in helping me shape this work and overcome the challenges encountered along the way.

I am also deeply grateful to my supervisors at CERN, Fernando Pedrosa and Drew Mason-Laurence, for welcoming me and for supporting me throughout this experience. Their support, technical guidance, and trust allowed me to grow both academically and personally during this period.

A special thank you goes to my two colleagues, João Coutinho and Manuel Alves, who shared this journey to CERN with me. I am thankful for your companionship, for the shared challenges, and for all the moments that made this experience easier and more enjoyable.

I would also like to thank all the people I had the pleasure of meeting during my time at CERN. Each conversation, collaboration, and shared moment contributed to this experience in a meaningful way. I am especially grateful to everyone who participated as a tester in my user study, for their time, interest, and valuable contribution to this work.

To my family and friends, thank you for your love, patience, and constant support. Your encouragement has always given me the strength to keep moving forward, even during the most demanding moments.

João Guimarães Mota

Declaração de Honra

Declaro que a presente dissertação é de minha autoria e não foi previamente apresentada e avaliada nesta ou em outra instituição. As referências a outros autores (afirmações, ideias, pensamentos), assim como a utilização de trabalhos publicados respeitam escrupulosamente as regras da atribuição, e encontram-se devidamente indicadas no texto e nas referências bibliográficas, de acordo com as normas de referência. Tenho consciência de que a prática de plágio ou de autoplágio constitui ilícitos académicos, conforme estabelecido no Código de Ética da U.Porto.

Declaro, ainda, que utilizei ferramentas de Inteligência Artificial para a realização de parte(s) da presente dissertação, e essa utilização e os seus resultados estão devidamente assinalados e foram objeto de cuidada verificação para deteção de erros, imprecisões, atribuições infundadas ou outras formas de enviesamento de conteúdos e argumentos, bem como de determinação de autoria.

João Guimarães Mota



“It worked yesterday.”

Anonymous

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Problem	3
1.4	Objectives	4
1.5	Document Structure	5
2	Background	6
2.1	CERN's Engineering Data Landscape	6
2.1.1	Engineering Systems and Model Repositories	6
2.1.2	Characteristics of Dense Engineering Models	7
2.1.3	Toward an Integrated Digital Twin	9
2.2	Fundamentals and Evolution of 3D Interaction	9
2.2.1	Complexity, Density, and Occlusion	10
2.2.2	Semantic Mapping and Metadata Retrieval	12
2.2.3	Conversational Interfaces and Spatial Reasoning	13
3	Related Work	15
3.1	Exploratory Review Process	15
3.1.1	Identification Phase	15
3.1.2	Screening Phase	17
3.1.3	Review and Analysis	17
3.2	State of the Art	19
3.2.1	Interaction Challenges in Dense Industrial 3D Models	20
3.2.2	Manual Selection Techniques	26
3.2.3	Semantic Enrichment and Metadata-Driven Interaction	34
3.2.4	Natural-Language Interaction for 3D Environments	39
3.2.5	Hybrid Interaction: Combining Manual and AI-Assisted Techniques	45
3.3	Synthesis: From the Literature to the Approach	49
4	Problem Approach	51
4.1	Detailed Problem Definition	51
4.2	Proposed Solution	52
4.3	Choice of a Conversational Interface	54
4.4	Risk Analysis	55

5	Implementation	58
5.1	The Source Data and Its Challenges	58
5.2	A Searchable Representation of the Scene	59
5.3	Resolving Natural Language to Prim Paths	61
5.4	Spatial Reasoning	66
5.5	The Conversational Control Layer	68
5.6	The Manual Interface	72
5.6.1	The Info Window	73
5.6.2	The CAD Tree	73
5.7	External Data Integration	74
5.8	The Assembled Interface	75
5.9	Development Challenges	76
5.10	Summary	78
6	Experimental Protocol	80
6.1	Evaluation Objectives	80
6.2	Participants and Recruitment	81
6.3	Ethical Considerations	81
6.4	Questionnaire Design	82
6.4.1	Pre-Test Questionnaire	82
6.4.2	Task-Specific Questionnaire	82
6.4.3	Post-Test Questionnaire	84
6.5	Test Moderator	85
6.6	Apparatus	86
6.7	Experiment Procedure	86
6.7.1	Procedure	86
6.7.2	Completion Criteria	87
6.8	Evaluation Methodology	87
6.8.1	Research Design	87
6.8.2	Objective Performance Metrics	88
6.8.3	Subjective Perception Metrics	88
6.8.4	Data Analysis	88
6.9	Pilot Test	89
6.10	Threats to Validity	89
6.11	Summary	90
7	Results	92
7.1	Sample Characterization	92
7.2	Task Performance	94
7.2.1	Correctness	94
7.2.2	Completion Time	95
7.2.3	Hints Required	96
7.3	Cognitive Load	97
7.4	System Usability Scale	98
7.5	Comparative Preference	99
7.6	Order and Task-Set Effects	100
7.7	Correlation With Demographics	101
7.8	Qualitative Feedback	104
7.9	Summary	105

8	Conclusions	107
8.1	Final Remarks	107
8.2	Future Work	108
	References	110
A	User Testing Script	116
A.1	Introduction	116
A.2	Informed Consent Form	117
A.2.1	Introduction	117
A.2.2	Voluntary Participation	117
A.2.3	Experiment Overview	117
A.2.4	Data Collection	117
A.2.5	Data Usage and Confidentiality	117
A.2.6	Contact Information	118
A.2.7	Consent	118
A.3	Pre-Test Questionnaire	118
A.4	Tool Orientation	119
A.4.1	Interface Configuration H0: Baseline Omniverse	119
A.4.2	Interface Configuration H1: Hybrid AI-Assisted	119
A.5	Task-Specific Questionnaire	119
A.5.1	Task Set A	120
A.5.2	Task Set B	120
A.6	Post-Test Questionnaire	121
A.6.1	NASA Task Load Index	121
A.6.2	System Usability Scale	121
A.6.3	Comparative Preference	122
A.6.4	Open Feedback	122

List of Figures

2.1	Tunnel-scale digital-twin view of CERN infrastructure.	8
2.2	Theoretical model of the Heisenberg effect in spatial interaction.	10
2.3	Cutaway and dynamic-transparency rendering for occlusion management.	11
2.4	Hierarchy-aware selection in HOM3R.	12
2.5	ChatTwin interface layout and components.	13
3.1	Distribution of reviewed publications by year and theme.	19
3.2	Depth ambiguity at the cursor in stereoscopic mouse-based 3D selection.	21
3.3	Completion time by cursor technique, density, and visibility.	22
3.4	Structural occlusion during interaction.	25
3.5	Comparison of ray-based selection and depth-oriented refinements: ray-cast, Depth Ray, RayCursor.	27
3.6	Gimlenses multi-view detail-in-context technique for 3D CAD models.	28
3.7	Virtual mirror selection under occlusion.	28
3.8	SQUAD interaction flow (sphere-cast + disambiguation).	29
3.9	Progressive refinement via discrete zoom	29
3.10	CloudLasso structure-aware selection pipeline.	31
3.11	Semantic influence field in Semantic Pointing	31
3.12	Explosion graph representation for interactive exploded views.	32
3.13	Projection-based automatic exploded view for assembly clarity.	33
3.14	Exploded view diagrams for 3D grids	34
3.15	Conceptual diagram of semantic pointing: relevance biases selection.	35
3.16	Task-specific BIM visualization configurations	36
3.17	CAD3A web interface layout and components.	37
3.18	Conceptual framework for adaptive personalized 3D user interfaces	38
3.19	ScanQA task framing: question answering with grounded 3D bounding boxes. . .	40
3.20	Chat-Scene overall model architecture	40
3.21	LL3DA interaction pipeline with multimodal grounding.	43
3.22	CoDAv2 overview for open-vocabulary 3D object detection	44
3.23	Multimodal interaction strategies in ASSISTVR.	47
4.1	Overview of the proposed hybrid architecture	53
5.1	Construction of the index from the loaded USD stage.	60
5.2	The three-stage resolution pipeline and the five-category result classification. . .	62

5.3	Comparison between an axis-aligned bounding box and an oriented bounding box around the same elongated component. The AABB inflates around the rotated component and can include nearby geometry that is not actually adjacent, whereas the OBB follows the component orientation and provides a tighter spatial filter for proximity queries.	67
5.4	Result demotion: the assembly-level OBB is used as a coarse filter, and only those leaf-mesh children in actual contact are retained.	68
5.5	The dispatch hierarchy applied by the modifier between the language model and the backend.	71
5.6	A connectivity query whose reference is the current selection. The vacuum chamber (CHAMBRE A VIDE) was selected in the viewport before the message was sent, so the deictic “this object” is bound to that selection. The agent’s reply names CHAMBRE A VIDE explicitly and lists the six connected components, which are outlined in orange in the viewport.	72
5.7	The Info Window for a selected component, showing the assembly card, scheduled tasks, and engineering metadata.	74
5.8	The same component shown in the default Omniverse stage tree (left) and in the CAD Tree (right). The default tree lists raw USD names alongside geometry and material nodes, whereas the CAD Tree presents each component under its PLM description and adds per-node action buttons.	75
5.9	The prototype running on the PS Booster, with the viewport (left), the CAD Tree and Info Window (right), and the conversational chat panel (bottom). The chat shows a natural-language query resolved to a single component, which is highlighted in the viewport, selected in the tree, and shown with its scheduled task in the Info Window.	76
7.1	Participant demographics.	93
7.2	Self-reported familiarity with the PS Booster facility.	93
7.3	Self-reported frequency of conversational AI tool use.	94
7.4	Correctness rate per task under each condition, annotated with the number of correct participants out of 16.	95
7.5	Completion time per task under each condition. Boxes show the interquartile range and median; whiskers show the range excluding outliers.	96
7.6	Mean Raw NASA-TLX rating by dimension and condition.	97
7.7	Mean SUS score under each condition.	98
7.8	Responses to “How predictable did you find the chat-assisted interface?”.	100

List of Tables

3.1	Primary access sources for the publications included in the reviewed set.	17
3.2	Inclusion and exclusion criteria for the screening phase.	18
3.3	Publication types within the reviewed set.	18
3.4	Virtual task-environment factors commonly used to evaluate 3D selection techniques.	24
3.5	Representative capabilities and failure modes of LLM-mediated 3D interaction. .	42
3.6	Summary of adaptive selection approaches and deployment constraints.	46
3.7	Design requirements for trustworthy hybrid interaction for semantic selection in dense 3D models.	48
4.1	Summary of likelihood and impact of each identified risk.	56
5.1	Example message handled at each level of the dispatch hierarchy.	70
6.1	Counterbalancing of hypothesis order and task-set assignment across the four participant groups.	88
7.1	Mean outcomes for each of the four counterbalancing configurations. Each configuration lists the interface used in the first block and the interface used in the second, together with the task set paired with each.	101
7.2	Mean outcomes by self-rated 3D experience, baseline vs. hybrid.	102
7.3	Mean outcomes by age group, baseline vs. hybrid.	102
7.4	Mean outcomes by PS Booster familiarity, baseline vs. hybrid.	103

List of Abbreviations

2D	Two-dimensional
2DOF	Two Degrees of Freedom
3D	Three-dimensional
AABB	Axis-Aligned Bounding Box
AI	Artificial Intelligence
API	Application Programming Interface
BIM	Building Information Modeling
CAD	Computer-Aided Design
CERN	European Organization for Nuclear Research (from French <i>Conseil Européen pour la Recherche Nucléaire</i>)
CI	Confidence Interval
DT	Digital Twin
EAM	Enterprise Asset Management
EDMS	Engineering Data Management Service
HCI	Human-Computer Interaction
IMPACT	Intervention Management Planning and Coordination Tool
JSON	JavaScript Object Notation
LHC	Large Hadron Collider
LLM	Large Language Model
NASA-TLX	National Aeronautics and Space Administration Task Load Index
OBB	Oriented Bounding Box
PLM	Product Lifecycle Management
PS	Proton Synchrotron
REST	Representational State Transfer
SAT	Separating Axis Theorem
SDK	Software Development Kit
SSO	Single Sign-On
SUS	System Usability Scale
UI	User Interface
USD	Universal Scene Description
VE	Virtual Environment
VR	Virtual Reality

Chapter 1

Introduction

Digital Twin technologies have emerged as powerful frameworks for connecting physical systems with high-fidelity virtual representations including geometric, historical, and operational data [28]. By offering comprehensive visualization, simulation, and analysis, digital twins improve monitoring, operation, and maintenance of complex infrastructures and enable more informed decision-making [62]. Their growing relevance in both scientific and industrial domains reflects a broader shift toward integrated system knowledge and modern data-driven engineering.

Large-scale facilities have particular difficulties in this environment because of the density, heterogeneity, and hierarchical complexity of their engineering models. As emerging computer graphics frameworks and open 3D scene standards are explored to support the real-time, collaborative visualization of massive infrastructures, new opportunities arise for more intuitive interaction with these models and for bridging geometric data with the semantic information stored across several engineering systems.

In order to facilitate workflows, such as asset identification, fault localization and maintenance planning, where geometric objects can be meaningfully linked with external metadata, this dissertation explores how engineers can efficiently browse, examine, and identify specific components in dense industrial 3D models through a hybrid interaction approach that combines conventional desktop input devices (mouse and keyboard) with conversational, natural-language querying.

The following sections provide the context, motivation, and problem framing behind this research, introducing the objectives that guide the dissertation.

1.1 Context

CERN¹, the European Organization for Nuclear Research, is widely regarded in the scientific community as the preeminent international facility for fundamental physics research. Located near Geneva, straddling the border between France and Switzerland, it was established in 1954 and has subsequently turned into a leading example of global scientific cooperation. Today, CERN unites thousands of scientists and engineers from around the world to push the frontiers of science

¹More information available online from: <https://home.cern>

and technology, provide unique particle accelerator facilities, foster technological innovation, train new generations of researchers, and engage society in the values of science.

Some of the biggest and most sophisticated scientific facilities in the world, including the Large Hadron Collider² (LHC), its related accelerator chain, and other experiments, are located at CERN. These machines consist of millions of components distributed over tens of kilometers of tunnels, supported by extensive cryogenic, electrical, mechanical, and control systems. Operating and maintaining these facilities requires sophisticated engineering information management, as millions of components must be monitored throughout their lifecycle, from design and manufacturing to operation and maintenance.

Geometric models, technical documentation, intervention schedules, and asset data are stored in a number of specialized systems that transmit the engineering data supporting these machines. Although each system serves a specific operational function, it is challenging to gain a comprehensive understanding of the accelerator's intricate assembly due to their separation.

These systems represent a substantial and well-maintained body of engineering information, accumulated across the full lifecycle of the accelerator complex. The way this information is organized, distributed across specialized systems and held separately from the 3D representations of the machines it describes, is characteristic of how large-scale scientific infrastructure is managed, and it shapes how engineers work day to day. It is within this setting that the interaction challenges addressed by this dissertation arise.

1.2 Motivation

CERN has increasingly adopted industry platforms for large-scale 3D visualization, including NVIDIA Omniverse, to support the interactive inspection of complex engineering assemblies. Such platforms provide high-performance rendering and collaboration capabilities and are being considered as enabling infrastructure for future Digital Twin initiatives. As these workflows come to depend more heavily on accurate 3D representations, how engineers actually explore and interact with those assemblies becomes decisive for whether model-based tasks can be carried out efficiently and reliably.

In many of these workflows, however, the 3D view remains primarily geometric: it shows the shape and position of components, but the operational and asset information engineers depend on is still reached through separate systems. The value of a shared 3D environment is therefore only partly realized, since the step from seeing a component to understanding its status, history, and constraints still happens elsewhere.

Improving the interaction between 3D geometry and engineering metadata is the central motivation for this work. Bringing relevant asset information closer to the point of interaction in the 3D environment, and reducing the reliance on scattered external systems, would make everyday engineering tasks such as asset identification, fault localization, and maintenance planning more

²For additional details on the LHC: <https://home.cern/science/accelerators/large-hadron-collider>

direct, and is a prerequisite for the more integrated digital engineering practices that digital twins promise.

Although this work is developed and evaluated at CERN, the problem it addresses is not specific to particle physics. Any organization operating large physical infrastructure, such as process plants, shipyards, power stations or manufacturing lines, maintains dense 3D models whose components are described in engineering systems that sit apart from the geometry. CERN is used here as an instance of this broader class. It combines extreme geometric density, deep assembly hierarchies and a mature engineering data ecosystem, a combination that makes the interaction difficulties particularly acute and the available metadata rich enough to build on.

1.3 Problem

Although models built on OpenUSD (Universal Scene Description, the open 3D scene format used by NVIDIA Omniverse) provide a modern and scalable foundation for visualizing accelerator components, interacting with these representations remains challenging due to their density, hierarchical structure, and frequent occlusion. The use of OpenUSD within NVIDIA Omniverse is treated in this dissertation as a contextual constraint imposed by CERN's platform decisions, rather than as an arbitrary technology choice. Two difficulties, in particular, stand in the way of using these models effectively.

The first arises from the limitations of conventional 3D interaction techniques when applied to high-density engineering environments. Prior research has shown that methods such as ray-casting tend to degrade in accuracy and reliability when objects overlap, when components are deeply nested, or when substantial portions of the model are occluded [65]. In these situations, users are often forced into repetitive camera adjustments simply to expose targets, losing both efficiency and spatial context. Desktop-based studies similarly report that without specialized visibility-support mechanisms such as exploded views, cutaway rendering, or transparency control, users encounter significant barriers when attempting to access hidden or tightly interlocked components [61]. These observations collectively indicate that traditional geometric interaction paradigms do not scale effectively to models containing thousands of interconnected elements, such as those found in CERN's accelerator complex.

The second difficulty is that the semantic information describing these components lives apart from the geometry. Detailed metadata, such as identifiers, operational status, maintenance history, and safety information, is rigorously maintained in separate systems like IMPACT (CERN's intervention and maintenance management system) and EAM (Enterprise Asset Management), yet it remains external to the geometric models used for spatial reasoning. Because these datasets live in separate environments, engineers must manually move between them whenever they need to understand both the location and the context of a component. Even when the metadata itself is complete and consistent, the lack of a direct link to the 3D representation forces users to perform repetitive lookups and cross-confirm details, adding cognitive and operational overhead to inspection and planning activities.

Taken together, these factors define the concrete problem this dissertation addresses: interaction with dense 3D engineering models at CERN is inefficient because conventional techniques do not scale to high-density, occluded assemblies, and because the semantic records needed to make sense of those models are held separately from the geometry. This interaction bottleneck is what currently prevents CERN’s existing information from being fully actionable for inspection, maintenance planning, and experiment preparation.

1.4 Objectives

The primary objective of this dissertation is to determine which interaction and navigation techniques most effectively support spatial and thematic exploration in dense 3D industrial models represented within NVIDIA Omniverse. The work is guided by the following research question:

- **RQ:** To what extent does a hybrid interaction approach, combining conversational natural-language input with conventional manual interaction, improve the spatial and thematic selection and filtering of individual objects within complex and dense 3D industrial models, compared to conventional manual interaction alone?

To answer this question, the dissertation proposes and evaluates an approach designed to:

- Integrate an AI-assisted conversational interface, utilizing Large Language Models, to support natural-language querying and context-aware selection, thereby reducing the manual effort required to locate specific assets.
- Enable semantic-driven selection and filtering, allowing users to isolate components based on metadata retrieved from external engineering systems rather than solely on geometric visibility.
- Explore visibility-oriented interaction techniques for addressing spatial occlusion and improving access to internal structures, such as interactive exploded views. This is an exploratory rather than central objective.

The proposed approach builds upon the extensibility of the Omniverse Kit SDK and the OpenUSD framework to develop a modular interaction layer capable of supporting multiple levels of model exploration, ranging from conventional manual selection to more advanced visual and spatial interactions, and extending toward a semantic layer that incorporates filtering mechanisms and AI-assisted selection.

To evaluate the effectiveness of these interaction strategies, the work adopts a comparative methodology grounded in realistic engineering scenarios, with a particular emphasis on tasks that resemble maintenance planning workflows. It contrasts a conventional manual baseline with the hybrid AI-assisted approach, assessing both quantitative measures, such as task completion time, selection accuracy, and error occurrence, and user-perceived cognitive workload and overall clarity of interaction. The central expectation is that the hybrid condition provides more reliable support

for identifying and selecting components in dense assemblies, particularly where occlusion and repeated parts increase ambiguity and make purely manual selection costly.

In doing so, this dissertation contributes to the field of Human-Computer Interaction in engineering by clarifying the conditions under which hybrid approaches, combining spatial interaction, semantic filtering, and AI assistance, offer advantages in dense 3D environments, and by supporting more effective exploration and interpretation of complex engineering assemblies for maintenance workflows and decision-making within CERN’s large-scale scientific infrastructure.

1.5 Document Structure

This document is organized into seven additional chapters after this introduction.

Chapter 2 provides background on the CERN engineering context and the characteristics of dense industrial 3D assemblies, including the role of OpenUSD and NVIDIA Omniverse within ongoing digital engineering initiatives.

Chapter 3 reviews related work and the state of the art in interaction and selection for complex 3D scenes, with emphasis on occlusion management, navigation, semantic selection, and recent conversational or language-mediated approaches to 3D interaction.

Chapter 4 defines the problem in detail and presents the proposed approach, setting out the proposed solution, the boundaries of the contribution, and the associated risk analysis.

Chapter 5 describes the implementation of the prototype, covering the searchable representation of the scene, the natural-language resolver, the spatial reasoning, the conversational control layer, and the manual interface.

Chapter 6 presents the experimental protocol, including the participants, the questionnaire and task design, the apparatus, and the evaluation methodology.

Chapter 7 reports and discusses the results of the user study across task performance, cognitive load, usability, and comparative preference.

Finally, Chapter 8 draws conclusions, answering the research question, reflecting on the limitations of the work, and outlining directions for future research.

Chapter 2

Background

To contextualize the themes explored in this dissertation, it is crucial to look at both the engineering environment where the work is done and the technical ideas that support how people interact with large-scale 3D models today. This chapter provides an overview of these foundational elements which sets the stage for understanding the systems, data structures, and interaction paradigms referenced in subsequent chapters.

Section 2.1 presents CERN’s engineering data landscape, outlining the main systems used to manage geometric and semantic information, and describing the characteristics of the complex assemblies that form the accelerator infrastructure. Section 2.2 traces the evolution of methods for navigating and interacting with dense digital models, emphasizing fundamental concepts such as occlusion and spatial complexity, the role of semantic metadata in supporting structured exploration, and recent developments in natural-language and spatial reasoning interfaces.

2.1 CERN’s Engineering Data Landscape

CERN operates the world’s largest and most complex scientific instruments, including the Large Hadron Collider (LHC), a 27-kilometer ring of superconducting magnets and accelerating structures.¹ Extreme density and scale characterize the engineering reality of these facilities: millions of highly specialized components, from massive cryostats to microscopic sensors, are crammed into kilometers of tunnels. During operation, direct physical inspection is challenging or impossible because these components are frequently found in radioactive or physically inaccessible environments. Consequently, engineers and technicians rely on digital representations to plan interventions, manage assets, and understand the machine’s configuration.

2.1.1 Engineering Systems and Model Repositories

At CERN, mechanical design and technical documentation have historically been managed via a specialized Product Lifecycle Management (PLM) system, alongside the legacy CAD-drawing

¹More information available online from: <https://home.cern/about>

archives. As part of a strategic modernization effort, recent announcements confirm that the long-standing CERN Drawing Directory (CDD), originally used to store over half a million drawings since 1996, was decommissioned at the end of 2024 in favor of a unified PLM platform that will handle all CAD data, including CATIA-based mechanical models.²

Central to this ecosystem is the CERN Engineering Data Management Service (EDMS). This platform has developed to integrate asset tracking capabilities from the EAM (Enterprise Asset Management) system with PLM functionalities, particularly for managing CAD files and complex assemblies.³ By connecting design geometry, version control, documentation, and lifecycle metadata across all stages of design, construction, operation, and maintenance, this integration demonstrates CERN's commitment to preserving the "digital thread".⁴

Despite these integration efforts, the independent evolution of these systems has resulted in separate data silos. Many mechanical parts are defined in CATIA or similar CAD tools, while crucial semantic information, such as maintenance records, part identifiers, and work-permit data, is stored in asset databases or document archives. Even though the migration to a unified PLM/EDMS environment aims to converge these datasets, engineering workflows often still need manual cross-referencing between geometric models and operational metadata. This separation imposes significant constraints on the interactive exploration, querying, and annotation of 3D models, particularly within the context of large-scale complex assemblies.

2.1.2 Characteristics of Dense Engineering Models

The 3D assemblies that constitute CERN's infrastructure exemplify industrial-scale engineering models. These assemblies are distinguished by their exceptionally high geometric fidelity, frequent use of repeating subassemblies, and deep hierarchical structures. In order to make its accelerator complex available as a digital twin, CERN has started a major project that involves a comprehensive reconstruction of tunnels and experimental zones in a three-dimensional setting (Fig. 2.1).

When subassemblies are integrated into complete installations, these assemblies frequently combine thousands to millions of discrete components. Particular difficulties with interaction and visualization are brought about by this geometric density, being the complexity of the hierarchical structure the most important one. Engineering-centric hierarchies are used to arrange parts, subassemblies, and systems, usually moving from individual components to modules, sections, and eventually entire tunnels. This organization produces deeply nested trees that pose significant challenges for meaningful traversal in the absence of well-structured metadata.

Furthermore, there is a lot of visual clutter and occlusion in these settings, causing components to frequently overlap or hide each other in densely crowded regions such as detection zones, cryogenic modules, and tunnels. This spatial congestion complicates the selection, isolation, and

²The official announcement regarding the phase-out of the CDD is available at: <https://home.cern/news/announcement/computing/cern-drawing-directory-cdd-be-phased-out-end-2024>

³Further details and documentation regarding the EDMS platform can be accessed via: <https://edms-service.web.cern.ch/faq/EDMS/pages>

⁴More insights into CERN's Information Management can be found at: <https://en.web.cern.ch/group/im>

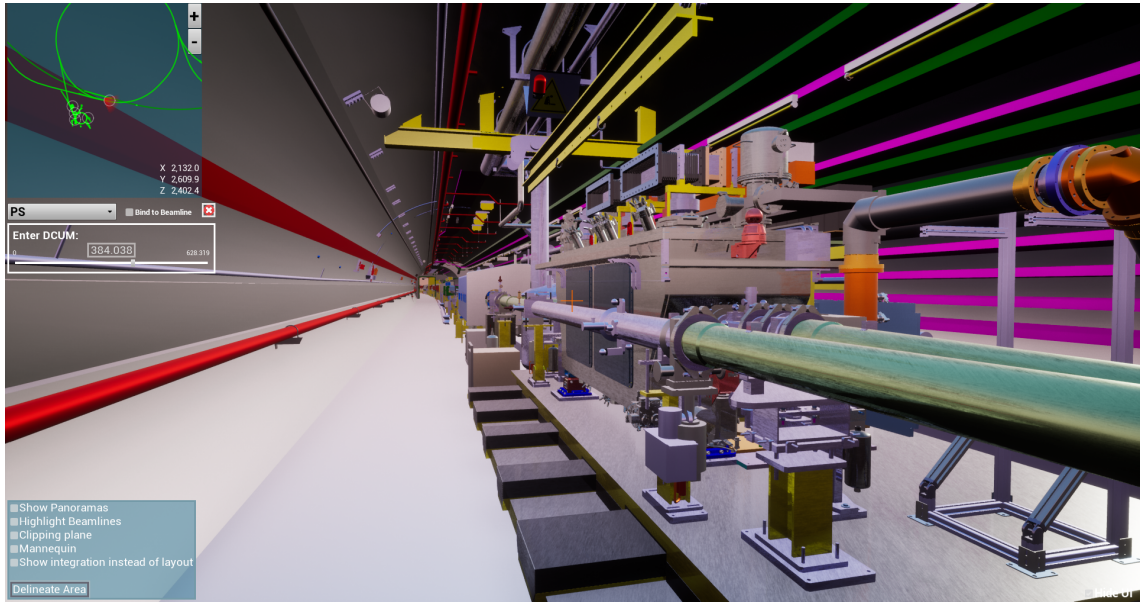


Figure 2.1: Tunnel-scale digital-twin view illustrating the geometric density and visual clutter typical of CERN installations.

inspection of individual parts, making conventional ray-casting or direct-picking techniques ineffective. These issues are prevalent in large-scale CAD and BIM environments and are a frequent topic of discussion within the domain of 3D interaction [20].

CERN’s accelerator complex is built as a chain of machines, each raising the energy of the particle beam before passing it to the next. One of these is the Proton Synchrotron Booster, commonly referred to as the PS Booster, which receives beams from the linear accelerator and accelerates them before injecting them into the Proton Synchrotron. Its four superimposed synchrotron rings sit in a tunnel densely filled with magnets, beam pipes, supporting structures and cabling, arranged in repeated periods around the ring. The PS Booster illustrates the characteristics described above with particular clarity, combining repeated near identical assemblies, a deep hierarchical structure and heavy occlusion within a compact volume.

A further challenge arises from the heterogeneity of data origins and modeling standards. Many teams, contractors, and outside vendors have provided CAD models that follow a variety of standards for units, naming conventions, and levels of detail throughout decades of operation. Research on large-scale digital twin initiatives, this variability creates significant obstacles for version control, dataset integration, and ensuring consistency in metadata. Consequently, interaction paradigms that are more sophisticated than those usually used for smaller CAD models or simpler 3D scenarios are required when navigating, viewing, and interacting with full-scale accelerator models for tasks like inspection, maintenance planning, or training [18].

2.1.3 Toward an Integrated Digital Twin

CERN has progressively acknowledged the drawbacks of keeping geometry, documentation, and asset information in distinct systems, particularly as engineering tasks require more extensive cross-referencing between design data, installation records, and maintenance workflows. In response, the organization has presented a long-term vision that aligns with digital twin concepts, where geometric models and semantic data could ultimately be accessed via a unified virtual representation of the accelerator infrastructure. Although this vision is still emerging, it represents a coordinated institutional attempt to decrease fragmentation and enhance the uniformity of engineering data throughout its lifecycle.

The upgrading of the geometric data pipeline is a key element of this approach. CERN has officially embraced NVIDIA Omniverse as its platform for real-time 3D visualization and collaborative engineering spaces, along with the Universal Scene Description (USD) format as the standard method for representing and sharing geometry. As a result, legacy CAD assemblies, mainly from CATIA and historical drawing repositories, are being converted into USD to facilitate unified visualization, multi-user inspection, and future simulation-based workflows within Omniverse.

Currently, however, semantic and operational metadata remain separate from USD-based scenes. Dedicated systems, such as CERN's PLM/EDMS ecosystem and its asset-management tools, continue to house data including equipment identification, maintenance records, asset hierarchies, and operating limits. While it has not yet been implemented in the deployed pipeline, connecting this metadata to geometric objects in Omniverse represents a significant step toward the integration process that will lead to a fully functional digital twin.

As a result, interactive systems created for this setting must function in a hybrid environment: geometry is available through a modern Omniverse-USD pipeline, while semantic information must still be referenced through external systems. The coexistence of legacy datasets, evolving pipelines, and emerging digital twin ambitions outlines the practical limitations and possibilities in which new interaction techniques have to operate, a context that directly motivates the challenges addressed in later chapters.

2.2 Fundamentals and Evolution of 3D Interaction

Interaction with three-dimensional environments has been an active research topic within Human-Computer Interaction for several decades, motivated by the increasing availability of complex geometric data and the limitations of traditional two-dimensional interfaces [34]. Early work on 3D interaction focused primarily on navigation and manipulation in comparatively simple or controlled virtual scenes [11].

However, as 3D models grew in scale and fidelity, particularly in engineering, CAD, and scientific visualization contexts, the limitations of conventional interaction metaphors became increasingly apparent. Prior research therefore emphasizes the challenges introduced by visual density

(clutter) and semantic heterogeneity, which collectively motivate the development of more robust selection, filtering, and exploration techniques [12, 48].

Interaction is no longer restricted to geometric inspection in scientific and industrial domains, such as large-scale engineering assemblies. Instead, it must facilitate tasks that call for identifying specific components, comprehending their spatial relationships, and connecting them to external semantic data [52]. With special focus on occlusion management [23], semantic enrichment, and evolving conversational interfaces [43], this section describes the fundamental interaction challenges presented by dense 3D environments and traces an evolution of techniques designed to address them.

2.2.1 Complexity, Density, and Occlusion

The combined impacts of geometric density and occlusion are one of the most enduring problems in 3D interaction [9]. As models scale from tens to thousands or millions of components, overlapping objects in screen space cause visual congestion and ambiguity during selecting. Studies on 3D object selection consistently report that standard ray-casting techniques degrade significantly under such conditions [30]. This is because even little changes in cursor location or camera orientation might lead to unintentional target acquisition or selection failure [65]. This phenomenon, often described as the “Heisenberg effect” [69] of selection, forces users into repeated viewpoint adjustments, increasing both task time and cognitive load. Figure 2.2 summarizes the Heisenberg effect and its relationship to hand jitter and target size.

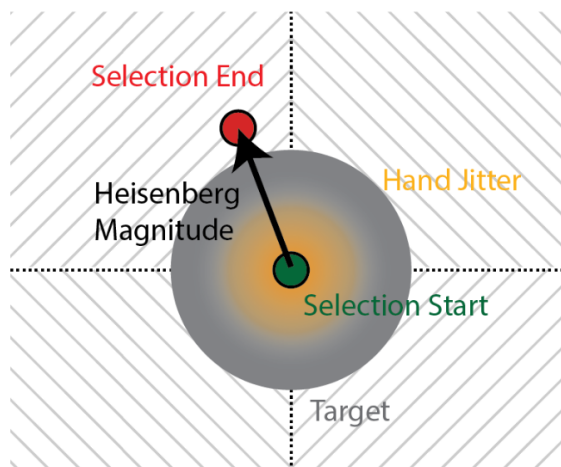
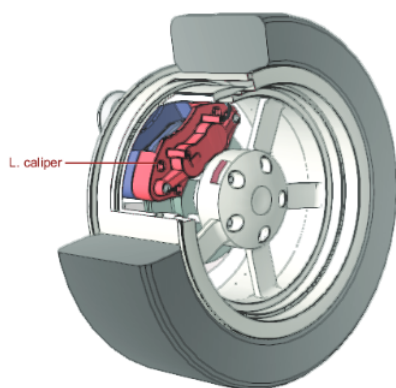


Figure 2.2: Theoretical model of the Heisenberg effect in spatial interaction, illustrating the offset between selection start and end, its relationship to hand jitter, and the definition of a Heisenberg error (a selection that starts inside the target but ends outside due to input disturbance). Taken from Wolf et al. [69].

A significant amount of research has investigated different selection methods and occlusion strategies to overcome these problems [2]. Surveys of 3D selection approaches have developed taxonomies that classify methods by their underlying metaphor, that is, the real-world action the interaction imitates, such as pointing along a ray, sweeping a volume, or moving a cursor, and

by the control mechanism they use, in order to show how human motor limitations and visual input constraints fundamentally affect performance in crowded environments [2]. Building on these taxonomies, empirical research shows that when targets are small, distant, or obscured, accuracy can be greatly improved by techniques such as volume-based cursors, which select within a region of space rather than at a single point; depth-aware rays, which help disambiguate between objects lying along the same line of sight; and progressive refinement, which narrows a large set of candidates to the intended target over successive steps, though often at the cost of additional interaction steps [37, 66].

Parallel to advances in selection, visualization research has developed a rich design space for occlusion management. Recurring design patterns, such as numerous synchronized views [56], virtual X-ray rendering [21], cutaways [41], projection distortions [19], and interactive exploded views, are identified by taxonomies of occlusion management systems [23]. Figure 2.3 contrasts cutaway rendering with dynamic transparency as two representative occlusion-management strategies.



(a) Cutaway rendering for occlusion management, revealing hidden structures while preserving spatial context. Taken from Li et al. [41].



(b) Dynamic-transparency rendering for occlusion management, revealing hidden structures while preserving spatial context. Taken from Elmqvist, As-sarsson, and Tsigas [21].

Figure 2.3: Comparison of cutaway and dynamic-transparency rendering techniques for occlusion management.

For example, obstruction-free lensing implements interactive focus+context distortions to locally reduce occlusion without removing the surrounding context [64], while fast cutaway methods compute view and distance dependent cutaways to reveal hidden structures at interactive rates [36]. These approaches aim to reveal hidden structures while preserving spatial context, which is essential for comprehending intricate assemblies. Because they reduce occlusion without permanently changing the underlying model, interactive exploded views and virtual disassembly approaches have proven particularly useful for inspecting densely interlocked components in mechanical systems [40, 70]. These techniques, however, frequently depend on pre-established hierarchies or presumptions on part separability, which might not always hold true in heterogeneous industrial datasets.

Despite these developments, the challenges presented by dense engineering models cannot be entirely solved by a single method. Research increasingly suggests that, depending on the unique characteristics of the task and scene, effective interaction in such environments requires the combination of several strategies, such as progressive refinement for accurate selection and focus+context visualization for spatial understanding. This insight directly influences the hybrid interaction approaches explored later in this dissertation.

2.2.2 Semantic Mapping and Metadata Retrieval

Although it is a significant interaction barrier, geometric complexity is merely one aspect of the problem in industrial contexts. The capacity to link geometric items to semantic data, such as IDs, functional responsibilities, maintenance status, or system membership, is equally crucial. This metadata is not intrinsically integrated into the 3D representation used for visualization in major engineering organizations. Instead, it is usually kept in external databases or PLM systems.

According to research in semantic-driven visualization and Building Information Modeling (BIM), more structured and task-oriented exploration is made possible by integrating semantic attributes into 3D settings [27]. By allowing users to separate subsets of components according to non-geometric criteria, techniques like semantic filtering, color coding, and rule-based visualization configurations greatly reduce visual clutter and enhance interpretability [33]. Coupling geometry with explicit structure metadata might provide a more effective traversal of large assemblies, as shown by systems that synchronize spatial views with hierarchical or tabular representations. Industrial product-model viewers such as HOM3R explicitly target complex hierarchical assemblies, reinforcing the practical importance of hierarchy-aware interaction in large models [29]. Figure 2.4 illustrates hierarchical selection at different levels in HOM3R.

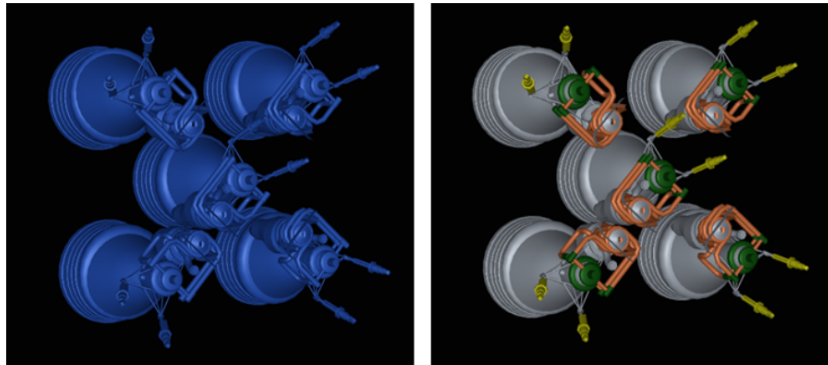


Figure 2.4: Hierarchy-aware selection in HOM3R at different levels of the product hierarchy. The viewer supports selecting parts or areas at different hierarchy levels and highlighting them with color to convey additional information. Taken from González-Toledo et al. [29].

However, the quality and accessibility of information have a significant impact on how well semantic interaction works. Formal semantic models can be used to direct both visualization and interaction in ontology-driven and knowledge-based approaches. Nevertheless, these systems frequently assume the availability of complete and well-defined ontologies [59]. Semantic data may

be dispersed across several systems, inconsistent, or incomplete in practice, particularly in legacy or evolving engineering contexts. As a result, interaction techniques frequently have to function in hybrid environments where semantic enrichment is either partially or externally referenced but geometry is easily accessible [10].

For digital twin applications, the gap between semantic richness and geometric representation is a crucial obstacle. The lack of direct links to 3D objects forces users to do manual cross-referencing, diminishing the potential efficiency gains of semantic visualization even in cases when detailed metadata is available. Addressing this disconnect requires interaction mechanisms that can take advantage of available identifiers and partial metadata while remaining robust to heterogeneity and incompleteness.

2.2.3 Conversational Interfaces and Spatial Reasoning

Recent advances in artificial intelligence, particularly in Large Language Models (LLMs), have introduced new interaction paradigms for accessing and interacting with complex digital environments [45]. Figure 2.5 shows an example of such a conversational interface integrated with a visual front-end. Conversational interfaces allow users to express intent using natural language, perhaps avoiding low-level geometric manipulation in the context of 3D interaction. Users can perform semantic queries, such as finding all components of a particular type or isolating assets connected to a particular system or place, instead of directly navigating to and choosing individual objects.

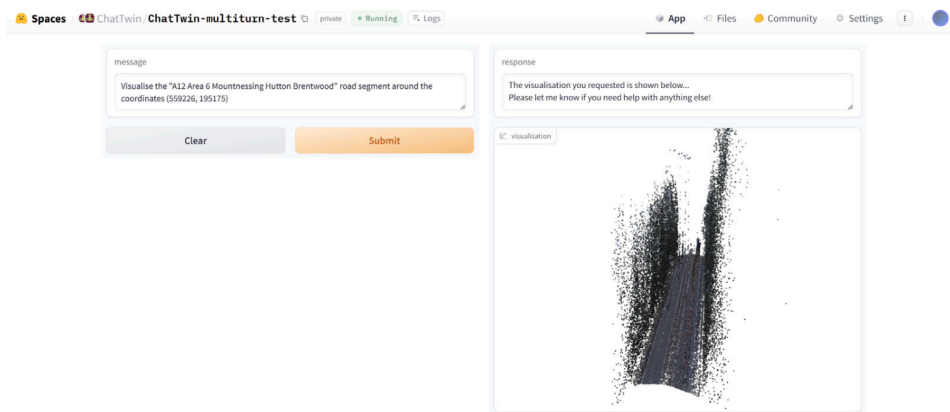


Figure 2.5: ChatTwin interface, showing the natural-language query input, the output area for retrieved results and responses, and an integrated visualization component. Taken from Luo et al. [43].

Research on multimodal and vision-language models demonstrates that grounding language in structured representations of 3D scenes enables a form of semantic selection that complements traditional interaction techniques [16, 32]. LLMs can map verbal references to specific sections of a 3D environment using methods that link geometric entities with object identifiers, embeddings, or scene graphs. In addition to highlighting significant advancements in tasks like object grounding,

spatial question answering, and open-vocabulary understanding, surveys of 3D vision-language systems also highlight ongoing issues with data scarcity, spatial ambiguity, and hallucination [67].

From an HCI standpoint, conversational interfaces should be seen as complementary mechanisms that lessen the cognitive and motor effort needed for specific types of tasks rather than as alternatives to manual interaction [26]. According to empirical investigations of LLM-assisted 3D systems, when conversational tools are available, users naturally use high-level, bulk-selection techniques, saving manual interaction for verification and fine-grained modification [15, 35]. However, issues of trust, transparency, and error recovery continue to be major difficulties, especially in engineering or safety-critical fields.

In industrial digital twins, conversational interaction is particularly attractive for semantic access in dense scenes, where manual selection can become time-consuming and error-prone. However, its practical effectiveness depends on the existence of stable identifiers, consistent metadata, and deterministic back-end selection mechanisms. Consequently, conversational interfaces are often most effective when embedded within hybrid interaction models that combine natural-language querying with established visualization and selection techniques [58]. This perspective underpins the design choices and evaluation strategy adopted in this dissertation.

Chapter 3

Related Work

To adequately address the dissertation’s research question, it is necessary to first examine the existing body of knowledge on interaction, selection, and filtering in dense 3D environments. This chapter reviews the state of the art in techniques that support the spatial and thematic selection of objects within complex CAD-derived models, with particular attention to challenges introduced by occlusion and visual clutter, and to approaches that connect geometry with semantic metadata. The chapter is organized into two parts: section 3.1 describes the exploratory review process used to identify, expand, and screen the relevant literature through iterative citation chaining; section 3.2 synthesizes the selected works into a thematic discussion of the main technique families and their implications for the interaction paradigm developed in this dissertation.

3.1 Exploratory Review Process

Interaction and selection in dense 3D environments spans multiple research threads, including classic 3D selection techniques under occlusion and clutter, occlusion-management and decomposition strategies, and more recent approaches that leverage semantic metadata and natural-language interfaces. Because these contributions are distributed across different communities and venues, the objective of this literature review is to consolidate the most relevant works and design implications into a coherent State of the Art that directly supports the dissertation’s research question.

To that end, this chapter adopts an exploratory review methodology. The review is organized as a structured multi-phase process that supports iterative discovery while maintaining transparency about how publications were identified, screened, and synthesized. The methodology is presented as a sequence of phases (identification, screening, and analysis) which are described in the following subsections.

3.1.1 Identification Phase

The review began with the identification of an initial seed set of publications selected for relevance and influence. This seed set included foundational surveys and canonical techniques related to 3D

object selection under occlusion, occlusion management strategies, decomposition methods, and more recent work on semantic enrichment and natural-language interaction with 3D environments.

Seed identification relied on targeted searches using domain-specific keywords (e.g., “3D selection occlusion dense scenes”, “progressive refinement selection”, “semantic metadata selection 3D”, “LLM interaction 3D scenes”) and by prioritizing widely cited contributions in the relevant technique families. The goal at this stage was not to exhaustively enumerate all papers returned by keyword search, but to obtain high-leverage starting points (surveys, taxonomies, and widely referenced technique papers) from which citation chaining could expand coverage reliably.

After establishing the seeds, the publication set was expanded through iterative citation chaining using ResearchRabbit. For each seed publication, two complementary exploration strategies were applied:

- Backward chaining: inspection of referenced works cited by the publication.
- Forward chaining: inspection of subsequent works that cite the publication.

During chaining, candidate publications were retained when titles and abstracts indicated a direct contribution to one or more of the following themes: selection techniques for dense/occluded 3D environments, occlusion management and focus+context methods, decomposition strategies, semantic enrichment and metadata-driven exploration, natural-language/LLM interaction with 3D environments, or hybrid interaction models combining manual and semantic workflows.

This iterative procedure was performed over three cycles, with periodic consolidation to avoid drift into domain-specific applications areas that did not provide transferable interaction or evaluation insight for the dissertation’s constraints. The output of the chaining stage was a candidate set of 75 publications selected for full-text screening.

The citation chaining process was terminated when additional iterations exhibited diminishing returns, meaning that newly discovered publications reintroduced the same canonical papers through multiple paths, and did not materially extend the conceptual taxonomy used in the State of the Art synthesis. In practice, saturation was considered reached once successive iterations contributed only marginal refinements rather than introducing new technique families or interaction paradigms relevant to the research question.

Publications were accessed through a combination of digital libraries and indexing platforms. Table 3.1 summarizes the primary sources from which the selected publications were retrieved. The counts refer to the primary access location, even when the paper was discovered via citation chaining. The “Others” category encompasses literature obtained from various independent publisher websites, open-access preprint servers, and institutional archives, such as arXiv¹, Wiley Online Library², and MDPI³.

¹<https://arxiv.org>

²<https://onlinelibrary.wiley.com>

³<https://www.mdpi.com>

Table 3.1: Primary access sources for the publications included in the reviewed set.

Source	Number of Publications
IEEE Xplore ⁴	30
ACM Digital Library ⁵	18
ScienceDirect (Elsevier) ⁶	7
SpringerLink ⁷	5
Others	15
Total	75

3.1.2 Screening Phase

Following identification and citation expansion, candidate publications were screened to construct a reviewed set aligned with the dissertation’s scope and constraints.

Screening was performed in two stages:

1. Title and abstract screening to rapidly exclude clearly irrelevant records.
2. Full-text screening for borderline cases and for confirming methodological relevance (i.e., the publication details the underlying technique, interaction design, or evaluative evidence).

Table 3.2 presents the inclusion and exclusion criteria applied during the screening phase. Across the reviewed candidates, 5 publications were excluded at full-text stage because they did not contribute transferable interaction mechanisms, evaluation evidence, or design implications for dense 3D selection and filtering under the dissertation’s constraints. While peer-reviewed journal and conference papers were prioritized, a small number of other high-quality literature sources (thesis, technical report, book) were retained when they provided foundational background or rapidly evolving evidence not yet consolidated in archival venues. The resulting reviewed set comprised 70 publications.

3.1.3 Review and Analysis

The final stage of the review consisted of synthesizing the reviewed publications into a structured analysis that directly supports the dissertation’s research question. Rather than treating the reviewed set as an undifferentiated list, the publications were grouped into recurring contribution themes that later structure Section 3.2 (e.g., selection techniques, occlusion management, decomposition methods, semantic enrichment, natural-language interfaces, and hybrid interaction models).

⁴<https://ieeexplore.ieee.org>

⁵<https://dl.acm.org>

⁶<https://www.sciencedirect.com>

⁷<https://link.springer.com>

Table 3.2: Inclusion and exclusion criteria for the screening phase.

Phase	Criteria
1	Record is an article, conference paper, book, thesis, or technical report.
2	Publication is written in English.
3	(<i>Exclusion</i>) Not a domain-specific application (e.g., medicine, biology) without detailing the underlying technology.
4	Addresses at least one core theme of this dissertation: selection in dense/occluded 3D scenes; occlusion management; decomposition methods (e.g., cutaways/exploded views); semantic enrichment/metadata-based selection; natural-language interaction for 3D environments; or hybrid interaction models.
5	Interaction and evaluation insights are applicable to desktop workflows (mouse/keyboard) or provide transferable principles that do not rely on specialized hardware.

This synthesis serves two complementary purposes. First, it establishes an organizing framework for Section 3.2 by clarifying how different technique families address selection and filtering under occlusion and clutter. Second, it helps ensure that the State of the Art reflects evidence across the relevant subfields, rather than implicitly privileging a single venue, methodology, or interaction paradigm.

To improve transparency regarding the composition of the reviewed set, the analysis also includes a compact descriptive summary of the included publications. Specifically, it reports the publication year distribution to contextualize the balance between foundational interaction work and more recent semantic and LLM-driven approaches, the thematic distribution across the State of the Art subsections and the distribution of publication types (Table 3.3). This summary is provided as a reference aid for the reader and supports traceability between the review process and the subsequent synthesis.

Table 3.3: Publication types within the reviewed set.

Type	Number of Publications
Conference Paper	36
Article	30
Book	2
Master’s Thesis	1
Technical Report	1
Total	70

Figure 3.1 presents the publication year distribution of the selected records, using three high-level themes consistent with the organization adopted in Section 3.2: direct manipulation and

visibility management, semantic metadata and model enrichment, and conversational and LLM-mediated interaction. Beyond providing a compact overview of coverage, this view helps distinguish foundational work that shaped interaction under occlusion from more recent contributions that emphasize semantic and conversational access pathways.

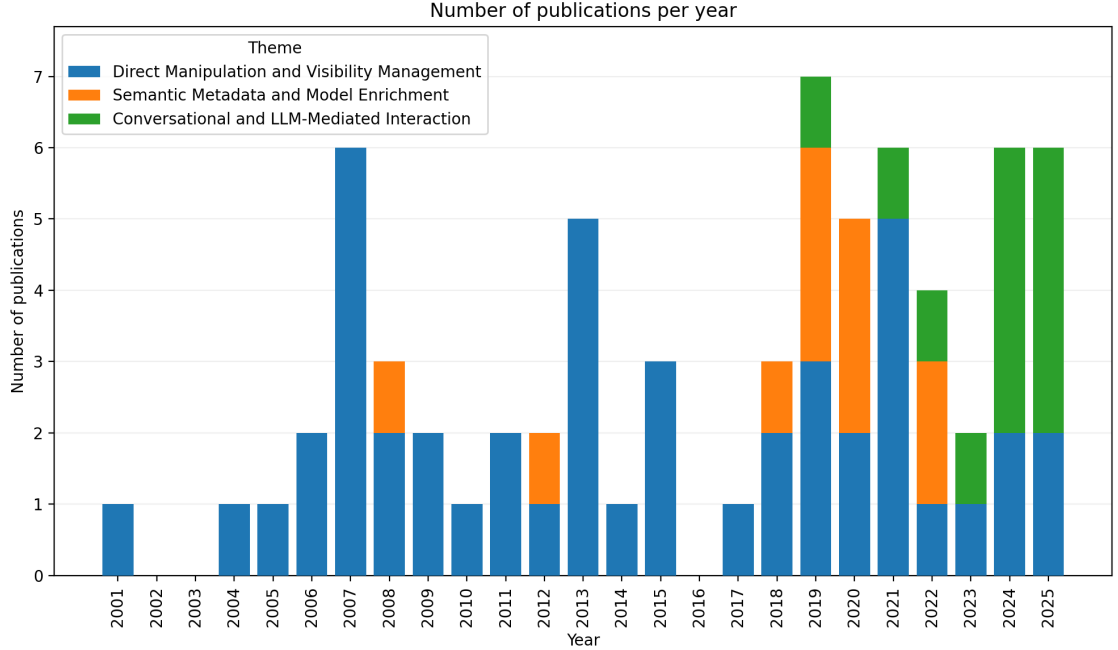


Figure 3.1: Number of publications per year in the reviewed set, grouped into three themes: direct manipulation and visibility management; semantic metadata and model enrichment; and conversational and LLM-mediated interaction.

3.2 State of the Art

In complex 3D environments, particularly dense, hierarchical industrial models, user performance is often constrained less by rendering fidelity than by the interaction bottlenecks of finding, disambiguating, and selecting the intended target among many candidates. Previous research consistently treats selection as a foundational primitive that supports higher-level operations, such as inspection, navigation, manipulation, and workflow execution. Consequently, weaknesses in selection propagate to the entire interaction pipeline [2]. At the same time, conventional pointing metaphors exhibit well-documented failure modes when targets are small, distant, or visually cluttered, and when confirmation actions introduce instability under dense target competition [66]. In parallel, occlusion and visual crowding motivate a large design space of occlusion-management strategies, ranging from multi-view layouts to X-ray and distortion-based approaches, whose purpose is to reveal otherwise inaccessible structure while preserving spatial context [24].

Accordingly, this section synthesizes the literature along four complementary lines that directly support interaction requirements in dense 3D model exploration:

- Interaction and selection techniques designed for dense/occluded settings, including progressive refinement and structure-aware selection
- Approaches leveraging semantic cues during manual selection to reduce ambiguity
- Recent language-mediated paradigms for 3D interaction and the associated grounding/feedback challenges
- Hybrid interaction models that combine manual precision with semantic and AI-assisted mechanisms

This organization also follows evaluation practice in the selection literature, where scene factors such as density, target visibility/occlusion, and task-environment design are treated as primary drivers of technique performance and workload.

3.2.1 Interaction Challenges in Dense Industrial 3D Models

Industrial-scale 3D models exhibit interaction constraints that differ qualitatively from the “general 3D UI” cases typically assumed by CAD viewers or generic virtual environment (VE) toolkits. In dense assemblies, the core difficulty is not simply reaching a target in 3D space, but rather identifying the intended object, maintaining awareness of alternatives, and confirming selection with high confidence in the presence of extensive occlusion, repeated parts, and tight spatial packing. Surveys of 3D selection techniques consistently highlight that traditional pointing metaphors (notably ray-casting) remain widely used due to their simplicity, yet they systematically degrade under cluttered conditions where multiple objects compete along a line of sight and small motor perturbations can lead to large depth errors [2, 17]. More generally, selection can be modeled as a set of coupled motor-control processes that unfold over time and are shaped by available feedback and target dynamics. Yu et al. [71] shows how such a multi-process framing can be operationalized as a design method by deriving and evaluating selection techniques that manipulate cursor feedback and target behavior to improve acquisition efficiency. In this setting, interaction failures are not minor inconveniences since they shape the feasibility of downstream tasks such as inspection, asset identification, and work planning.

Beyond target ambiguity, dense environments also amplify the cost of error. Under high occlusion, an incorrect selection is often difficult to detect immediately, and can lead to additional time-consuming actions such as refiltering, reorienting the camera, or performing corrective de-selection operations. Empirical work examining the relationship between environment density, target visibility, and selection performance demonstrates that selection time and error rates are strongly affected by these environmental factors, and that baseline techniques such as ray-casting can become unreliable in dense or heavily occluded scenes [66].

3.2.1.1 Density, Occlusion, and Target Ambiguity

A recurring conclusion across selection literature is that performance cannot be attributed solely to the user’s pointing accuracy. It emerges from an interaction between motor control, visual

feedback, and the structure of the scene. Argelaguet and Andujar [2] emphasizes that selection difficulty depends not only on target distance and size, but also on occlusion, visual clutter, and the feedback available to the user during the action. In dense assemblies, these factors combine to produce a form of “target competition”, in which multiple candidate objects fall within the effective selection area of the pointer, and small differences in cursor position or timing can redirect selection to a neighboring object. The resulting experience is frequently one of instability, where the user cannot reliably predict which object will be acquired at the moment of confirmation.

In desktop-based 3D interaction, these ambiguity effects are amplified by input asymmetry as the user’s control is typically limited to 2D pointer motion while the task requires 3D disambiguation. Teather and Stuerzlinger [63] shows that mouse-based 3D selection is sensitive not only to target size and distance, but also to how depth is visually encoded at the cursor. This depth-cue conflict is illustrated in Fig. 3.2, which shows how stereoscopic cursor rendering can create ambiguous depth anchoring at the point of commitment.

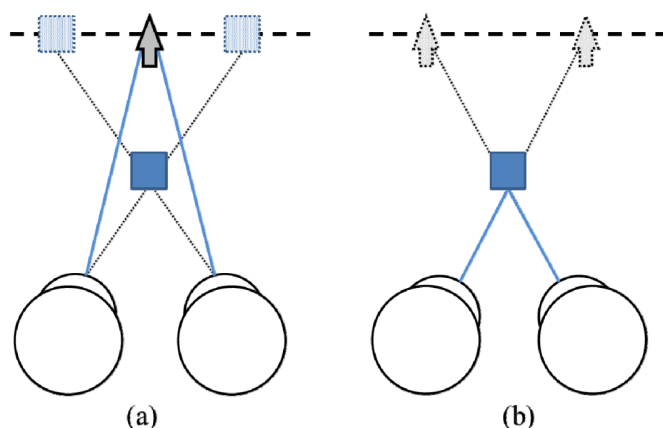


Figure 3.2: Depth ambiguity at the cursor in stereoscopic mouse-based 3D selection. Taken from Teather and Stuerzlinger [63].

In particular, conventional stereoscopic cursor renderings can introduce depth ambiguity (e.g., diplopia and mismatched depth cues), which in turn increases corrective motion and selection instability in cluttered configurations. Their results motivate a practical interpretation for dense industrial assemblies, namely that even before considering advanced selection metaphors, baseline “point-and-confirm” performance depends strongly on whether the interface provides stable and interpretable depth feedback at the moment of commitment, because the user must resolve both which object is targeted and where along depth the interaction is anchored. This reinforces the view that dense-model selection failures are not simply precision errors, but emerge from a coupling of motor control limits with inadequate or overloaded visual feedback under occlusion and crowding.

Vanacken, Grossman, and Coninx [65] shows that increasing environment density and reducing target visibility produce systematic changes in technique effectiveness, and that no single selection method is uniformly superior across all conditions. Figure 3.3 shows completion times

by cursor technique under varying density spacing (left) and occlusion (right). Larger spacing generally reduces time and occlusion increases time for all techniques.

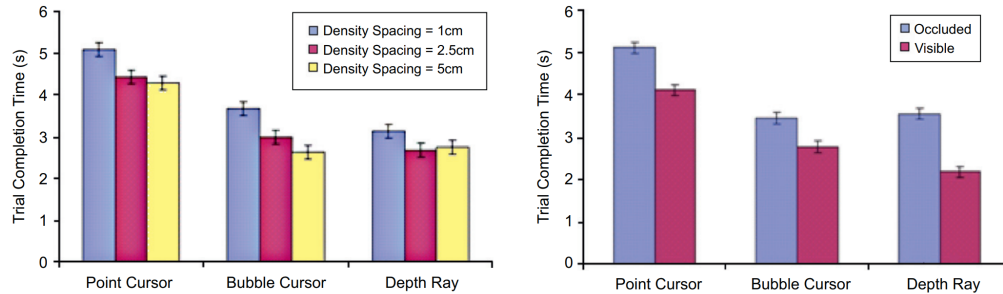


Figure 3.3: Completion time by cursor technique as a function of density spacing (left) and visibility/occlusion condition (right). Taken from Vanacken, Grossman, and Coninx [66].

In particular, techniques designed to help with general selection can fail when the occluded target case dominates, because the user’s interaction is forced to operate through incomplete visual information. Follow-up work expands this perspective by demonstrating that the addition of non-visual feedback channels can mitigate some of the ambiguity inherent in dense, occluded selection tasks. This is an observation that highlights that selection reliability depends on feedback design, not only on geometric pointing mechanisms [66]. While the feasibility of haptic feedback varies by platform, the general implication remains transferable since dense scenes require redundant and interpretable cues to support confident target acquisition.

Occlusion further complicates the problem because it is not simply a visibility issue, but rather changes the nature of the user’s task. Elmqvist and Tsigas [23] formalizes occlusion management by distinguishing between tasks such as discovery, access, and spatial relation understanding, and by characterizing techniques through dimensions such as purpose, strength, and depth cues. For dense industrial models, this taxonomy is particularly useful because selection often requires first identifying candidate objects and then making them selectable, while preserving sufficient spatial context to avoid disorientation. In other words, the user’s difficulty is not resolved by revealing the target alone. The system must reveal it in a way that maintains interpretability, allowing the user to confirm that the revealed object is indeed the intended one.

A practical difficulty in consolidating findings across this literature is that selection performance is highly contingent on experimental design choices that are often under-specified or inconsistent between studies. Bergström et al. [8] synthesizes two decades of evaluation practice and argues that comparability requires clear reporting of task structure, target properties, environment configuration, input or hardware constraints, and dependent measures (e.g., time, error, throughput), as well as careful alignment between what is measured and what the technique claims to improve. Although their guidelines are articulated in a VR-centric context, the underlying implication transfers directly to dense industrial digital twins. Density, occlusion, target multiplicity, and feedback conditions are not secondary variables, but defining parameters that must be made

explicit when interpreting reported performance differences. To operationalize this comparability requirement, Table 3.4 consolidates commonly used task-environment factors from established evaluation taxonomies and subsequent updates, providing a grounding checklist for interpreting results across otherwise heterogeneous studies.

From a methodological perspective, the importance of density and occlusion is also reflected in how selection studies define their experimental conditions. Bashar and Batmaz [7] provides a recent synthesis of how virtual task environment factors are operationalized in 3D selection literature, showing that classical parameters (distance, size, density, occlusion/visibility) remain central, while newer dimensions, such as adaptive selection and application-specific task environments, are increasingly recognized but inconsistently standardized. This is directly relevant for industrial Digital Twins. When the target environment is an LHC-scale model rather than a controlled laboratory scene, the interaction problem should be understood as a multi-factor regime (density $\times$ occlusion $\times$ target multiplicity $\times$ hierarchical structure), not as a single-variable optimization.

3.2.1.2 Limits of Navigation-Driven Interaction

In practice, users often compensate for selection difficulty by relying on navigation, moving closer, changing viewpoint, or repeatedly zooming until the target becomes visually isolated. While this strategy can work for sparse scenes, it becomes increasingly inefficient as density and occlusion rise, because every viewpoint change can introduce new occlusions and new candidates. Arge-laguet and Andujar [2] notes that selection techniques are frequently evaluated in isolation, but real tasks involve coupled processes that collectively shape performance and fatigue.

In dense industrial models, the navigation “burden” is not merely time overhead, it creates a cognitive workload associated with maintaining spatial context, re-identifying previously seen components, and remembering semantic intent across viewpoint changes. This dynamic is visible in interaction work focused on hidden-object manipulation. Even when the target is known, occluding geometry can force users into repeated camera adjustments or multi-step operations to maintain target visibility during interaction, as can be seen in Fig. 3.4.

Techniques that attempt to keep objects visible during manipulation (e.g., rendering masks or transparency overlays) implicitly acknowledge that camera motion alone is an insufficient solution when occlusion is structural rather than incidental [61]. The implication for dense CAD-like environments is that a system that relies primarily on viewpoint changes for disambiguation effectively shifts the problem onto the user, increasing interaction cost and reducing predictability.

A further limitation of navigation-driven workflows is that the user’s ability to resolve ambiguity does not scale linearly with model complexity. Dense industrial scenes frequently contain repeated or near-identical components, and occluded objects may be visually indistinguishable until the user performs deliberate disocclusion operations. In such cases, interaction requires mechanisms that reduce the candidate set without requiring exhaustive viewpoint exploration. This is precisely why much of the selection literature has moved toward strategies that either provide richer disambiguation mechanisms (e.g., depth cues, cursor expansions, candidate sets) or modify

Table 3.4: Virtual task-environment factors commonly used to evaluate 3D selection techniques, focusing on the LaViola et al. [38] core parameters and subsequent updates consolidated by Bashar and Batmaz [7].

A. Core task parameters (LaViola classification)

Target distance & direction	Authors vary distance and direction to the targets to understand what effects this has on the user's perception, accuracy, and time.
Target size	Researchers investigate the effect on 3D selection performance and task completion time changes in the target's width, height, and size.
The density of the objects around the target	Authors evaluated the effects of object density on 3D selection performance by designing virtual environments with a high concentration of objects in proximity to the target.
Number of targets to be selected	The impact of object size, dimension, and quantity on user selection performance was investigated by researchers in scenarios where users were tasked with choosing a specific number of unique objects from a set of similar objects.
Target occlusion	Researchers designed the environment in an occluded manner and varied the degree of occlusion, such as the size and location of the occluding object, to measure how these factors affect task completion time, error rate, and user selection preferences.

B. Extensions (task-environment categories)

Depth	In virtual environments, target depth refers to the spatial distance extending from the user to a designated point of interest, impacting the perception of objects in terms of size, location, and movement within the virtual space.
Targets inside a volume	Numerous targets were placed within a volume to enhance object density and occlusion. These studies were conducted by researchers to examine how object density, occlusion, and target collisions within a confined space affect user performance and perception in virtual environments.
Application specific environments	VR systems were incorporated by researchers to create 3D selection VEs, allowing users to explore virtual space, view objects from various angles, and modify object attributes like position, size, orientation, and environmental conditions to facilitate effective selection.

C. New parameters (recent work)

Adaptive selection	Adaptive target selection in virtual environments involves the dynamic process of identifying and selecting targets that exhibit changes in attributes such as movement, size, shape, appearance, or disappearance. These papers focused on users' quick and accurate identification and selection of targets, considering factors like target characteristics (size, shape, color, movement patterns), which significantly impact the selection process.
Accessibility of objects	Accessibility of objects/targets in virtual environments is a complex factor that includes the target's location, size, shape, color, layout, and design of the virtual environment, as well as the user's expertise level and preference. Researchers investigated user perception and performance in selecting specific areas within large, dense, 3D-immersed virtual environments.

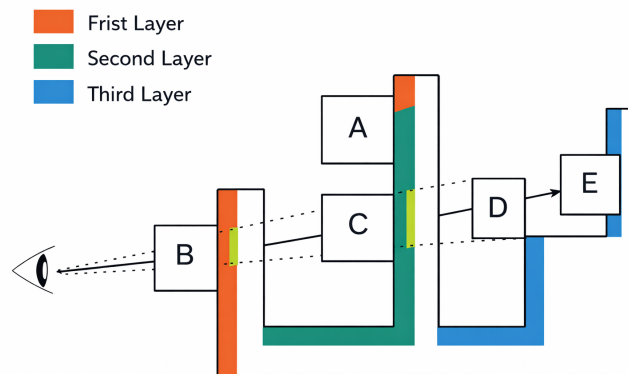


Figure 3.4: Structural occlusion during interaction: several intended targets remain fully hidden despite navigation, motivating explicit disocclusion support. Taken from Sun and Stuerzlinger [61].

the display to manage occlusion while preserving context, approaches that are difficult to achieve through navigation alone [24, 65].

A related scalability issue is that many industrial tasks are not single-target selections, but set-definition tasks (e.g., isolating a subsystem, selecting all instances of a component type, or forming a “working set” for inspection). In this context, interaction bottlenecks arise not only from picking the correct object once, but from repeatedly acquiring many objects under the same occlusion and clutter conditions. Lucas et al. [42] distinguishes serial acquisition from parallel or region-based approaches and emphasize that multi-object selection becomes a first-order concern as scene complexity increases. For dense assemblies, this observation is important because it re-frames part of the interaction problem as a candidate-set construction task, in which the interface should provide mechanisms to select, filter, or group objects in ways that are robust to occlusion and repeated structures, rather than forcing the user through repeated point selections and corrective navigation. This perspective complements the selection literature focused on single-target accuracy by highlighting why dense-model workflows often demand higher-level operations that act on collections (e.g., subassemblies, spatial regions, semantic categories) rather than isolated picks.

Taken together, these limitations indicate that selection in dense environments is best treated as a system-level interaction problem rather than a purely geometric pointing problem. This framing guides the remainder of this State of the Art, which considers ambiguity-aware selection techniques under density, occlusion-management mechanisms that support discovery and access, and semantic and language-based approaches that reduce navigation burden by enabling higher-level expressions of intent [7]. To support cross-study interpretation of these approaches, Table 3.4 summarizes the virtual task-environment parameters most commonly varied in 3D selection evaluations. The gaps that can be identified are that these factors are still not reported and controlled consistently across studies, which limits cross-paper comparability in dense and occluded condi-

tions, and that evaluation setups rarely operationalize selection as an end-to-end identification step that includes user-verifiable confirmation and recovery from ambiguity at the moment of commitment.

3.2.2 Manual Selection Techniques

Manual selection remains the dominant interaction modality in desktop-based 3D systems due to its directness, predictability, and compatibility with conventional input devices such as mouse and keyboard. However, as established in the previous subsection, dense and occluded environments exacerbate the limitations of naïve pointing metaphors. Consequently, a substantial body of research has focused on extending or restructuring manual selection techniques to improve accuracy, reduce ambiguity, and maintain user control under challenging spatial conditions. This subsection reviews the principal families of manual selection techniques relevant to dense 3D models, with an emphasis on approaches that are compatible, conceptually and practically, with non-immersive, desktop interaction.

3.2.2.1 Ray-Based Selection and Incremental Refinements

Ray-casting has long served as the baseline technique for 3D object selection, owing to its conceptual simplicity and strong analogy with 2D pointing. In its canonical form, a ray is projected from the camera or cursor into the scene, and the first intersected object is selected. Despite its widespread adoption, extensive empirical evidence shows that ray-casting suffers from reduced precision in dense environments, particularly when multiple objects lie along the ray direction or when targets are small and distant [2]. Lens-based extensions of ray-casting have also been proposed to improve accuracy in dense 3D environments while preserving the familiar ray metaphor [54]. These issues are amplified in desktop contexts, where depth perception is limited to monocular cues and subtle cursor movements can cause large depth variations at distance.

Several works refine the ray-based selection paradigm by augmenting it with additional depth cues rather than replacing it entirely. The Depth Ray incorporates an explicit depth marker along the ray to indicate the current selection depth, allowing users to anticipate which object will be selected even under partial occlusion [65]. Follow-up studies demonstrate that while such depth-aware variants improve performance in high-occlusion scenarios, they remain sensitive to motor noise and do not fully resolve competition among closely packed targets [66].

Other refinements focus on improving control over ray sensitivity. Ro, Lee, and Woo [55] proposes a dynamic depth-variable ray-casting interface in which the control-display ratio is adjusted based on target distance, allowing finer control when interacting with distant objects. While effective in reducing selection time for far targets, such approaches still assume that the correct target can be visually identified along the ray, an assumption that often breaks down in dense assemblies.

A complementary strategy is to decouple pointing direction from final target selection by introducing an explicit cursor along the ray. RayCursor allows users to slide a virtual cursor forward and backward along the ray, effectively transforming depth selection into a one-dimensional

adjustment task [5]. This design reduces inadvertent target switching caused by small angular deviations and has been shown to outperform both standard ray-casting and Depth Ray for small or distant targets. However, it introduces an additional control dimension that must be mapped to available input devices, which can complicate interaction in desktop setups without dedicated secondary controllers. Figure 3.5 contrasts standard ray-casting with two depth-oriented refinements using the same scene.

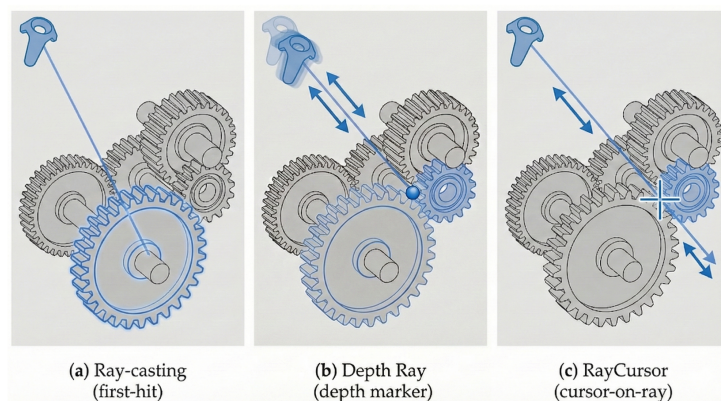


Figure 3.5: Ray-based selection and depth-oriented refinements in a common scene: (a) ray-casting selects the first intersected object, (b) Depth Ray introduces an explicit depth marker along the ray to target deeper objects, and (c) RayCursor places a movable cursor on the ray to control the selected depth. Based on Argelaguet and Andujar [2], Baloup, Pietrzak, and Casiez [5], and Vanacken, Grossman, and Coninx [65].

A complementary line of work addresses dense occlusion by re-representing the candidate space rather than adding depth control to the primary ray. Related focus+context drill-in strategies use lens metaphors to access occluded structure without fully abandoning context. For instance, Pindat et al. [53] introduces Gimlenses, a multi-view detail-in-context technique for navigating and selecting within complex models (Fig. 3.6).

As an alternative auxiliary-view strategy, Lee and Park [39] proposes 3D Mirrored Object Selection, which enables selecting occluded objects through a virtual mirror view while preserving the surrounding context (Fig. 3.7).

Similarly, the Interactive Disocclusion Mini-map family of techniques introduces disocclusion widgets that externalize candidate objects into a compact auxiliary representation, reducing dependence on repeated viewpoint changes for target acquisition [47]. Although these methods are demonstrated in VR settings, they remain relevant to desktop industrial viewers as design patterns because they decouple an object’s spatial location from the mechanism used to select it, a separation that can be implemented through a transient auxiliary panel or focus view without permanently altering the global camera.

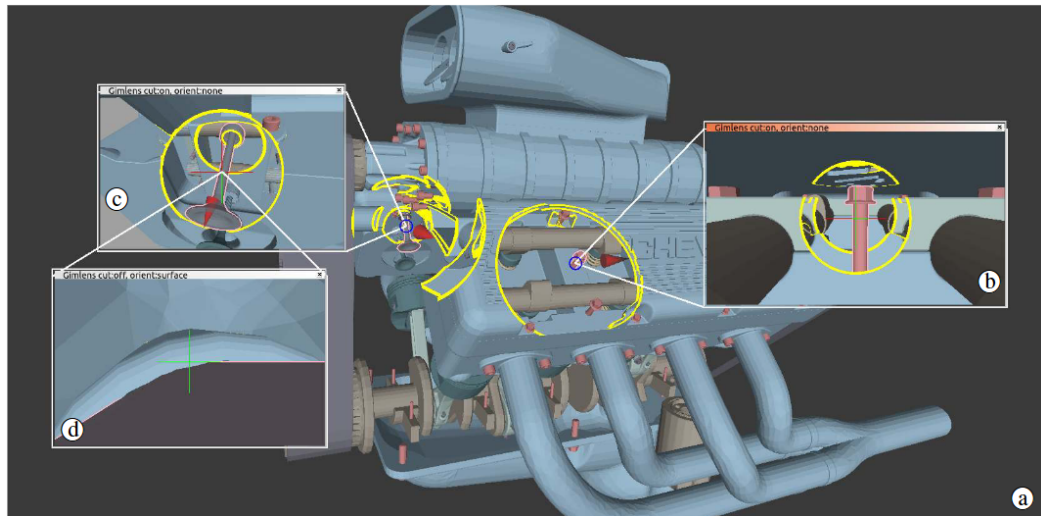


Figure 3.6: Gimlenses as a multi-view detail-in-context technique for exploring complex 3D CAD models: (a) context view of a car engine; (b-d) three lens views providing magnified drill-in perspectives at different orientations and magnification levels to reveal components buried within the model while preserving their location in the global scene. Taken from Pindat et al. [53].

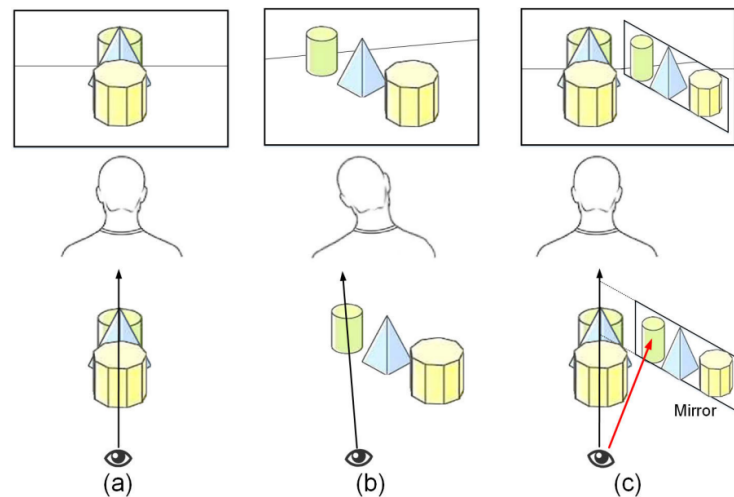


Figure 3.7: Virtual mirror selection under occlusion: (a) from the front view, the target object is occluded by other objects, hindering accurate selection; (b) the target becomes selectable after a viewpoint change; (c) the mirror provides an alternative view of the occluded target without requiring user movement, supporting discrimination and selection through the reflected scene. Taken from Lee and Park [39].

3.2.2.2 Progressive Refinement and Multi-Step Disambiguation

While ray-based refinements aim to preserve immediacy, another class of techniques explicitly embraces multi-step interaction to trade speed for accuracy. Progressive refinement techniques separate selection into a coarse identification phase followed by a fine disambiguation phase, thereby reducing the need for precise initial pointing in cluttered scenes. Kopper, Bacim, and Bowman [37] introduces SQUAD (Sphere-Casting Refined by QUAD-Menu), which first gathers candidate objects within a coarse spherical volume and then presents them in a structured menu for final selection (Fig. 3.8).

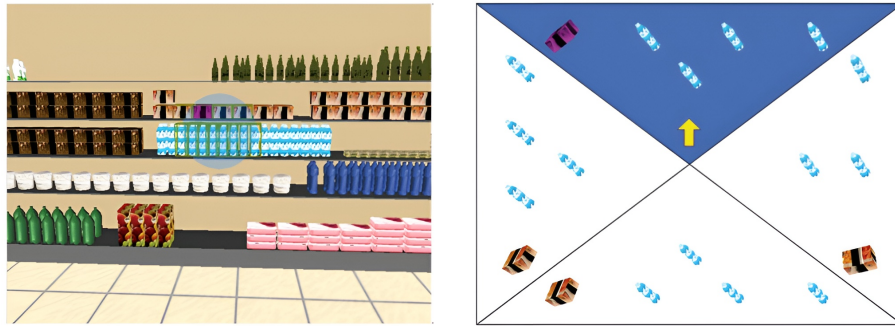


Figure 3.8: Interaction flow of SQUAD (Sphere-Casting Refined by QUAD-Menu): coarse sphere-casting gathers candidate objects, followed by menu-based disambiguation and final selection. Taken from Kopper, Bacim, and Bowman [37].

This approach effectively mitigates errors arising from cursor instability and yields very low error rates for small or occluded targets.

Subsequent work expands this paradigm by exploring alternative refinement mechanisms, such as discrete and continuous zoom-based disambiguation [4], as illustrated in Fig. 3.9. These techniques allow users to temporarily magnify a subset of the scene or candidate targets, improving precision without permanently altering the global view. Experimental results consistently show that progressive refinement significantly improves accuracy in dense environments, while increasing task completion time relative to immediate-selection techniques.

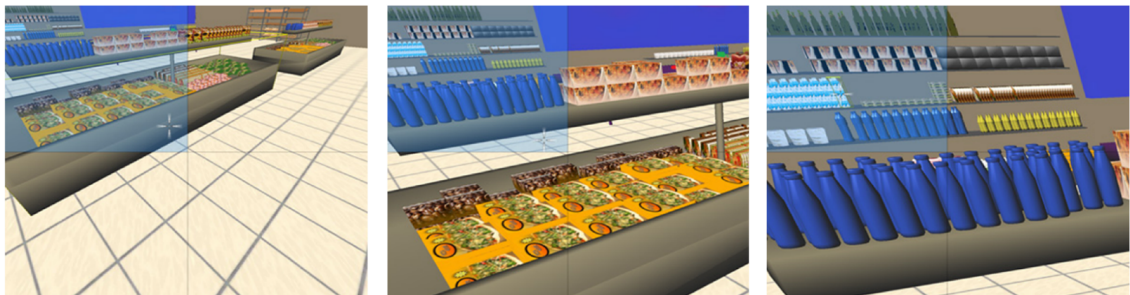


Figure 3.9: Progressive refinement via discrete zoom: the user repeatedly magnifies a selected screen region to increase the apparent size of candidate targets, enabling disambiguation with reduced pointing precision. Taken from Bacim, Kopper, and Bowman [4].

From an industrial interaction perspective, the relevance of progressive refinement lies in its robustness. Dense CAD assemblies often prioritize correctness over speed, especially when selections trigger costly downstream actions. However, the requirement for visually distinct candidates or structured menus introduces design challenges when dealing with repeated or visually similar components, a common characteristic of large-scale engineering models. This limitation is also emphasized in work that evaluates occluded selection under editor-like tasks where many distractors are visually similar. In particular, Maslych et al. [47] explicitly notes that a realistic stress case is selecting among objects that look the same, and discusses adapting the technique by tuning the size of the highlighting cone to manage candidate overload. The broader implication for dense CAD assemblies is that refinement must often be paired with candidate-management mechanisms because geometry-only cues may be insufficient to disambiguate repeated instances.

3.2.2.3 Structure and Context-Aware Manual Selection

A different line of work seeks to reduce ambiguity not by refining pointing accuracy, but by leveraging scene structure and semantic organization. In scientific and engineering datasets, objects are rarely isolated primitives, they belong to clusters, hierarchies, or functional groupings.

This structure-aware direction is also documented more broadly in visualization interface surveys, which position lasso-based and view-dependent methods as systematic attempts to infer 3D intent from 2D input under depth ambiguity. For instance, Besançon et al. [9] reviews view-ray-based picking methods (e.g., WYSIWYP-style approaches that select salient structures along the ray) and lasso-driven techniques that recover volumetric or clustered selections from screen-space gestures, explicitly citing CloudLasso as an example of extending a 2D lasso into 3D while using density structure to match user intent (Fig. 3.10). This framing is useful in industrial Digital Twins because it motivates structure-aware selection not as a niche alternative, but as a principled response to the mismatch between 2DOF input and complex 3D spatial targets. For example, in a dense assembly, a user's 2D lasso drawn over a visually coherent substructure can be interpreted as selecting that substructure's corresponding 3D cluster, rather than only the screen-space projection of whichever elements happen to be front-most.

Yu et al. [72] proposes structure-aware selection techniques for dense point cloud visualizations, such as TeddySelection and CloudLasso, which infer selection volumes based on local density and spatial coherence rather than strict geometric primitives. Figure 3.10 illustrates the CloudLasso workflow, where a screen-space lasso is transformed into a coherent 3D selection volume by estimating local density and extracting the above-threshold region, enabling cluster-level selection with minimal manual input.

By allowing users to select entire clusters through simple 2D gestures, these techniques significantly reduce the number of interaction steps required in dense scenes.

Although originally developed for point cloud data, the underlying principle of selecting meaningful groups rather than individual geometric elements readily extends to complex CAD and Digital Twin contexts, in which assemblies are composed of semantically related parts. A related concept is Semantic Pointing, which biases selection toward objects that are more relevant

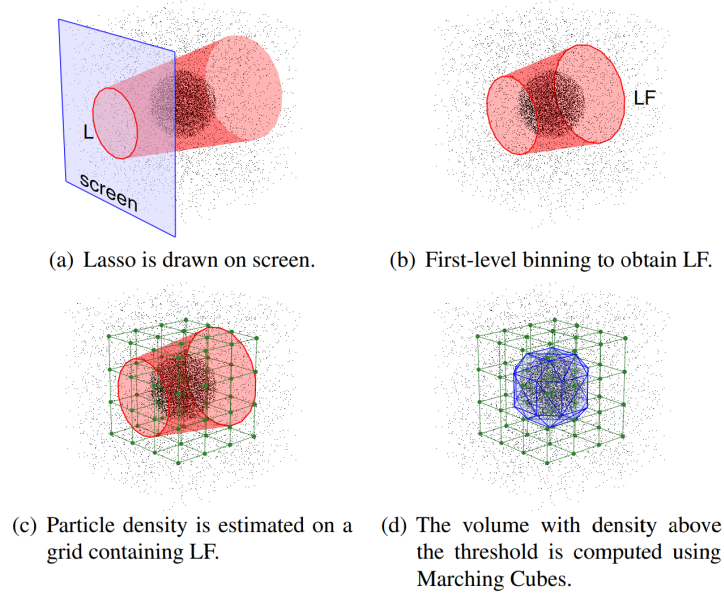


Figure 3.10: CloudLasso structure-aware selection pipeline: a 2D lasso is drawn in screen space, local density is estimated in the corresponding region, and a coherent 3D selection volume is extracted. Taken from Yu et al. [72].

given the user’s task context or interaction history, effectively reshaping the selection space to favor likely targets. Elmqvist and Fekete [22] illustrates this mechanism by overlaying the induced semantic influence field on the view, showing how proximity to selectable objects produces a target-weighted selection space (Fig. 3.11).

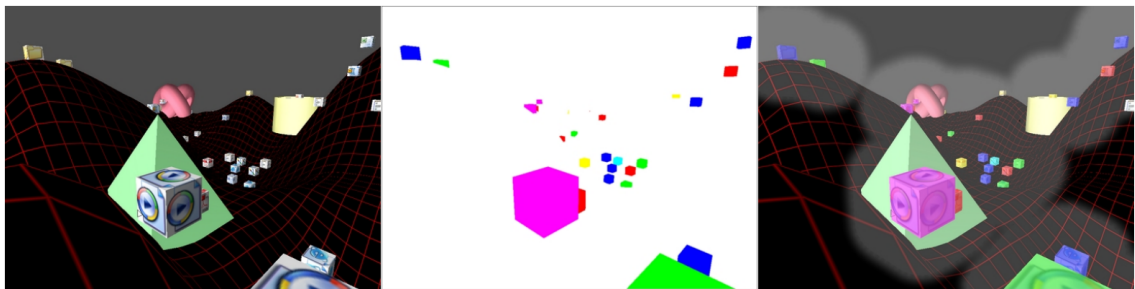


Figure 3.11: Semantic space for 3D object picking: the rendered view (left), the corresponding semantic/object buffer (middle), and the semantic influence field overlaid on the image space (right). The visualization illustrates how target proximity induces a target-weighted selection space that compresses empty regions and expands areas around selectable objects. Taken from Elmqvist and Fekete [22].

These approaches highlight an important shift, in which manual selection need not operate purely in geometric screen space. By embedding semantic or structural knowledge into the selection mechanism, systems can reduce ambiguity without introducing additional interaction steps. However, such techniques rely on the availability and quality of underlying structure or metadata, a constraint that is not always satisfied in heterogeneous industrial datasets.

3.2.2.4 Exploded Views and Scene Decomposition as Selection Enablers

While the previous families (ray-based, progressive refinement, and structure-aware selection) primarily aim to improve pointing precision under clutter, another complementary strategy is to reduce the ambiguity of selection by changing the scene configuration. In heavily occluded assemblies, a user often cannot reliably acquire a target because the correct object is not visually accessible in the first place. Therefore, interaction techniques that restructure visibility can be regarded as pre-selection enablers that transform an access problem into a simpler selection task.

A representative approach is the use of interactive exploded views, where parts are displaced to reveal internal structure while preserving global context. Li et al. [40] proposes a system that instruments a complex 3D model into an explosion graph encoding the allowable relative displacements between parts, and uses this structure to support interactive exploration without violating blocking constraints (Fig. 3.12).

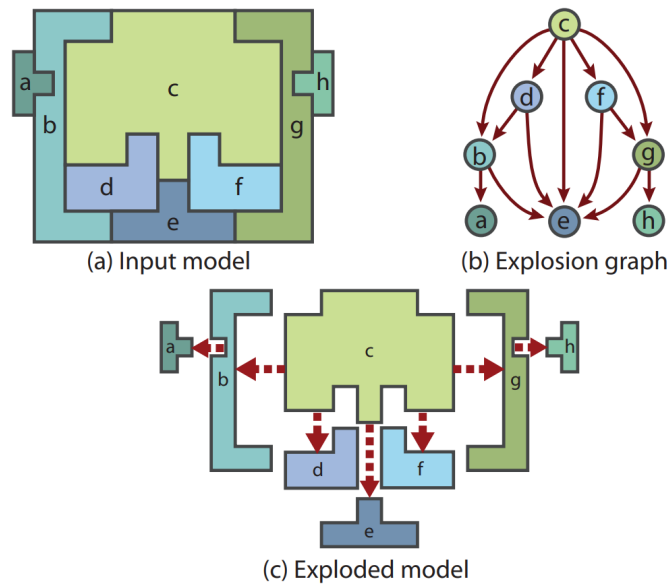


Figure 3.12: Explosion graph representation for interactive exploded views: (a) input model, (b) explosion graph, and (c) an exploded configuration generated under blocking constraints (i.e., parts intersecting or passing through one another). Taken from Li et al. [40].

The method explicitly accounts for part hierarchies and includes strategies for common interlocking conditions, enabling users to explore assemblies at different semantic levels. Beyond the representation, the interaction design emphasizes both direct controls (e.g., expand and collapse along explosion directions) and higher-level modes intended to help users locate specific parts of interest. Importantly, the system can also generate an exploded configuration that exposes user-selected targets without exploding the entire model, mitigating clutter that often arises in fully exploded illustrations.

From an implementation standpoint, the paper is also useful because it makes the underlying constraints concrete. For example, the work details how to split interlocked sub-assemblies and

container parts so that they become visible, and it discusses view-dependent precomputation to avoid interaction lag. These design details matter if exploded views are treated as an interactive, user-driven occlusion-management tool rather than a static illustration.

Finally, the reported precomputation costs and limitations are directly relevant when considering exploded views in production-scale digital twins. Preprocessing is dominated by contact computation, and the approach assumes interference-free inputs and restricts part separation to rigid linear translations, which is problematic for threaded or screw-like removals. These practical constraints are critical when deciding whether exploded views should be deployed globally or selectively, for example within a local neighborhood or subassembly.

Complementary lines of work explore exploded views from different angles than explosion-graph instrumentation. For instance, projection-based automatic generation methods (Fig. 3.13) target viewpoint-specific clarity by selecting explosion directions and distances to reduce projected interference under a specified view angle [57].

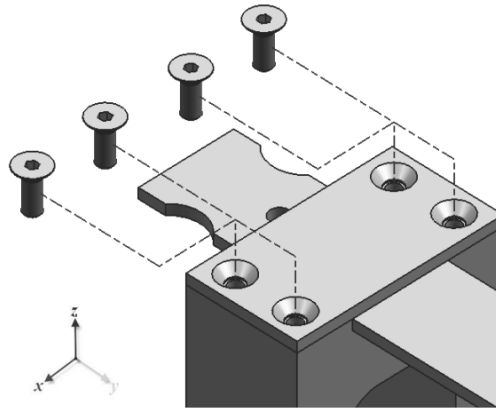


Figure 3.13: Projection-based automatic exploded view for a mechanical assembly: parts are separated along selected directions and distances to reduce projected interference under a given viewpoint, improving target discernibility without requiring extensive viewpoint changes. Taken from Shao et al. [57].

In visualization contexts, interactive exploded view diagrams for generalized 3D grids (Fig. 3.14) have been proposed to expose occluded interior structures while retaining global context [25].

More recently, BANG reframes exploded views as a generative process, producing coherent sequences of “exploded states” that progressively separate parts while maintaining consistency over time [73].

3.2.2.5 Summary

Taken together, manual selection techniques for dense 3D models span a spectrum from incremental refinements of ray-casting to multi-step progressive refinement and structure-aware approaches. Each family addresses a different facet of the density and occlusion problem: ray extensions improve depth control, refinement techniques reduce sensitivity to motor error, and

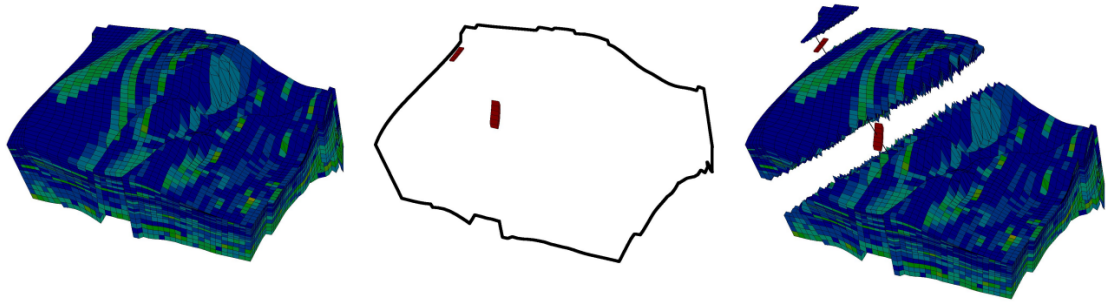


Figure 3.14: Exploded view diagrams for a 3D grid: the representation is separated into layers to expose occluded interior structures while preserving a coherent global layout and contextual cues. Taken from Filho, Brazil, and Sousa [25].

semantic or structure-aware methods shrink the effective candidate space. Importantly, no single technique fully resolves the interaction challenges posed by large, dense industrial assemblies. The gaps that can be identified are that many manual techniques still assume that candidates can be reliably discriminated using geometric or view-derived cues, which becomes fragile under repetition and strong visual similarity, and that several high-accuracy approaches introduce additional control dimensions, precomputation overheads, or part-separability assumptions that complicate direct adoption in production-scale desktop Digital Twin viewers. This observation reinforces findings from methodological surveys that emphasize the context-dependent nature of selection performance and the absence of universally optimal solutions [7].

For desktop-based Digital Twins, these results suggest that effective interaction design must balance immediacy, precision, and cognitive load, potentially combining multiple techniques rather than committing to a single paradigm.

3.2.3 Semantic Enrichment and Metadata-Driven Interaction

While the techniques reviewed in 3.2.2 improve the mechanics of manual pointing, they largely treat the 3D scene as a geometric field in which the user must resolve ambiguity through motor precision, view changes, or multi-step refinement. In dense industrial assemblies, this assumption is often misaligned with engineering practice, where intervention planning and diagnosis typically begin from semantic intent (e.g., “this family of magnets,” “equipment in this sector,” or “assets with an upcoming maintenance window”) and only later require precise spatial identification of a particular instance. This mismatch motivates approaches that enrich 3D geometry with semantic structure so that interaction can be expressed at a higher level of abstraction than purely geometric selection [22, 60]. In this view, metadata is not merely an annotation layer, it becomes an interaction substrate that can reduce ambiguity, constrain candidate sets, and enable task-driven exploration.

3.2.3.1 From Geometry to Semantics: What Semantic Enrichment Enables

A consistent theme in Digital Twin and built-environment literature is that 3D geometry alone is insufficient to support operational tasks because it does not encode function, identity, or history. Semantic enrichment pipelines therefore aim to attach higher-level information, often derived from ontologies, classification, or external databases, to geometric representations so that the model can answer queries aligned with user goals [60]. Although much of this work is motivated by facility management and indoor environments, the underlying interaction implication is more general. Once objects are associated with identifiers and semantic labels, users can move beyond point-based selection toward meaning-based selection, such as selecting all instances of a class, assets matching a property, or elements participating in a relation. This shift is particularly important in dense environments, where repeated components make purely visual disambiguation fragile.

Importantly, semantic enrichment does not eliminate the need for manual selection because it shifts when and why manual selection is needed. Semantics can be used to narrow the candidate set before any pointing occurs, bias pointing toward likely targets, or validate and contextualize a manual selection after the fact. Semantic pointing represents a concrete example of this integration, as it modulates selection behavior based on semantic cues and thereby reshapes the selection space to favor meaningful targets without removing user control [22]. Figure 3.15 illustrates this idea by showing that, rather than changing visibility, semantic relevance acts as a weighting over candidates near the cursor, increasing the likelihood that the selection resolves to a meaningful target. The broader implication is that interaction can be improved by embedding semantic priors into low-level techniques, rather than only improving geometry-based precision.

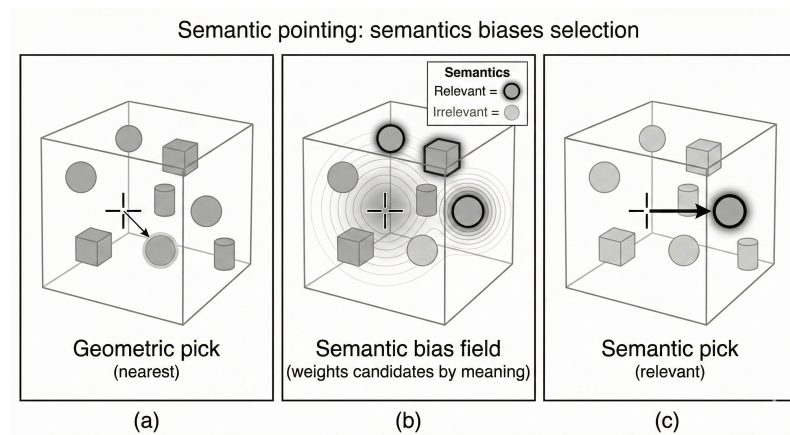


Figure 3.15: Conceptual diagram of semantic pointing: compared to purely geometric picking, semantic relevance biases the effective selection space so the chosen target is more likely to match user intent. Adapted from Elmqvist and Fekete [22].

3.2.3.2 Semantic Filtering Versus Semantic Selection

A key distinction for dense industrial models is between semantic filtering and semantic selection. Semantic filtering reduces visual complexity by controlling visibility, styling, or emphasis

according to categories and task contexts. Semantic selection, in contrast, produces an explicit set of objects that becomes the target of downstream actions such as inspection, metadata retrieval, or automated reasoning. Many systems in the BIM space emphasize semantic filtering through task-specific visualization configurations, illustrated in Fig. 3.16, that combine techniques such as X-ray, clipping, color coding, and focus+context rendering [27]. Such approaches are valuable because they acknowledge that dense environments require active complexity management. However, they often treat interaction as the toggling or composition of visualization modes rather than as an object-level acquisition problem.

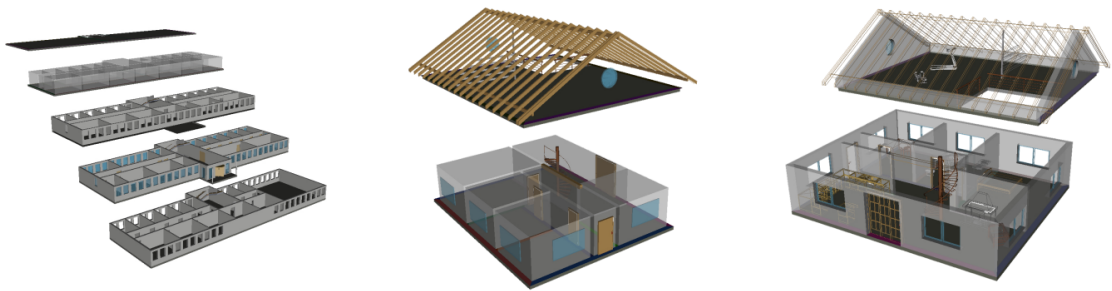


Figure 3.16: Task-specific BIM visualization configurations, achieved by combining semantic-driven exploded, cut-away, and ghosted views to support the inspection and filtering of building components. Taken from Florio, Trapp, and Döllner [27].

A pragmatic perspective on semantic filtering and metadata-driven exploration is provided by CAD3A, a web-based system designed explicitly for CAD assemblies [44]. The authors motivate the need to inspect internal parts through interaction mechanisms such as adjusting transparency of occluding components and defining section views via one or more planes, reflecting common industry practices for coping with occlusion during inspection. They further highlight that “important informative data” may be stored as annotations or implicitly deduced from model analysis, while additional semantics may reside in external enterprise systems (PLM), reinforcing the broader “geometry-metadata split” that also appears in large industrial contexts such as CERN’s.

From an interaction-design perspective, CAD3A provides a concrete example of how semantic information becomes actionable within a conventional mouse and keyboard workflow (Fig. 3.17). The interface offers explicit controls for toggling opaque and transparent views, showing or hiding parts globally or hierarchically, recoloring components, and resetting the view.

Critically for selection validation, selecting a part in the 3D view highlights the corresponding entry in the hierarchy tree, strengthening the coupling between spatial picking and structural identification. The same interaction style extends to derived semantic results (interfaces/contacts, functional classes, patterns), where selecting a computed record filters the 3D rendering to the relevant subset, illustrating a practical pathway from “semantic enrichment” to “semantic selection”.

From the standpoint of maintenance-oriented Digital Twins, semantic filtering is necessary but rarely sufficient. The gaps that can be identified are that semantic filtering is often treated as

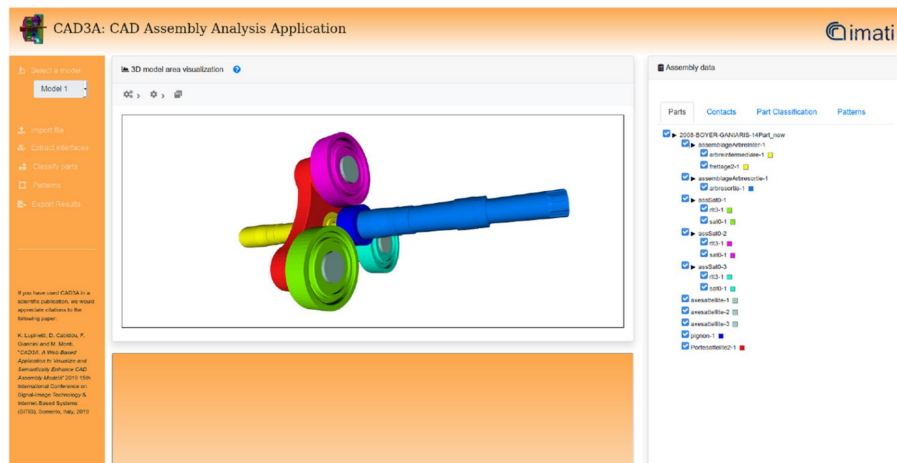


Figure 3.17: CAD3A web interface coupling 3D view, hierarchy tree, and derived semantic results; semantic records can drive explicit subset selection in the 3D rendering. Taken from Lupinetti et al. [44].

an end point, while the interaction required to iteratively refine large semantically defined candidate sets into a small, verifiable working set remains comparatively under-specified. In dense industrial assemblies, even correct semantic predicates can return many visually similar instances, making refinement and confirmation the dominant interaction bottleneck. Even after non-relevant categories are hidden, users must still select specific instances, often among many similar components, while preserving spatial context. This tension is visible in occlusion-management literature as well, where techniques may increase visibility, yet still leave the user with ambiguity if many plausible targets remain [24, 41]. Consequently, the most interaction-relevant semantic mechanisms are those that convert semantic intent into concrete candidate sets and that provide user-controllable pathways to refine those sets when necessary.

3.2.3.3 Data and Knowledge Requirements: Identifiers, Ontologies, and Reasoning

Semantic interaction is only as reliable as the underlying semantic substrate [10]. Work on semantic enrichment of point clouds explicitly highlights that end-to-end semantics often require a service-oriented stack: acquisition, segmentation/classification, data management, and a knowledge layer that can represent domain concepts and support queries [60].

A recurring limitation is that constructing and maintaining a suitable ontology is itself non-trivial, particularly in heterogeneous environments and evolving facilities. These observations are directly relevant to Digital Twins, where full metadata coverage may not exist and interaction design must therefore tolerate incomplete semantics and support progressive enrichment rather than assume comprehensive, clean labels.

A second implication is that semantics can support adaptive interaction when the system can reason over context. Conceptual frameworks for adaptation and personalization (Fig. 3.18) propose using user models and environmental descriptors (e.g., density and occlusion) to select or adjust interaction techniques dynamically [49, 50].

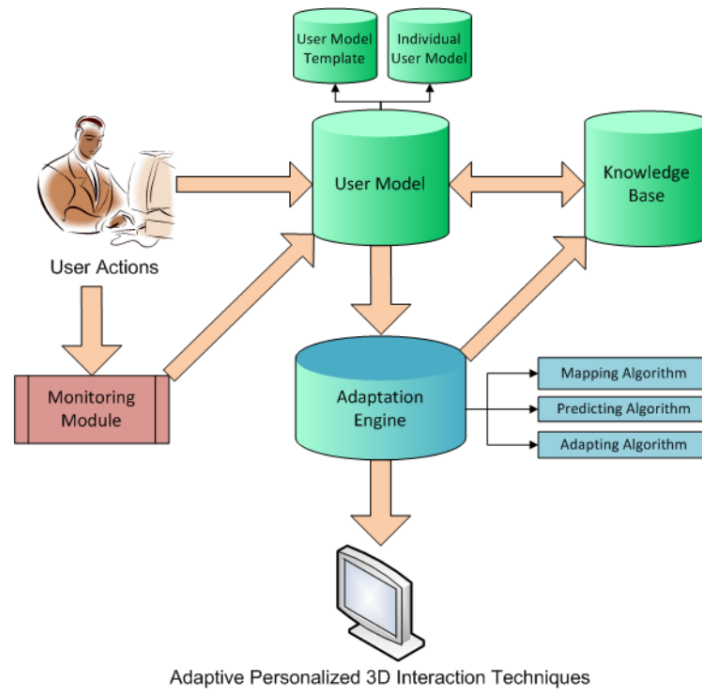


Figure 3.18: Conceptual framework for adaptive personalized 3D user interfaces: information about users’ working methods, performance, preferences, and abilities is gathered to build knowledge that supports adaptation and personalization, either by selecting the most suitable 3D interaction technique or by regulating how a technique behaves. Taken from [50].

Although these works are not technique contributions *per se*, they reinforce an important design direction in which semantic context, such as object identity, multiplicity, and occlusion, can guide the choice between interaction modes, including immediate pointing, progressive refinement, and visibility manipulation. Related context-awareness approaches similarly argue for a semantic reasoning layer that can trigger tool adaptations, even if practical deployment remains challenging due to uncertainties in intention recognition [20]. In dense industrial models, such “semantic control loops” are particularly compelling because they align with how engineers switch strategies depending on task phase (broad exploration vs pinpoint verification).

3.2.3.4 Implications for Dense Industrial Digital Twins

For dense industrial Digital Twins, the interaction value of semantic enrichment is best understood as a reduction in interaction entropy. Instead of forcing users to resolve ambiguity through repeated pointing and navigation, semantic structure allows systems to present meaningful partitions of the model (by asset class, subsystem, location, or status) and to translate those partitions into both filtering operations and explicit selection sets.

However, the literature also underscores that semantic approaches introduce their own constraints, as they depend on stable identifiers, consistent labeling practices, and maintainable knowledge representations [27, 60]. In practice, this motivates interaction designs that degrade grace-

fully when metadata is partial, allow users to validate and refine system-generated candidate sets, and preserve manual control for final confirmation.

If semantics enables “selection by meaning,” then language-based interaction can be seen as a mechanism for expressing semantic intent at scale, provided that the system can ground that intent to concrete objects reliably. The critical question is therefore not whether language can be used, but how it can be grounded, verified, and combined with manual interaction without introducing unacceptable uncertainty.

3.2.4 Natural-Language Interaction for 3D Environments

Following the semantic enrichment perspective in 3.2.3, several systems propose natural-language interaction as an additional modality for expressing semantic intent in 3D environments. In these approaches, language is treated as a query interface for specifying targets and candidate sets.

Natural-language interaction promises an appealing abstraction for dense 3D environments because it can bypass the mechanical overhead of manual pointing and navigation when the user’s intent is naturally expressed as a semantic predicate (e.g., “all dipole magnets in sector 5,” “assets with inspection due next month”). However, language-based interaction introduces a new set of challenges that are particularly pronounced in dense industrial scenes, as the system must map underspecified linguistic references to concrete objects in 3D, resolve ambiguity under occlusion, communicate uncertainty, and preserve user agency when automated interpretations are imperfect [3, 45]. This subsection reviews the foundations of grounding language in 3D environments, the recent shift toward LLM-mediated interaction, and the implications for reliability and human control in high-density models.

3.2.4.1 Grounding Language in 3D Scenes: Task Formulations and Bottlenecks

A useful way to understand language-based 3D interaction is through the task formulations developed in the 3D vision-language community, which operationalize “language grounding” as the joint problem of answering a query and localizing relevant objects. ScanQA formalizes 3D question answering as a task where the system must produce both a textual answer and corresponding 3D object bounding boxes, reflecting the requirement that language be tied to explicit spatial referents rather than remaining a purely linguistic response [3]. Figure 3.19 illustrates this operational definition of grounding, answering the question while simultaneously returning the spatial referents (3D bounding boxes). This matches the requirement that Digital Twin interfaces provide actionable, highlighted sets rather than purely textual responses.

This explicit grounding requirement parallels what a Digital Twin interface must ultimately provide, namely actionable selections or highlighted object sets that can drive inspection and metadata retrieval. ScanQA also illustrates a core issue that becomes more severe in dense scenes. Many queries are inherently ambiguous unless the system can disambiguate among multiple similar objects, or unless the user provides additional constraints.

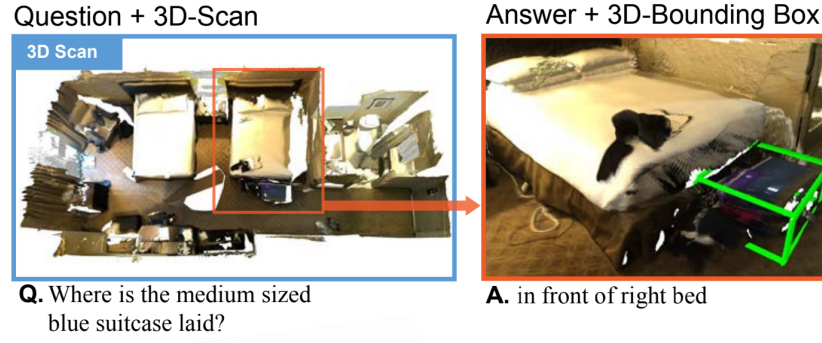


Figure 3.19: ScanQA task framing: a natural-language question over a 3D scene yields a textual answer and grounded 3D object references (bounding boxes). Taken from Azuma et al. [3].

Frameworks such as Chat-Scene (Fig. 3.20) address related limitations by representing a 3D scene as a sequence of object identifiers coupled with object-centric embeddings, enabling an LLM to reference objects through stable tokens rather than relying on raw geometry alone [32].

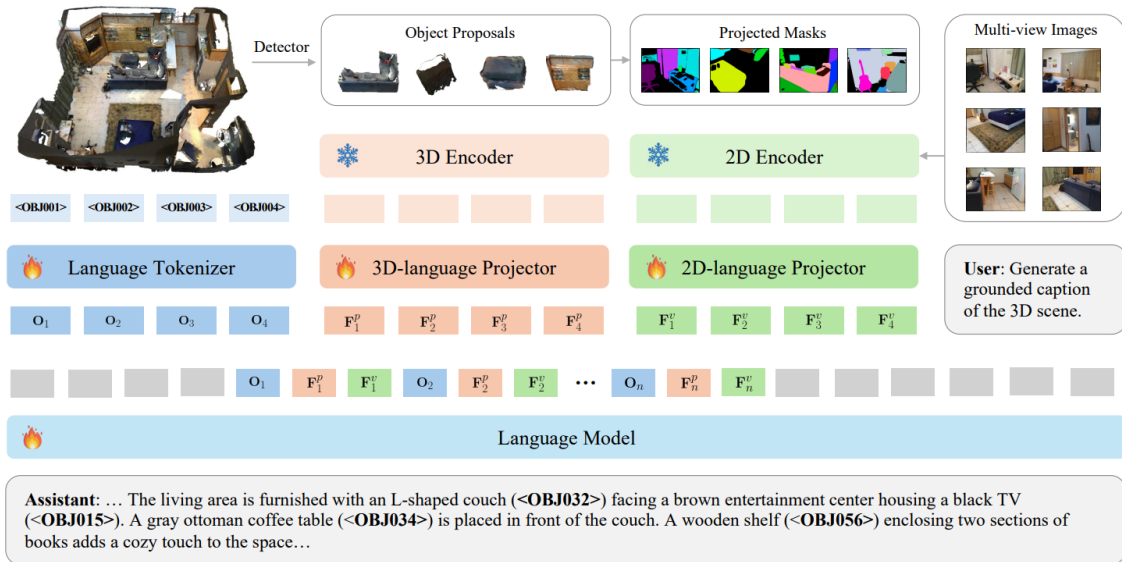


Figure 3.20: Chat-Scene overall model architecture: a point cloud scene is decomposed into object proposals via a pre-trained detector; 3D and 2D encoders extract object-centric representations that are projected into the language model embedding space. Unique object identifiers are added to the language model vocabulary and linked to proposals, enabling object referencing and grounding. Scene embeddings are formed as a sequence of object-level embeddings and fed into the LLM. Taken from Huang et al. [32].

The emphasis on identifiers is notable for industrial Digital Twins as it aligns closely with the practical need to inject or preserve stable IDs (e.g., PLM identifiers) so that conversational references can be grounded to the correct object instances even when visual similarity is high. These works suggest that language-based interaction in 3D becomes tractable when the system is not required to infer object identity solely from visual appearance, but can instead rely on an

explicit object inventory that the language model can reference.

3.2.4.2 From Language Interfaces to LLM-Mediated Interaction: Capabilities and Failure Modes

Recent interest in Large Language Models has shifted the discussion from specialized query interfaces toward more flexible conversational interaction [45]. Survey work cataloging LLMs for 3D tasks emphasizes that LLMs enable open-ended instruction following, reasoning over semantic attributes, and zero-shot generalization to new categories, making them attractive as interaction mediators for complex 3D data. At the interface level, this translates into a capability to accept user intent in a natural form, often closer to the language engineers already use, rather than forcing users into rigid filters, command syntax, or hierarchical browsing.

At the same time, the literature is clear that LLM-mediated 3D interaction is not equivalent to solving selection problems. As Chen, Grubert, and Kristensson [15] finds, it relocates the difficulty from motor control to semantic grounding, uncertainty management, and feedback design. Their study shows that users exploit language to perform higher-level bulk operations efficiently, but also encounter breakdowns due to recognition errors, unclear feedback, and the risk of incorrect system actions that disrupt the spatial task flow. The study is particularly informative because it shows that users compensate for these limitations by adopting interaction strategies that combine modalities, using speech for high-level intent and pointing for verification or fine-grained operations. This empirically supports the claim that language-based interaction is most effective when it is paired with an explicit mechanism for user confirmation and refinement rather than treated as a fully autonomous command channel [15].

Table 3.5 summarizes representative LLM-enabled capabilities and recurring failure modes reported in survey and interaction-study literature, and attributes each breakdown primarily to model behavior, UI/interaction design, or both. Two gaps recur: limited support for previewing and verifying intended targets before execution, and limited mechanisms for predictable error correction. In addition, many approaches assume curated object lists or clean identifier/metadata mappings, leaving grounding fragile in dense, occluded models where semantics are incomplete or inconsistent.

3.2.4.3 Visual Prompting and Multimodal Grounding for Disambiguation

A major recent trend is the recognition that language-only interfaces are fragile when the user’s referent is visually ambiguous. LL3DA explicitly operationalizes this insight by incorporating interactive visual prompts, like clicks and bounding boxes, alongside text, improving the model’s ability to understand and plan actions in 3D scenes when the user provides lightweight spatial anchoring [16].

While LL3DA is presented in the context of a 3D vision-language model, its design implication is directly applicable to interface design. Allowing a user to point at an exemplar can transform an

Table 3.5: LLM-mediated 3D interaction: representative capabilities and failure modes reported in survey and interaction-study literature. Primary limitation indicates whether the dominant breakdown is attributable to model behavior (Model), interface/feedback/integration design (UI/Interaction), or both (Mixed). Adapted from Chen, Grubert, and Kristensson [15] and Ma, Smart, Bhalgat, et al. [45].

LLM-enabled capability	Operational benefit in dense 3D	Typical failure mode	Primary limitation
Bulk intent expression (set-level commands)	Fast set creation + batch edits	Over-broad edits; misinterpreted intent / recognition errors	Mixed
Semantic abstraction / open-vocabulary referring	Query by type/attributes vs geometry-first navigation	Ambiguous referents under repetition; underspecified instance selection	Model
Multi-turn dialogue (iterative refinement / memory)	Incremental constraint refinement (broad → precise)	Context drift; unclear system state (“what is selected?”)	UI/Interaction
Reasoning over semantic/spatial constraints	Compositional requests (filter → group → act)	Plausible-but-wrong inferences when evidence/metadata is incomplete	Model
Natural-language justification / explanations	Supports validation when tied to evidence	Misleading rationales not matching executed predicate (“explanation hallucination”)	Model
Multimodal division of labor (language + pointing)	Language scopes; pointing confirms/disambiguates	Mode-switch costs; unclear commit/feedback; latency	UI/Interaction

ambiguous instruction (“select that valve”) into a grounded reference (“select objects of this type-/ID/class”) without requiring complex manual selection of every instance. Figure 3.21 summarizes this interaction pattern. The user provides an intent in natural language and resolves ambiguity with minimal spatial anchoring (e.g., a click), enabling grounded reference without exhaustive manual picking.

Chat-Scene similarly emphasizes object identifiers and object-centric representations as a way to stabilize referencing in 3D scenes [32]. Taken together, these works point toward a pragmatic interaction pattern in which natural language is used to express intent, while minimal user-provided grounding signals, such as selecting a representative object, defining a region, or confirming a candidate list, are employed when ambiguity is high.

For dense industrial models, where repeated parts and occlusion are common, this pattern can be more realistic than attempting fully automatic language grounding from geometry alone.

3.2.4.4 Open-Vocabulary Recognition: Limits of Inference-Only Approaches

A key question for industrial Digital Twins is whether language-based selection can rely on AI inference (recognizing objects from geometry/appearance) or must rely on explicit metadata. Open-vocabulary 3D object detection methods attempt to recognize novel categories beyond a fixed label set by aligning 3D representations with 2D vision-language models. CoDAv2 (Fig. 3.22) is an approach consisting of a framework that improves novel object detection through data enrichment

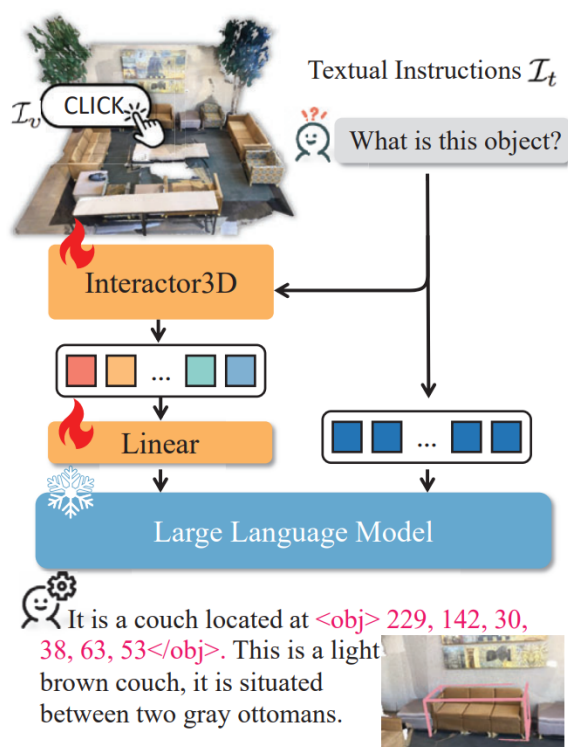


Figure 3.21: LL3DA interaction pipeline illustrating multimodal grounding via text plus lightweight visual prompts (click). Taken from Chen, Chen, Zhang, et al. [16].

and refined cross-modal alignment mechanisms, highlighting both progress and persistent constraints such as long-tail scarcity and performance degradation as the number of novel categories increases [13]. For interaction design, the implication is that inference-driven recognition can be powerful for broad semantic queries, but it may be breakable in the very regimes that matter for dense industrial scenes, such as rare component types, occluded structures, and visually similar instances.

From a systems perspective, this reinforces a conservative design principle in which conversational interfaces ground their actions in stable identifiers and curated semantic metadata when available, while treating open-vocabulary inference as a supporting mechanism rather than the sole authority. In environments where the cost of an incorrect selection is high, AI-driven suggestions must be accompanied by transparent presentation of evidence (e.g., why a set of objects was selected) and easy user override [1].

3.2.4.5 Positioning Conversational 3D Systems: Explanatory Oracles versus Operational Interfaces

Not all conversational 3D systems target operational selection tasks. VOICE illustrates a “visual oracle” paradigm where an LLM mediates explanation and guided exploration of 3D scientific visualizations, prioritizing interpretability and narrative guidance rather than precise, object-level selection in dense assemblies [35].

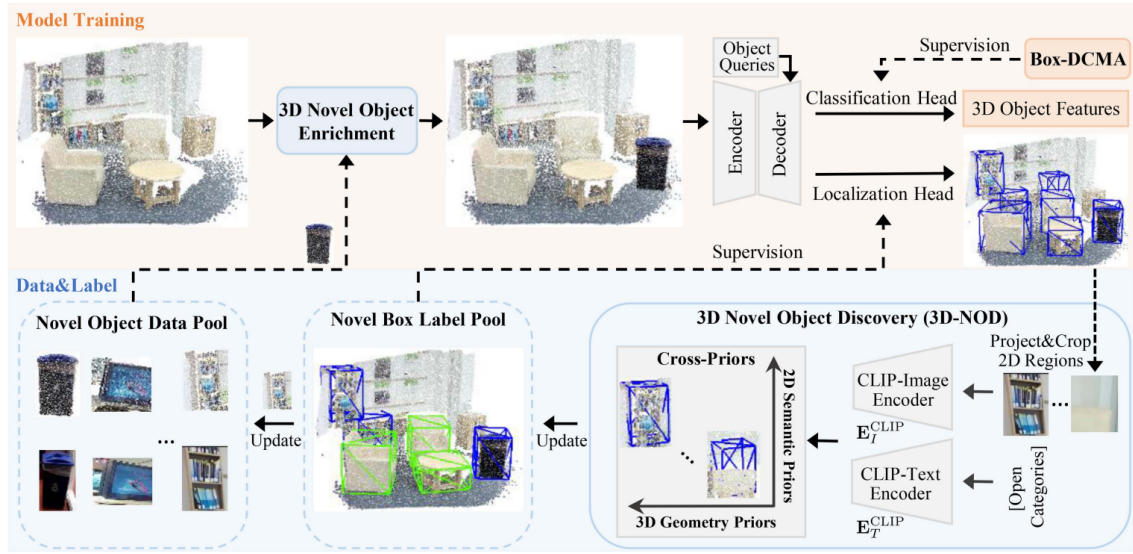


Figure 3.22: Overview of the CoDAv2 open-vocabulary 3D object detection framework: the system integrates a transformer-based 3D detector (3DETR) with 3D Novel Object Discovery using 3D geometric priors and 2D open-vocabulary semantic priors (from CLIP), an enrichment strategy that increases the occurrence of discovered novel objects during training, and a discovery-driven cross-modal alignment module extended with 2D box guidance to improve foreground/background discrimination. Taken from [13].

While such systems demonstrate the feasibility of real-time conversational control over 3D visualizations, their primary value for industrial Digital Twins lies in clarifying what is not solved by conversation alone: precise grounding, verification under occlusion, and task-critical correctness constraints. This distinction is useful when motivating the need to couple conversational capabilities with explicit semantic inventories and manual confirmation mechanisms in engineering environments.

3.2.4.6 Summary: Why Language Helps, but Cannot Replace Manual Selection

In summary, natural-language interaction introduces a powerful abstraction for expressing intent in dense 3D environments, particularly for bulk operations and semantic queries. However, the literature consistently indicates that language interfaces must confront three interrelated challenges:

- (i) Ambiguity and underspecification in user requests
- (ii) Rounding language to concrete objects under occlusion and repetition
- (iii) Trust and agency issues arising from opaque or incorrect system actions

The most promising direction is therefore not replacement but integration, namely combining language with lightweight spatial prompting (e.g., clicking exemplars) and stable identifiers to achieve reliable, user-verifiable semantic selection.

3.2.5 Hybrid Interaction: Combining Manual and AI-Assisted Techniques

The preceding subsections motivate a central design conclusion that neither manual selection nor language-based interaction is sufficient on its own for dense industrial 3D environments. Manual techniques offer precision and direct control, but they impose high interaction cost under occlusion and repetition. Conversely, natural-language interaction can express intent at the appropriate semantic level, yet it remains fragile when grounding is ambiguous or when system confidence and feedback are not appropriately communicated.

Hybrid interaction models emerge as a principled response to this trade-off, combining complementary modalities, manual pointing for confirmation and local disambiguation, semantic and language-driven mechanisms for candidate-set reduction and high-level intent expression, while adapting the interface to the user's context and the scene's properties [15, 50].

In this subsection, hybrid models are discussed in two layers. First, adaptive interaction work shows how systems can switch among manual techniques based on user performance and environment conditions, providing an early foundation for hybrid thinking even before conversational interfaces became feasible. Second, recent AI-enabled systems demonstrate how language and pointing can be combined into practical workflows where users retain agency while benefiting from semantic abstraction and automation.

3.2.5.1 Adaptive Selection: Switching Between Manual Techniques

A long-standing observation in the selection literature is that technique effectiveness is highly context-dependent, as occlusion, density, and target visibility systematically influence which manual technique is preferable. Rather than seeking a universally optimal selection technique, adaptive approaches propose dynamically selecting the best technique for the current environment and user state.

Early conceptual frameworks formalize this by combining a user model, a knowledge base, and an adaptation engine capable of selecting or tuning interaction techniques based on measurable environmental factors [50]. Subsequent work argues that such adaptation can reduce cognitive load and frustration by offloading the strategy choice from the user to the system, especially for novices in complex environments [51].

Empirically, the feasibility of adaptation hinges on two issues. First, the system must infer environment state robustly (e.g., density and occlusion) and map it to an interaction strategy. Second, the adaptation must be intelligible and non-disruptive, since technique switching can introduce confusion if users cannot predict the system's behavior or if the interface changes mid-action [14]. This tension is particularly relevant in dense industrial models where users operate under time pressure and must maintain trust in the interface.

As a result, the strongest implication of adaptive selection research for industrial Digital Twins may be conceptual rather than prescriptive. It provides evidence that multiple selection strategies are needed, and that their coordination must respect user expectations and provide clear feedback when switching occurs.

Table 3.6: Adaptive selection approaches: what is adapted (scene vs user factors), what is changed (technique switching vs parameter tuning), and key constraints for deployment in dense 3D environments. Adapted from Cashion, Wingrave, and LaViola [14] and Octavia, Raymaekers, and Coninx [50, 51].

What is adapted	What changes	Signals used	Representative framing (literature)	Key risk / constraint
Scene difficulty (density, occlusion, visibility)	Switch technique (discrete choice)	Estimated density/occlusion; target visibility proxies; task phase	Adaptation engine selects an interaction technique based on environmental descriptors, supported by a knowledge base	Predictability: switching can confuse users if behavior changes mid-action
Scene difficulty (continuous variation)	Tune parameters (continuous)	Same as above + stability of user motion	Rather than switching paradigms, adjust parameters within a technique (e.g., sensitivity/control–display mappings) using context descriptors	Transparency of control: users must understand why behavior varies
User factors (expertise, performance, preferences)	Switch or tune (personalization)	Performance history; error rates; interaction tempo; explicit preference	Conceptual user models drive adaptation/personalization decisions; goal is to offload “strategy choice” from the user	Trust/agency: adaptation must be intelligible and user-controllable
Hybrid context (scene + user)	Constrained switching (guardrails)	Scene signals + user signals; uncertainty estimates	Practical implication: multiple strategies are needed, but coordination must respect expectations and provide explicit feedback when switching occurs	Disruption cost: avoid unexpected mode shifts; provide confirmation/undo and clear state

Table 3.6 summarizes adaptive selection approaches by distinguishing whether they adapt to scene factors or user factors, and whether they switch techniques or tune parameters, while highlighting the predictability and user-confusion risks reported in prior work.

3.2.5.2 Hybrid Workflows: Language for Intent, Pointing for Verification

Recent work on LLM-assisted 3D interaction provides concrete evidence that hybrid interaction is not only technically feasible but often emerges naturally in user behavior. In an LLM-assisted VR scene-editing setting, users progressively adopt strategies that combine high-level language commands for bulk actions with direct pointing or local interaction for precise operations, reflecting an implicit division of labor between modalities [15]. Figure 3.23 illustrates these two observed strategies in ASSISTVR providing a concrete example of the modality division that motivates hybrid selection workflows.

Although this work is situated in immersive interaction and speech input, the interaction principle generalizes. Language is most valuable for expressing semantic intent and high-level structure, whereas manual input remains critical for reference, confirmation, and fine-grained control, especially when the system’s interpretation is uncertain.

This hybrid principle is further reinforced by 3D vision-language work that explicitly incorporates interactive prompts. LL3DA integrates textual instructions with user-provided visual cues



Figure 3.23: Example workflow in ASSISTVR showing two common multimodal strategies: bulk modification via language-driven set selection and incremental exploration via local refinement; the final scene matches the target task state. Taken from Chen, Grubert, and Kristensson [15].

(clicks and bounding boxes), improving the model’s ability to perform grounded reasoning and planning in 3D scenes [16].

Conceptually, such systems treat user input not as a single channel but as a feedback mechanism that allows the user to correct ambiguity at minimal cost. In dense environments, where multiple objects satisfy a linguistic description, an exemplar click can convert an underspecified request into a grounded selection intent without requiring full manual selection of all instances.

Similarly, object-identifier-based representations proposed in Chat-Scene (Fig. 3.20) demonstrate that stable object inventories can enable LLMs to refer to and manipulate 3D content with reduced hallucination risk, because selection is performed over known objects rather than inferred pixel-level entities [32]. For industrial Digital Twins, this is directly compatible with the use of persistent IDs (e.g., PLM identifiers) embedded in the scene graph.

Taken together, these works motivate a pragmatic hybrid interaction pattern for dense industrial models in which users first express intent in natural language to generate a candidate set based on type, location, or metadata constraints, then inspect the results through representations that expose uncertainty (e.g., highlighted sets), and finally perform low-effort manual confirmation or refinement using clicks, filters, or progressive refinement operations.

This pattern also aligns with the limitations of inference-based open-vocabulary recognition, as even state-of-the-art methods remain sensitive to background confusion and long-tail scarcity, suggesting that AI inference should often be treated as suggestive rather than authoritative when operational correctness is required [13].

3.2.5.3 System-Level Considerations: Trust, Predictability, and Error Recovery

Hybrid interaction models introduce new system-level requirements beyond those of purely manual selection (Table 3.7). First, they must support predictability, as users need to understand what the system will do in response to both language and manual actions, and how the two channels interact. Adaptive technique switching already shows that even within manual selection, unpredictable changes can harm user confidence [14]. In AI-assisted interaction, this issue is amplified

because errors can be semantically plausible yet operationally incorrect. Accordingly, hybrid designs must provide feedback that is both local (what is selected, where, and why) and semantic (what predicates or interpretations led to the selection).

Second, hybrid systems must support error recovery as a first-class interaction pathway. In dense scenes, errors are not rare, therefore, the interface should make it easy to undo, refine, and constrain selections without forcing the user back into exhaustive viewpoint navigation. The shift from “selection as an event” to “selection as an iteratively refined set” is consistent with the progressive refinement paradigm and with interaction patterns observed in LLM-assisted systems where users converge on efficient strategies through iterative correction [4].

Table 3.7: Hybrid interaction design requirements for trustworthy semantic selection in dense 3D models: predictability, transparency, uncertainty communication, confirmation, and error recovery, grounded in prior findings on adaptive selection, progressive refinement, and LLM-assisted 3D interaction. Adapted from Bacim, Kopper, and Bowman [4], Cashion, Wingrave, and LaViola [14], Chen, Grubert, and Kristensson [15], and Dennemont et al. [20].

Requirement	Why it matters in hybrid selection	Concrete UI / interaction mechanisms
Predictability & mode awareness	Users must anticipate what language vs. manual input will do, and how they compose; semantically plausible errors are especially damaging to trust	Explicit “preview-then-commit” stages; clear mode/state indicators (e.g., “filtering”, “candidate review”, “committed selection”); constrained automation (no silent global actions)
Local + semantic feedback (“what, where, why”)	Hybrid systems must expose both the selected set and the interpretation/predicate that produced it; otherwise users cannot validate intent-to-selection mapping	Highlight set + count; ranked list; show predicate summary (type/location/metadata constraints); selection inspector with parent/subsystem context; evidence panel when inference is used
Uncertainty communication	Ambiguity is common under repetition/occlusion; interfaces must surface uncertainty so users can refine rather than guess	Confidence / ambiguity cues; “top- <i>k</i> candidates” presentation; disambiguation prompts (click exemplar, box/region); ghosting non-candidates to preserve context
Confirmation mechanisms (low-effort verification)	Language works best when paired with lightweight grounding and explicit confirmation rather than autonomous execution	One-click “Confirm set”; “Confirm exemplar → expand to similar/linked IDs”; progressive refinement on candidate set; quick isolate/section/explode on candidate neighborhood
Error recovery as a first-class path	Errors are inevitable in dense scenes; recovery must be cheaper than repeating navigation and repicking	Undo/redo with semantic history (what constraint changed); “refine constraints” controls; reversible visibility/decomposition operations (collapse, un-isolate); safe rollback for batch actions
Ground-truth anchors (stable IDs & curated metadata)	Reliable operational selection requires stable referents; inference-only grounding is brittle in industrial regimes (rare parts, occlusion, long tail)	Selection and actions resolved over an explicit object inventory (IDs/aliases); ID injection into scene graph; show ID/alias in inspector; deterministic back-end predicates where possible
Autonomy boundaries & governance	Context-aware reasoning can help, but deployment is hard when intention recognition is uncertain; autonomy must be bounded to preserve agency	Human-in-the-loop defaults; conservative automation; explicit “assistive suggestions” vs. “authoritative actions”; audit trail of decisions and predicates

Finally, hybrid interaction highlights the central importance of ground truth anchors (stable

object identifiers and metadata) especially in heterogeneous Digital Twin environments where semantic completeness may be partial.

While context-aware semantic reasoning engines have been proposed as a means to adapt interaction tools in non-semantic environments, practical deployment remains challenging due to difficulty in reliably monitoring user intentions and managing system autonomy [20].

This underscores a practical direction for industrial systems, namely the use of explicit identifiers and curated metadata wherever possible, while employing AI primarily as an interaction mediator that accelerates access to these anchors rather than as the sole source of truth.

3.2.5.4 Summary: Relevance to Dense Industrial Digital Twins

Hybrid interaction models provide a coherent framework for dense industrial Digital Twins because they explicitly address the core bottleneck identified throughout the literature, namely the need to reliably map user intent to the correct objects under conditions of density and occlusion without imposing excessive manual effort.

Adaptive selection research supports the idea that multiple manual techniques are needed and that their coordination must preserve user predictability.

Recent AI-assisted interaction research demonstrates that users naturally adopt hybrid workflows and that interactive grounding cues can improve disambiguation in 3D tasks. These findings collectively motivate the hybrid interaction paradigm adopted in this dissertation, where advanced manual interaction techniques for decomposition and occlusion management are complemented by an AI-assisted conversational interface grounded in stable object identifiers.

3.3 Synthesis: From the Literature to the Approach

The five themes reviewed above approach the same environment from different directions. This section draws them together and sets out how they shaped the approach taken in this dissertation.

The first theme establishes that density and occlusion are not a single obstacle but a compound regime, combining geometric density, target multiplicity and deep hierarchical structure. Revealing a hidden target is not sufficient on its own, because the user still has to establish that the component now visible is the one intended.

The second theme shows that manual selection techniques distinguish targets through geometric and visual cues. Those cues lose their discriminating power when components repeat, and the techniques precise enough to compensate generally depend on input capabilities that do not transfer to a desktop setting.

The third theme establishes that metadata allow components to be addressed by meaning rather than by visibility alone. The limitation is that even a correctly formulated predicate commonly returns more candidates than a user can inspect, and the metadata itself is often partial or inconsistent across a model assembled over decades.

The fourth theme shows that natural language expresses intent with a flexibility that direct manipulation cannot match. Its weakness is grounding, which degrades under exactly the conditions

found in industrial models, namely occlusion and repetition. Actions whose basis is not visible to the user further erode trust in the system.

The fifth theme indicates that users adopt hybrid workflows of their own accord, that reliable operation in industrial settings depends on stable referents rather than on inference alone, and that system autonomy has to be bounded if the user is to retain agency.

Read together, the five themes converge on a single gap. Each line of work improves one stage of the interaction, whether by revealing what is hidden, pointing more precisely, filtering by meaning or expressing intent flexibly, yet none provides a way for the user to verify that the system arrived at the right component and to refine the result when it did not. Manual techniques give control but no scalable way to distinguish repeated parts. Semantic filtering gives meaning but returns candidate sets too large to check by hand. Language gives expressiveness but its grounding degrades precisely under the conditions that characterize industrial models. The missing element is not a better technique within any one of these families, but a way of combining them in which language proposes and the user confirms.

Chapter 4

Problem Approach

Having established the context and the relevant prior work, this chapter presents the approach taken to address the research problem: enabling domain engineers to interact efficiently with large-scale 3D engineering models for maintenance planning purposes. The chapter begins by defining the problem in concrete terms, identifying the specific barriers that existing tools leave unresolved. It then describes the proposed solution, explaining the architectural and design decisions that informed it and outlining how they relate to the defined challenges. The chapter concludes with a risk analysis that identifies the uncertainties encountered during development and the strategies used to mitigate them.

4.1 Detailed Problem Definition

At CERN, scientists and engineers plan and execute maintenance campaigns on very large and complex particle accelerator facilities. These campaigns involve thousands of individual tasks spanning equipment removal, installation, consolidation, and routine maintenance, distributed across tunnel sectors that contain tens of thousands of distinct components. The maintenance planning workflow involves two largely independent sources of information: external scheduling databases that track the tasks, and 3D visualization tools that render the physical model. Although a web-based 3D tool exists for displaying which sections of the accelerator are under maintenance, it operates on simplified models and does not integrate deeply with the engineering metadata attached to individual components. Engineers who need to locate a specific part, understand what work is scheduled on it, or reason about what surrounds it must switch between systems and reconcile identifiers manually.

What makes this workflow slow and error-prone is less the quality of the information than the difficulty of reaching it through the model, and three difficulties dominate.

1. **Reaching a component.** Conventional point-and-click interaction becomes cumbersome in dense scenes, where a target may be occluded, deeply nested in the hierarchy, or identifiable only by an engineering name that does not match its visual appearance. An engineer who knows they need the wire scanner in period 4, but not where to look in the viewport, cannot

effectively use a scroll-and-click hierarchy. A complementary way of expressing intent, in the engineering terms the work is actually described in, is needed to bridge the gap between what the user knows and where the object sits.

2. **Reasoning about physical context.** Maintenance planning frequently depends on the spatial relationships between components: what is adjacent to a given magnet, what equipment falls within a tunnel sector, or what is physically connected to a reference assembly. These questions cannot be answered by metadata search alone; they require operating on the geometry of the scene.
3. **Separating geometry from meaning.** The metadata that identifies a component and records the work scheduled on it, its identifiers, operational status, maintenance history, and safety information, is maintained rigorously, but in systems that are external to the model. Locating a part and understanding its context are therefore two disjoint manual steps, and the engineer must move between environments and reconcile identifiers by hand even when the data itself is complete.

Addressing these difficulties is constrained by the conditions of the deployment, which shape any viable solution. A single tunnel sector contains very large numbers of objects, so search and spatial operations cannot traverse the full scene graph on every query and must instead run over purpose-built structures if the application is to stay interactive. Component metadata comes from two independent sources: the PLM system, which records engineering identity such as document number, revision, and description, and the Layout system, which records installation context such as tunnel position and layout name. Engineers mix identifiers from both within a single reference, so resolution cannot assume one naming scheme; the PS Booster model used here carries only PLM metadata, but a system tied to a single source would not generalize to accelerators where both are populated. More generally, treating metadata as a set of independent sources rather than a single fixed schema keeps the design open to additional information layers such as further engineering systems or task and safety data beyond those used here, without reworking the core resolution mechanism. Validated task data, finally, is reached through an external authoring service rather than read from the model.

Together, the interaction difficulties define what the work must solve and the deployment conditions define the limits within which it must be solved: a prototype that makes the scene searchable in the terms engineers use, answers spatial questions over its geometry, and links components to their validated task data, exposed through both a conventional window-based interface and a conversational agent operating over the same scene.

4.2 Proposed Solution

The proposed solution is a hybrid interaction prototype for NVIDIA Omniverse that combines conventional point-and-click 3D interaction with a natural-language conversational interface, both

operating over the same dense engineering scene. The central argument is that for maintenance planning in large-scale accelerator models, the two modalities address different aspects of the interaction problem and are more effective together than either would be alone. Manual interaction provides precision and direct spatial awareness once a user knows what to look at; conversational interaction removes the barrier of needing to know internal identifiers or exact hierarchy positions to get there.

The prototype is built around a shared backend that serves both modalities identically. This backend pre-processes the scene into a searchable index, resolves natural-language references to USD prim paths, executes scene operations (navigation, selection, visibility), answers spatial proximity queries, and retrieves validated task-to-CAD assignments from the digital-twin authoring layer. Figure 4.1 gives an overview of this architecture, distinguishing the parts provided by the platform, the parts adapted from existing components, and the parts built for this work.

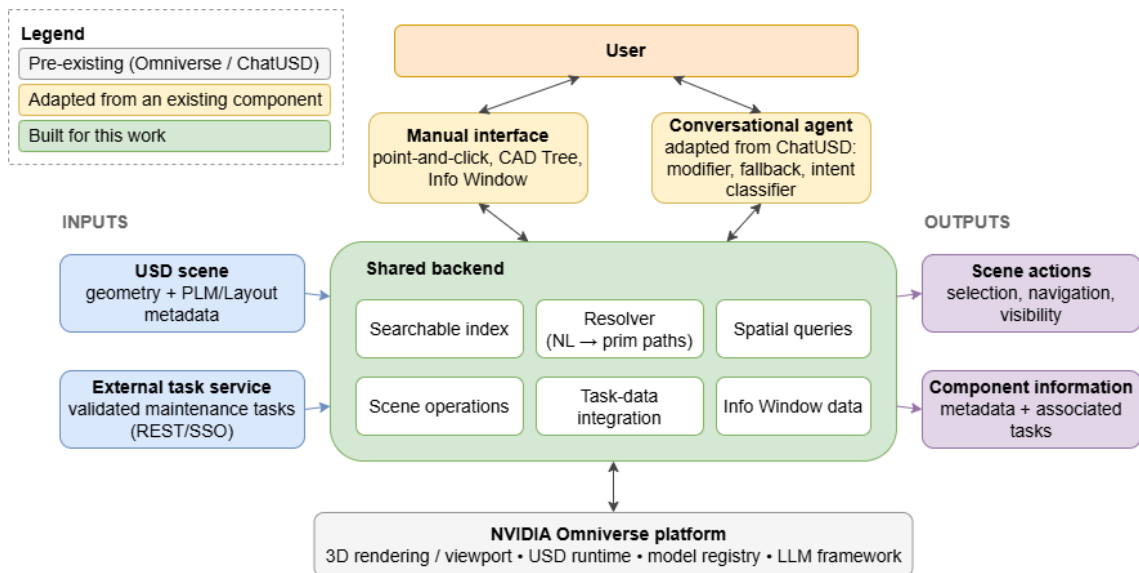


Figure 4.1: Overview of the proposed architecture. Both interaction modalities operate over a shared backend that draws on the same inputs (the USD scene and the external task service) and produces the same outputs (scene actions and component information). Color distinguishes what the platform provides, what was adapted from an existing component, and what was built for this work.

The prototype draws on the platform for everything below the interaction layer. The 3D rendering and viewport, the USD scene runtime, the model registry, and the language-model framework are provided by NVIDIA Omniverse and its ChatUSD bundle, and are used as they come. The conversational agent starts from the default ChatUSD workflow, whose tool set is oriented toward content creation, and reworks it toward locating and inspecting components that already exist in the loaded model. The remaining behavior is specific to this work: the searchable index, the deterministic resolver that turns a request into prim paths, the spatial-query layer, the task-data integration, and the Info Window and CAD Tree of the manual interface. It operates on two fixed external inputs, the USD scene, which carries both geometry and PLM/Layout metadata, and the external

authoring service from which validated maintenance tasks are retrieved, and produces two kinds of output: scene actions applied to the model (selection, navigation, and visibility changes) and component information surfaced to the user (metadata and the tasks associated with a selection).

The hybrid configuration adds two elements to the baseline Omniverse environment: an Info Window that displays PLM metadata and associated scheduling tasks for the currently selected component, and a conversational agent through which users can navigate, search, filter, and query the scene in natural language. The Info Window is also present in the baseline configuration, where it is the only interface addition. In that case the user must locate components through the standard Omniverse scene tree, which searches only against internal technical identifiers, and identify relevant tasks by clicking through the hierarchy manually. This contrast is precisely what the user study is designed to evaluate.

Building this on top of the platform proceeds in a sequence of steps. The dense scene must first be made searchable in the engineering terms used at CERN, since neither the raw USD hierarchy nor the platform's own search supports this. A free-form request must then be resolved, deterministically, to the specific components it refers to, and the system must be able to answer questions about physical context that depend on geometry rather than metadata. These capabilities are driven from natural language by adapting the conversational agent so that it calls them reliably, and exposed through the manual interface by the Info Window and the CAD Tree. Finally, components are linked to their validated task data from the external service, and the two configurations evaluated in the user study are assembled from these parts.

The contribution of this work is therefore not the 3D rendering infrastructure, the model registry, or the LLM framework, all of which are provided by NVIDIA Omniverse and its ChatUSD bundle. The prototype should instead be understood as the apparatus through which the research questions are investigated. Its backend services, conversational orchestration layer, and Info Window make the study possible, but they are not, by themselves, the main result. What this work contributes is an empirical and design-oriented account of how conversational interaction can be combined with conventional manual interaction for dense 3D engineering models. The first part of this contribution is empirical, providing evidence from a controlled comparison with CERN engineers on whether and how far pairing conversational interaction with conventional manual interaction improves the location and inspection of components in dense 3D engineering scenes. The second part is design knowledge, identifying the choices that keep the conversational layer predictable and make its spatial queries meaningful. These choices are not tied to this particular prototype or platform and can inform similar interfaces for other dense industrial models.

4.3 Choice of a Conversational Interface

A conversational interface is not the only way to address the difficulties and it is worth stating why the alternatives were judged insufficient on their own.

The simplest alternative is a free-text search field over the component metadata. This is a real improvement over scrolling the hierarchy and it is inexpensive to build, but it assumes the user can

supply text that matches how the component was recorded. Engineers frequently do not know the internal identifier or the exact description, and a search field offers no help when the remembered phrasing and the stored phrasing differ.

A second alternative is a structured or faceted query builder, in which the user narrows a candidate set by choosing values for attributes such as component type, layout region or period. Faceted interfaces work well when the schema is small and stable and the user knows which attribute carries the distinction being sought. Here the metadata comes from independent sources with uneven coverage, so the set of usable facets varies across models, and the user has to learn the schema before formulating a request. The approach also inverts the burden that this work is trying to remove, because it requires the user to express the query in the vocabulary of the system rather than in their own.

A third alternative is a formal query language over the metadata and the scene graph. This would be expressive enough to capture relational conditions, but it requires learning a syntax and an underlying schema, which is not a reasonable expectation for occasional use during maintenance planning.

A fourth alternative is to improve manual interaction alone, through richer hierarchy browsing and advanced selection techniques. This is a necessary part of the solution and is the reason the CAD Tree was built, but such techniques lose their discriminating power under repetition and that the most precise among them do not transfer to desktop input.

None of these alternatives addresses the relational and spatial requests such as asking what is physically connected to a given assembly. A relation of that kind is not an attribute that can be filtered, and it is not recorded anywhere in the metadata, so it has to be computed from the geometry in response to the request. Natural language is the one modality among those considered that lets a user state such a request without first knowing the vocabulary, the schema or the location of the component, and it accepts identity-based and relational requests through a single entry point rather than requiring a separate interface for each.

The conversational interface is not proposed as a replacement for the alternatives above. Where a request is a simple attribute lookup, a search field would serve as well, and the prototype accordingly resolves requests of that kind without consulting the language model at all. Where several components share a description and cannot be told apart by text, language cannot disambiguate them and manual confirmation remains necessary. These limits are the reason the approach adopted here is hybrid rather than conversational alone.

4.4 Risk Analysis

The main risks identified during planning and development, together with their likelihood, potential impact, and the adopted mitigation strategy, are summarized in Table 4.1 and discussed below.

R1: LLM reliability and consistency. Large language models may produce inconsistent outputs or struggle to conform to strict formatting constraints when interfacing with external tools

[45]. This can lead to system errors, such as malformed tool calls or the failure to emit a command altogether. This risk was considered high-likelihood and high-impact, since the entire conversational interface depends on well-formed model output. It was mitigated through explicit argument sanitization at dispatch time, a hierarchy of fallback mechanisms in the modifier layer, and an intent classifier that provides a last resort when the primary model path fails.

R2: Metadata quality and completeness. The usefulness of the index and the resolution pipeline depends on the metadata that the authoring pipeline has written into the USD scene. Missing or inconsistent metadata degrades search accuracy and may prevent task resolution. The prototype was designed to handle multiple metadata families and to degrade gracefully when some fields are absent, but the evaluation is ultimately limited to the PLM-only PS Booster model.

R3: Spatial query approximation. OBB-based intersection is an approximation of true geometric contact. For objects with thin protrusions, concave geometry, or complex inner structure, the OBB may be significantly larger than the actual occupied volume. This risk was accepted as a pragmatic trade-off: full mesh intersection would require a physics simulation pipeline that is not available for imported CAD geometry in the streaming environment, and the OBB approach was found to be accurate enough for the spatial queries in the evaluation tasks.

R4: Performance at scale. Dense USD scenes make full traversal prohibitively slow for interactive queries. The risk was addressed through the index-based iteration strategy: both search and spatial queries operate over the indexed item set rather than the full scene graph, and the two-stage AABB/OBB filtering pipeline further reduces the number of expensive intersection tests.

R5: Network dependency for task data. The REST API integration requires network access to the authoring layer and SSO authentication. In fully offline scenarios, task data would be unavailable. This risk was accepted as inherent to the live-data integration approach; the system is designed to degrade gracefully by continuing to provide metadata and spatial queries when the API is unreachable.

R6: Platform evolution. NVIDIA Omniverse and its bundled extensions are updated periodically, with potential breaking changes to the LLM model registry, the viewport API, and the agent framework. The risk was mitigated by discovering available models at runtime rather than hardcoding model identifiers, and by applying viewport settings defensively after layout loads to guard against deferred resets.

Table 4.1: Summary of likelihood and impact of each identified risk.

Risk	Likelihood	Impact
R1: LLM reliability	High	Incorrect or missing scene operations
R2: Metadata quality	Medium	Degraded search and task resolution
R3: Spatial approximation	Medium	False positives/negatives in adjacency
R4: Performance at scale	Medium	Slow response times
R5: Network dependency	Low	Task data unavailable offline
R6: Platform evolution	Medium	Breaking changes after updates

In summary, this chapter has defined the problem as a difficulty of interacting with dense

models whose semantic data is held separately, set out the conditions that constrain a solution, proposed a solution based on a shared backend with dual interaction modalities, and identified the risks that shaped implementation decisions. Chapter 5 documents the implementation of each component in detail, explaining the design choices made at each stage and the consequences that followed from them.

Chapter 5

Implementation

This chapter documents the development of the prototype built to address the previously defined problem. The conversational agent included with Omniverse is first adapted to find components that already exist in a model rather than to create new ones, which calls for a searchable representation of the scene and a pipeline that turns natural language into specific objects. The system then gains the ability to reason about where components sit in space, and the conversational layer is reinforced to handle the language model's variable behavior. Finally, two manual interface windows and the maintenance-task data they surface complete the hybrid environment that the next chapter evaluates.

5.1 The Source Data and Its Challenges

The scene reaches the prototype as a USD stage exported from CAD, carrying both the geometry and the engineering metadata attached to it. Three characteristics of that data shaped the design of everything described in this chapter.

The first is that component identity is spread across independent sources. Engineering identity, such as document number, revision and description, is recorded by the PLM system, while installation context, such as tunnel position and layout name, is recorded by the Layout system. Engineers mix identifiers from both within a single reference, so resolution cannot assume one naming scheme. Coverage is also uneven across models. The PS Booster model used in the evaluation carries only PLM metadata, whereas other facility models encountered during development carried both families fully populated. The design consequence is that metadata is treated as a set of independent families rather than as one fixed schema, so that resolution continues to work when a family is absent, with reduced search power rather than outright failure.

The second is that names were never standardized. The model was assembled from CAD designs authored over several decades, during which no consistent naming convention was enforced, so descriptions appear in English, French and German depending on when a design was created and who authored it. Text-based retrieval therefore cannot rely on a consistent vocabulary even

within a single metadata family. The consequences of this and the mitigation adopted are discussed in Section 5.9.

The third is that the exported hierarchy reflects the structure of the CAD export rather than the way engineers describe the installation. A single engineering component may appear as a subtree of mesh fragments beneath intermediate grouping nodes that carry no metadata and correspond to nothing an engineer would name. Any operation acting on what the user clicked or on what a search matched therefore has to establish which node of that subtree is the component the user means, which is the reason for the selection-promotion behavior described in Section 5.4.

5.2 A Searchable Representation of the Scene

The starting point for the conversational interface was the default agent workflow provided by ChatUSD, an extension that is included with NVIDIA Omniverse. ChatUSD interacts with the loaded USD stage through Python-USD code generation and scene-information retrieval, and operates over a tool set oriented toward content creation: searching and importing USD assets, instantiating new geometry, and modifying scene attributes. When this agent was tested on a CERN scene, two problems became apparent. First, response times for queries that involved searching the scene were unacceptably slow, informally on the order of 15 to 20 seconds per query, because the agent was traversing the entire stage on every request. Second, the available tools were not appropriate for the task at hand, which is locating, inspecting and selecting components that already exist in the loaded model rather than creating new ones.

The performance issue was not simply a matter of stage size. CERN scenes are produced by an export pipeline that converts the existing CAD models into USD, and the resulting stages contain a large amount of intermediate hierarchy nodes and transform groups that exist for reasons related to the conversion process rather than the engineering identity of the components. Traversing this structure on every query is wasteful, since the same object is visited many times across queries, and most visited objects do not carry the metadata that the search is interested in.

The decision adopted was to build a separate index at stage-load time and to redirect all subsequent queries against this index rather than against the live USD stage. The resulting pipeline, summarized in Fig. 5.1, builds a compact data structure once after the stage is loaded, persists it to disk, and reloads it in later sessions to avoid repeating the construction cost. It contains only the prims that carry meaningful engineering metadata, ignoring the intermediate structural nodes that the export pipeline produces. Queries that mention prim paths are still tied to the live USD stage, but the search itself runs against the index. The serialized index is keyed by a hash of the stage's root layer, so the cache is reused automatically when the same scene is reopened and rebuilt automatically when the upstream USD changes.

The index builder visits the Xform-type prims (the transformable nodes in the USD scene hierarchy that represent positioned objects) in the loaded stage and records the metadata stored in each one's custom attributes. This metadata falls into two families. The PLM attributes capture a component's engineering identity: document number, description, revision, drawing reference

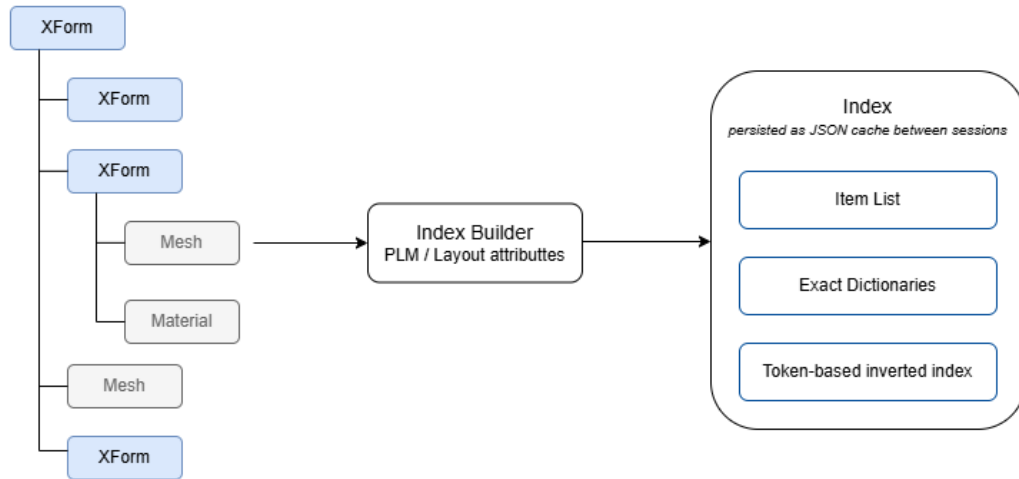


Figure 5.1: Construction of the index from the loaded USD stage.

and PLM identifier. The Layout attributes capture the physical installation context of beamline elements, the components that form part of the accelerator itself: layout name, type, tunnel location, floor and side. Not every object carries both. Structural components such as supports, racks and civil-engineering elements typically carry PLM information but no Layout data, because they are not part of the beamline and are not managed through the Layout database.

To make this concrete, consider a single beamline component as it is described across the two sources. In the PLM family it is identified by its engineering identity, through a document number, a textual description, a revision, a drawing reference, and a PLM identifier. In the Layout family the same component is described instead by its installation context, through a layout name, a component type, a tunnel location, a floor, and a side. In CERN’s source systems the two descriptions are maintained in separate databases and are connected only through shared identifiers, so an engineer who has located a component in one must look it up in the other by hand to obtain the full picture. In the exported USD stage, both families are written as custom attributes onto the single prim that represents the component, and it is this prim that the index records. Storing both descriptions against the same object is what lets a single query match a component either by its engineering identity or by its installation context, removing the manual cross-referencing that the separate source systems otherwise require.

Two lookup structures are constructed alongside the item list. Exact dictionaries map normalized identifier strings (document numbers, PLM IDs, layout names, region field values) directly to item indices, supporting constant-time retrieval for queries that match a known identifier. A token-based inverted index maps individual alphanumeric tokens to the items that contain them, supporting ranked retrieval for free-text queries.

Algorithm 1 sets out the construction in full. Three properties of it are worth drawing out. The traversal is performed once per stage rather than once per query, which is what removes the repeated full-stage traversal that made the original agent slow. Prims carrying neither family of metadata are discarded during the same pass, so the index holds only objects an engineer would

name, and its size is governed by the number of real components rather than by the size of the exported hierarchy. The two lookup structures are built in the same pass as the item list, which is what allows a later query to be answered either by constant-time identifier lookup or by ranked token retrieval without revisiting the stage.

Algorithm 1 Construction of the searchable index from the loaded USD stage.

Require: Loaded USD stage S

Ensure: Index I holding the item list, the exact dictionaries and the token index

```

1:  $h \leftarrow \text{HASH}(\text{ROOTLAYER}(S))$ 
2: if a serialized index exists for key  $h$  then
3:   return  $\text{DESERIALIZE}(h)$  ▷ cache hit, no traversal needed
4: end if
5:  $items \leftarrow \langle \rangle$ 
6:  $exact \leftarrow \{\}, tokens \leftarrow \{\}$ 
7: for all prims  $p$  of type Xform in  $S$  do
8:    $plm \leftarrow \text{READPLMATRIBUTES}(p)$ 
9:    $layout \leftarrow \text{READLAYOUTATTRIBUTES}(p)$ 
10:  if  $plm = \emptyset$  and  $layout = \emptyset$  then
11:    continue ▷ intermediate node produced by the CAD export
12:  end if
13:   $i \leftarrow |items|$ 
14:  append  $(\text{PRIMPATH}(p), plm, layout)$  to  $items$ 
15:  for all identifier  $d$  in  $plm \cup layout$  usable as an exact key do
16:     $exact[\text{NORMALIZE}(d)] \leftarrow i$ 
17:  end for
18:  for all alphanumeric token  $t$  in the textual fields of  $plm \cup layout$  do
19:    add  $i$  to  $tokens[\text{NORMALIZE}(t)]$ 
20:  end for
21: end for
22:  $I \leftarrow (items, exact, tokens)$ 
23:  $\text{SERIALIZE}(I, h)$  ▷ reused when the stage is reopened
24: return  $I$ 

```

Both metadata pathways are implemented because the prototype was tested during development on scenes from different facilities, with different metadata coverage. The PS Booster model used in the user study only carries PLM metadata, but other facility models encountered during development carried both families fully populated. Supporting both families from the outset meant that no code changes were required when switching between scenes, and the system is ready for facilities where Layout metadata becomes the primary lookup path.

5.3 Resolving Natural Language to Prim Paths

Once the index exists, the next problem is resolution. A user request such as “move to the flow meter in the straight section of period 13” must be converted into one or more USD prim paths on which a scene operation can be performed. The resolver was developed iteratively, with the

current three-stage pipeline emerging from a sequence of corrections to specific failure modes that appeared during testing.

Turning a free-form request into specific objects is divided between the language model and a deterministic resolver. The model reads the user’s message, decides that it calls for a search, and invokes the resolver with a query. The resolver matches that query against the index and returns the objects it finds. This section describes the resolver, the deterministic half of that exchange, whose three-stage structure and result categories are summarized in Fig. 5.2. How a message becomes a resolver call in the first place is covered later, in section 5.5. Keeping the matching deterministic rather than handing it to the model is a deliberate choice, making the same query yield the same result regardless of which model is in use.

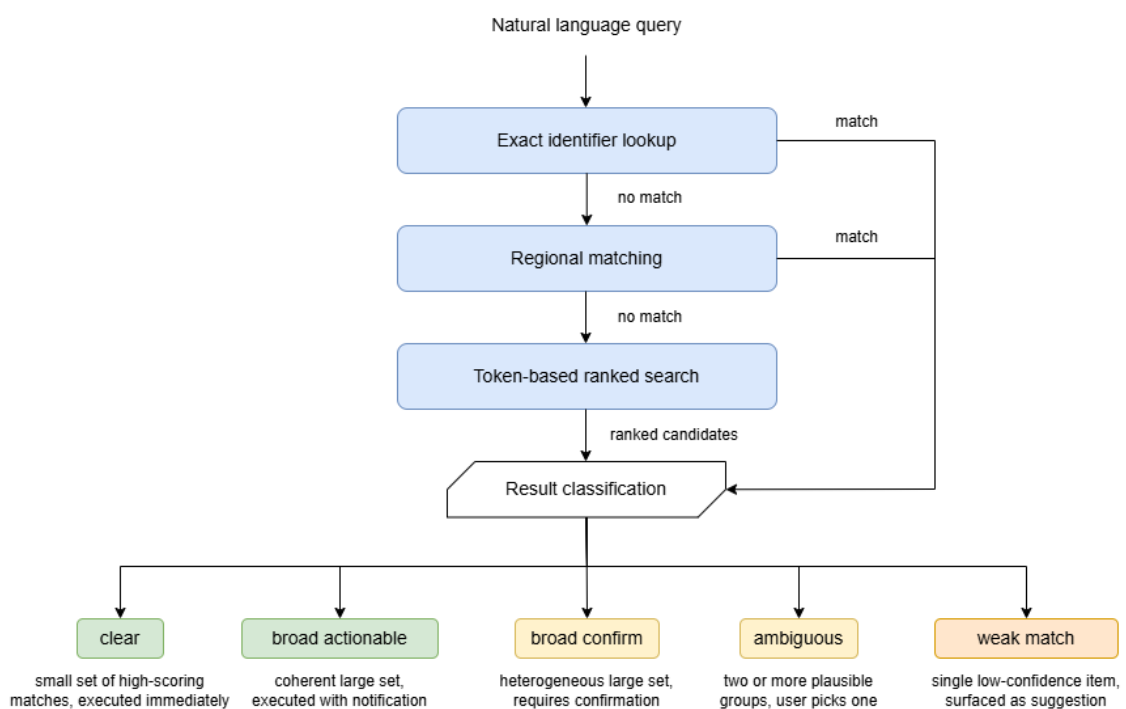


Figure 5.2: The three-stage resolution pipeline and the five-category result classification.

The first stage handles exact identifier lookups. When the normalized query string matches an entry in any of the exact dictionaries (a document number, a PLM identifier, a layout name), the matching items are returned immediately. This stage handles requests from users who already know an identifier, such as “ST0428283_02”, and is the cheapest path through the resolver.

The second stage handles regional queries, where the user asks for a whole area of the machine rather than a specific component. It was added after the token-based search produced a recurring class of false positives: region codes are short, structured identifiers, and tokenizing them yields tokens that unrelated codes can share, so a query for one area would match items belonging to an unrelated area whose code happened to contain one of the same tokens. Ranked relevance scoring is in any case the wrong tool here, because region codes are inherently broad and a user asking for an area wants everything inside it rather than the best-scoring few matches. Recognizing that

a request is region-scoped is left to the language model, which is prompted to treat a range of location expressions, such as regions, cells, sides, and named areas, as region references and to preserve the region text rather than searching for it as a component. The resolver then bypasses the ranked search and matches that region text by strict string equality against the layout and region metadata fields, returning every component within the identified area.

The third stage is the token-based ranked search. Each query token is normalized, expanded with singular/plural variants and fuzzy corrections, and matched against the inverted index. A scoring function then weights each candidate, with identifier-like fields (document number, PLM identifier, layout document number) carrying higher weights than descriptive fields (definitions, descriptions, region full names), since a match on a structured identifier is a stronger signal of intent than a match on free text. On top of this base score, a multi-token relevance multiplier became necessary after observing that candidates matching only one token of a multi-token query were being ranked above genuinely relevant ones. These multipliers are heuristic weights rather than learned parameters, and what carries the rationale is their relative ordering rather than their exact magnitude. Candidates whose query tokens all co-occur within a single descriptive field receive the largest bonus ($3\times$), because co-occurrence in one field is the strongest evidence that the tokens describe a single component. Candidates whose tokens span several descriptive fields receive a smaller bonus ($1.5\times$), reflecting a legitimate but weaker match. Candidates whose tokens are split between a descriptive and an identifier field are rejected, since identifier fields are structured codes where a token match is almost always coincidental rather than semantic. Candidates matching fewer than two-thirds of the query tokens are penalized by a $0.15\times$ factor, chosen to be aggressive enough to push partial matches below complete ones while still preserving them as low-ranked fallbacks. The magnitudes themselves were calibrated against the rankings expected on representative CERN queries, and fixed once the ordering matched expert expectations.

Even with this scoring in place, the resolver had to handle a different problem: deciding whether a result was authoritative enough to act on immediately or uncertain enough to require user confirmation. Acting on every search result without inspection was not acceptable, because a query such as “show me the magnets” returns hundreds of items and the user almost certainly does not want all of them displayed at once. Asking for confirmation on every query was equally not acceptable, because it adds friction to the cases where the result is unambiguous. The solution was to classify each result into one of five categories: clear (a small set of high-scoring matches, executed immediately); broad actionable (a coherent and homogeneous large set, executed immediately with a notification); broad confirm (a large but heterogeneous set, requiring confirmation before execution); ambiguous (two or more plausible but distinct match groups, requiring the user to pick one); and weak match (a single low-confidence item, surfaced as a suggestion rather than executed).

These categories are not arbitrary labels but the outcome of three measurable properties of the result set, evaluated in a fixed order: its size, its coherence, and the score separation between its leading interpretations. Results are first grouped by label, where a group collects all items sharing the same component definition. A group therefore represents one kind of component, and each

carries a member count and a top score. From these groups three quantities are derived. A result is treated as *large* when it exceeds any of three limits, namely more than 25 total items, a single group of more than 10, or more than 6 distinct groups, since any of these makes silent execution on the full set undesirable. A large result is treated as *coherent* when one group dominates it, defined as a single group, or a top group holding at least 75% of all results (and at least ten items) with no more than three groups present; otherwise it is *mixed*. Finally, two interpretations are treated as *competing* when more than one group exists and the score gap between the top two is small (≤ 1.2), indicating that the resolver cannot confidently prefer one reading over another. The decision procedure combining these quantities is summarized in Algorithm 2.

Algorithm 2 Result classification

```

1:  $large \leftarrow (n > 25 \vee n_{top} > 10 \vee g > 6)$ 
2:  $coherent \leftarrow (g = 1 \vee (n_{top} \geq \max(10, 0.75n) \wedge g \leq 3))$ 
3: if large then
4:   if coherent and intent is low-risk and  $n \leq 300$  then
5:     return broad actionable ▷ execute with a notification
6:   else
7:     return broad confirm ▷ hold for confirmation
8:   end if
9: else if  $g > 1$  and  $s_1 - s_2 \leq 1.2$  then
10:  return ambiguous ▷ user selects an interpretation
11: else if  $s_1 < 4.5$  then
12:  return weak match ▷ surface as a suggestion
13: else
14:  return clear ▷ execute immediately
15: end if

```

The distinction between *clear* and *broad actionable* is deliberate: both execute automatically, but the former acts silently while the latter always reports the number of affected items, so that a bulk operation is never performed without the user’s awareness. The procedure also defaults to the safer side at its boundaries, taking the confirmation path for large results unless they are both coherent and within a bounded size, and gating automatic execution of large coherent sets by intent so that it applies only to low-risk operations such as selection, highlighting, and visibility changes. The specific thresholds were calibrated on representative queries over the PS Booster model, so while the decision structure is general, the constants may require adjustment for a facility with different naming conventions or component density.

An important property of this classification is that it is computed deterministically by the resolver itself, before the result reaches the language model. An alternative would have been to return the raw result set to the model and to let it decide, in natural language, whether the result is actionable or needs confirmation. This was rejected for two reasons. First, the classification depends on quantitative properties of the result (size of the set, homogeneity of the matches, score gap between the top result and the next) that the model can only inspect indirectly through a textual summary, and the resulting decision would be both slower and less consistent. Second, a

deterministic classifier produces the same decision for the same input regardless of which language model is in use, keeping this behavior stable and independent of the model. This stability is also what makes the interface predictable to the user, since the same request maps to the same action whatever wording the model returns. For a conversational interface, that predictability matters as much as the breadth of language accepted, because a user who cannot anticipate the response has no basis for trusting it. The classification is consumed by the conversational modifier and by the manual interface to decide between immediate execution, warning, and clarification.

A further extension supports nested scope constraints. Maintenance planning queries frequently combine several layers of location scope, as in “the supporting tube inside multipole in period 3”, where a *period* denotes a spatial division of the PS Booster ring. Rather than interpreting the natural-language structure with the language model, scope detection is deliberately rule-based: the resolver applies a fixed regular expression that splits a query at an explicit, closed set of prepositional connectors, namely *in*, *within*, *inside*, *under*, *of*, *from*, and *part of* (each also matched in its “... the” form, as in “inside the”). The text before the connector becomes the target and the text after it becomes the enclosing scope.

When a split is found, it is applied recursively. The innermost scope is resolved first (period 3), the next level is filtered to the descendants of that result (multipoles inside period 3), and finally the target is searched within the filtered set (supporting tubes inside the qualifying multipoles), where descendant filtering is a strict prim-path containment test. The same mechanism composes with task constraints: “the supporting tube with a dismantling task inside multipole in period 3” applies the task filter as a final reduction step after the spatial scope has been established.

Although described above in operational terms, the operations the resolver composes can be understood more generally as set operations over the space of scene components. A scope constraint intersects the candidate set with the descendants of a region; a task constraint intersects it further with the components that carry a matching task; and disambiguation selects one subset among several. A second property is that this composition spans two representations: because task assignments are retrieved from the external authoring service and intersected with the geometric result at query time, a single request can reach into both the geometric model and the external task records, which neither could satisfy alone. This enables maintenance-oriented selections such as “select the supporting tubes in period 3 that have a dismantling task”, where the geometry locates the components and the task data filters them.

Framing selection as set composition also clarifies what the current resolver does and does not yet support. Consider a more demanding query such as “select the magnets in octant 3 that were last intervened on under two years ago and have no maintenance task scheduled next month”. Decomposed into set operations, this is the intersection of a component type (magnets) with a spatial scope (octant 3), further intersected with a temporal predicate on intervention history, and finally a negation that removes components with a task in a given window. The current resolver handles the type-and-scope intersection and the intersection with task presence directly. The temporal predicates and the set-negation are not implemented, but they extend the same compositional model rather than requiring a different one, since each is an additional filter applied to the candidate set

rather than a new kind of operation. Selection was therefore designed as a composition of set operations precisely so that richer forms, such as negation, temporal filters, or unions of heterogeneous criteria, can be added as further filters without redesigning the resolution pipeline.

Because detection relies on a fixed connector list rather than general parsing, the mechanism is fully predictable but bounded by that list, handling the prepositional and period forms enumerated above, but not interpreting scope expressed through connectors outside the set or through less explicit phrasing. This is a deliberate trade-off, favoring deterministic, inspectable behavior over the broader but less predictable coverage a model-based parser would provide.

5.4 Spatial Reasoning

Spatial proximity queries are not part of the metadata search problem and required a separate service. The first implementation tested intersection of axis-aligned bounding boxes (AABBs), computed from the world transformation of each candidate. AABBs are inexpensive to compute and to test for intersection, but the results were unsatisfactory on CERN tunnel models. Equipment in the tunnels is frequently installed at angles relative to the world coordinate axes. A long magnet mounted diagonally has an AABB that is substantially larger than the actual component body, so proximity tests produced numerous false positives. A query for objects connected to a diagonal magnet returned a long list of components that were near the AABB but not near the magnet itself.

The natural alternative to AABBs for rotated geometry is an oriented bounding box, which aligns the bounding volume to the object's local axes rather than the world axes and is therefore tight regardless of orientation, as illustrated in Fig. 5.3. Other options were considered and rejected. Bounding spheres would be even looser than AABBs for the long components that dominate a tunnel scene, such as magnets and beam pipes. Convex hulls would be more accurate but require access to the full mesh geometry at query time, which is not guaranteed for simplified or instanced CAD imports. Reducing the AABB overlap tolerance does not address the root cause, since the inflation of an AABB for a rotated object scales with the object's dimensions and no fixed tolerance compensates for it correctly across components of different sizes. OBBs are a good fit for this context because the transformation matrix that USD already stores for every prim is sufficient to derive the oriented box. The three axes come from the matrix columns and the half-extents from the local-space bounding range scaled by the column magnitudes, with no additional geometry processing required. Because this derivation depends only on the prim's transform and its local bounding range, it is independent of the component's shape: an elongated magnet, an L-shaped support, or any other awkwardly proportioned part yields a box correctly aligned to its own orientation, without any special handling per shape. The remaining approximation is the familiar one for box-based bounds, in that a strongly non-convex part, such as an L-shaped bracket, still has its concave region enclosed by the oriented box. Intersection between two OBBs is tested using the Separating Axis Theorem (SAT), which evaluates fifteen potential separating axes and is exact for convex polytopes.

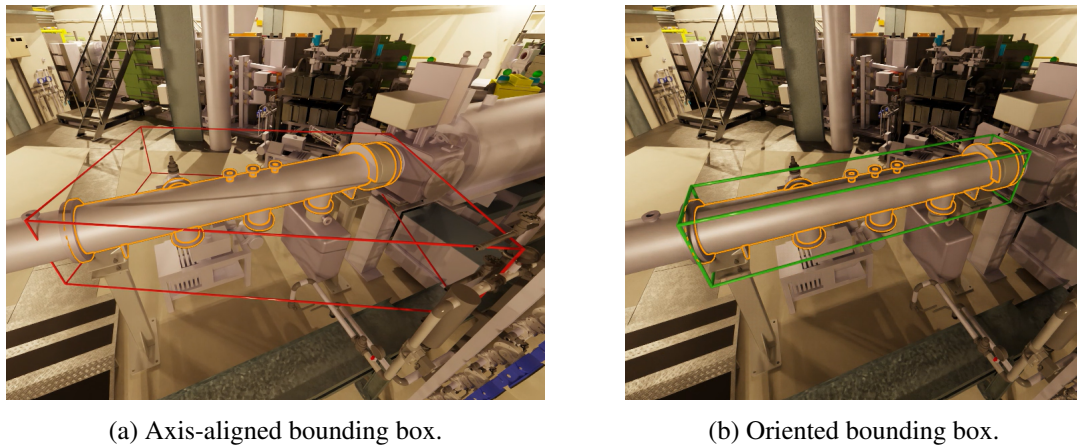


Figure 5.3: Comparison between an axis-aligned bounding box and an oriented bounding box around the same elongated component. The AABB inflates around the rotated component and can include nearby geometry that is not actually adjacent, whereas the OBB follows the component orientation and provides a tighter spatial filter for proximity queries.

Two further heuristics discard candidates whose bounding volume completely encloses the reference object, since these are infrastructure elements such as tunnel walls rather than adjacent equipment, and candidates whose volume is more than an order of magnitude larger than the reference. The latter is a coarse guard against a few oversized enclosing volumes that survive the first check, rather than a tuned parameter.

A second class of false positives appeared after the OBB implementation was in place, but this time related to the granularity of the index. The index represents objects at the assembly level, and assemblies have bounding volumes that cover all of their sub-components. When a query asked for objects connected to a specific object, the OBB test would return entire neighboring assemblies whose bounds happened to overlap with it, even when only one or two of their sub-components were physically touching it. The user interpretation of “connected to” is at the level of individual contact rather than overlap of containing volumes, so this had to be corrected. The correction adopted is a result demotion mechanism. When a candidate’s OBB passes the intersection test but is classified as geometrically loose, because its geometry fills less than 30% of the bounding box volume, the spatial service descends into the candidate’s children and tests each child’s leaf mesh OBBs against the reference geometry. As illustrated in Fig. 5.4, the assembly OBB is used as a coarse pre-filter, while the leaf mesh OBBs act as a precision filter. Only children with at least one leaf mesh in actual contact are included in the result, correcting the false positives at assembly level while preserving the granularity that the rest of the system expects. More generally, a spatial query is only useful when its result matches the relation the user has in mind, so where the boundary of a relation such as “connected” falls is a matter of meaning as much as of geometry.

A related problem appears when the user clicks deep inside a hierarchy and selects a single mesh fragment rather than a complete component. A spatial query started from such a fragment has no meaningful reference object: the fragment is an arbitrary slice of geometry that does not correspond to anything an engineer would think of as a part, and proximity to it is not what the

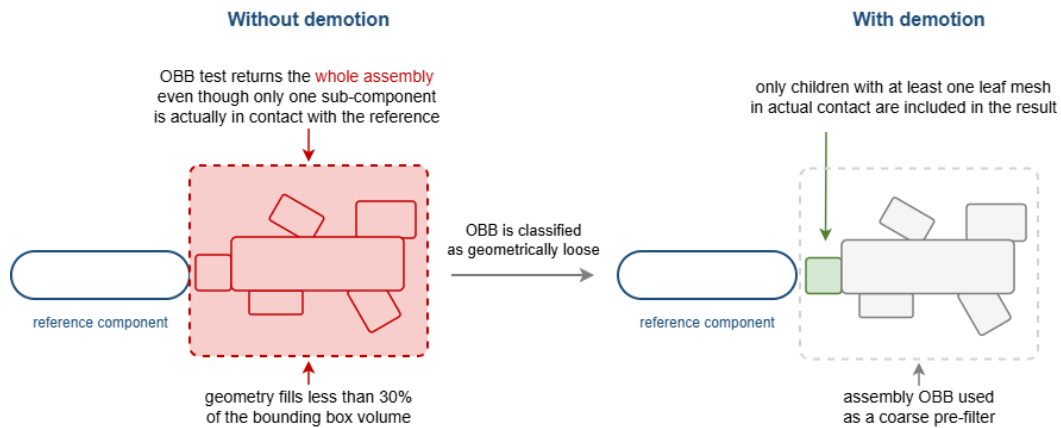


Figure 5.4: Result demotion: the assembly-level OBB is used as a coarse filter, and only those leaf-mesh children in actual contact are retained.

user intends. The service therefore promotes the selection upwards before performing the query, walking up the hierarchy from the clicked mesh until it reaches an ancestor that represents a complete component, and uses that ancestor as the reference object. The promotion is transparent to the user: the same query produces the same result whether the user clicks the parent assembly directly or a meaningless mesh deep inside it.

5.5 The Conversational Control Layer

The conversational interface accepts a natural-language request and translates it into one or more backend operations. The implementation challenge is that the language model produces output that is probabilistic and variable in structure, while the backend requires well-typed parameters and a definite operation name. The conversational control layer was developed iteratively, starting from a minimal structured-tool integration and acquiring additional defensive logic in response to specific failure modes that appeared during testing.

The starting point was a one-shot structured-tool integration. The backend operations are exposed to the language model through a JSON specification file describing each operation's name, description and typed parameter schema. At runtime, the system reads this specification and dynamically generates LangChain¹ StructuredTool objects with Pydantic² parameter validation. Defining the tools declaratively rather than as hand-written Python decorators was a deliberate choice: the same JSON specification drives the runtime tool binding, the textual tool list embedded in the system prompt, and the parameter validation, so adding or changing an operation is a single edit to one file. This decoupling of the operation set from the model integration also made the model migration described later in this chapter substantially less invasive, because the tool definitions did not have to change. The model is bound to these tools once per session, and at every user turn it receives the user's message together with the current viewport selection and emits

¹LangChain framework repository. Available at: <https://github.com/langchain-ai/langchain>

²Pydantic data validation library. Available at: <https://docs.pydantic.dev>

a structured tool call which is dispatched directly to the backend. This integration alone was not sufficient, for several reasons that emerged during development.

The first problem was that the model occasionally passed structurally incorrect parameter values: a dictionary where a string was expected, a list where a numeric tolerance was expected, or a nested resolver result where a prim path was expected. These were not failures of language understanding but of structural compliance, and they could be corrected without involving the model again. An argument sanitization pass was added between the model output and the backend dispatch, normalizing parameter values and falling back to defaults when the input could not be parsed. This pass is invisible to the user but eliminated a recurring class of dispatch errors.

The second problem was latency on commands that need no interpretation. The model is the normal path for a user message, but some messages are fully determined by their verb and a selection pronoun, and routing them through the model only adds latency, since it sometimes produced a multi-sentence reply or a clarification request where an immediate action was expected. For this narrow class of messages, a fast path intercepts the message before the model is consulted. It tokenizes the message and checks that every token is either a recognized action verb or a filler word. The recognized verbs are a fixed set covering selection (`select`, `highlight`, `pick`, `choose`) and visibility or navigation actions (`hide`, `show`, `isolate`, `focus`, `move`, `navigate`, `only`, `except`, `visible`, `invisible`); the fillers are prepositions and the selection pronouns `this`, `it`, `that`, `here`, and `there`. A message that passes this check and contains at least one selection pronoun is dispatched directly to the corresponding action on the current selection. Anything that also contains content words, such as the name of a component, fails the check and is left to the model. A few bare commands, namely `undo`, `redo`, `reset`, and `reset view`, are handled in the same way.

The third problem was redundant LLM round-trips. The most common pattern in the conversational interface is resolve-then-act: the user names a component, the model emits a resolver call, the backend returns the resolved prim paths, and a second LLM round-trip is used to dispatch the corresponding scene operation. The second round-trip is wasteful in this pattern because the user's intended action is already fully expressed by the verb in the original message. The choice made was to extend the resolver's signature so that the model can express both the search target and the intended downstream action in a single tool call. When the model does this, the action is applied to the resolved object as soon as the resolver returns. Bundling intent into the resolver call rather than splitting it across two calls covers the majority of navigation, search and isolation requests in a single LLM call.

The fourth problem was multi-step requests where the output of one tool call is needed as input to the next. A request such as "select all components connected to the vacuum chamber in the ejection line" requires three sequential operations: resolving the reference object, querying its spatial neighbors, and selecting the result. The single-shot integration could not handle these requests because the model would either collapse the request into one tool call (selecting the reference object directly rather than its neighbors) or produce only the first tool call with no follow-up. An agent loop was added that addresses both behaviors: after each tool dispatch, the result is

summarized and fed back to the model with a continuation prompt. The model either calls another tool or signals completion by producing a textual reply. The loop is capped at six tool calls per user turn: enough for the multi-step compositions the interface supports (which chain a resolution, a spatial query, and a selection) with headroom for the model to recover from an unproductive intermediate call, while still placing a hard limit on worst-case latency. The specific bound is a conservative guard rather than a tuned value.

The agent loop made multi-step requests possible, but it could not prevent the model from collapsing them. For a relational query the model still strongly preferred to answer in a single call: given a message with a spatial-relation phrase such as “connected to”, it would bundle the action into the resolver call, just as it does for a simple resolve-then-act request, so the reference object itself was selected instead of its neighbors and the spatial query was never run. A guard was added for exactly this case. When the message contains a spatial-relation phrase and the model bundles an action into the resolver call, that action is removed before the call is dispatched. With the action removed, the resolver call only resolves the reference object, and the request continues through the agent loop.

The resulting modifier follows the dispatch hierarchy in Fig. 5.5. A message that is fully determined by a verb and a selection pronoun, or a bare command such as undo or reset, is handled immediately, before the model is consulted. When the model identifies a single target and a clear action, that action is carried out as soon as the search returns, without a second model call. When the request is relational, the action is dropped so that it is forced through the agent loop rather than answered in one step. Any request that genuinely needs several steps is handled by the loop, one tool call at a time. Beyond this routing, the modifier also detects pushback: when the user signals that the last action was wrong, it undoes the action and asks for clarification.

Table 5.1: Example message handled at each level of the dispatch hierarchy.

Level	Example message	Model consulted?
Fast path (verb + selection pronoun)	“isolate it”, “move here”	No
Bare command	“undo”, “reset”	No
Single resolve-then-act	“move to the flow meter in period 13”	Once
Relational	“select the elements connected to this”	Loop
Multi-step composition	“select all components connected to the vacuum chamber in the ejection line”	Loop

A secondary intent classifier is registered as a last-resort fallback for the rare cases where the model fails to produce any structured tool call at all. This is an explicit design choice: a deployed system that depends on a remote language model cannot tolerate a complete failure mode, and a deterministic backup interpretation path ensures that the user receives some response, even if degraded, when the primary path produces nothing usable. The classifier handles a small set of canonical phrasings and is intentionally narrow, since it exists to recover from failure rather

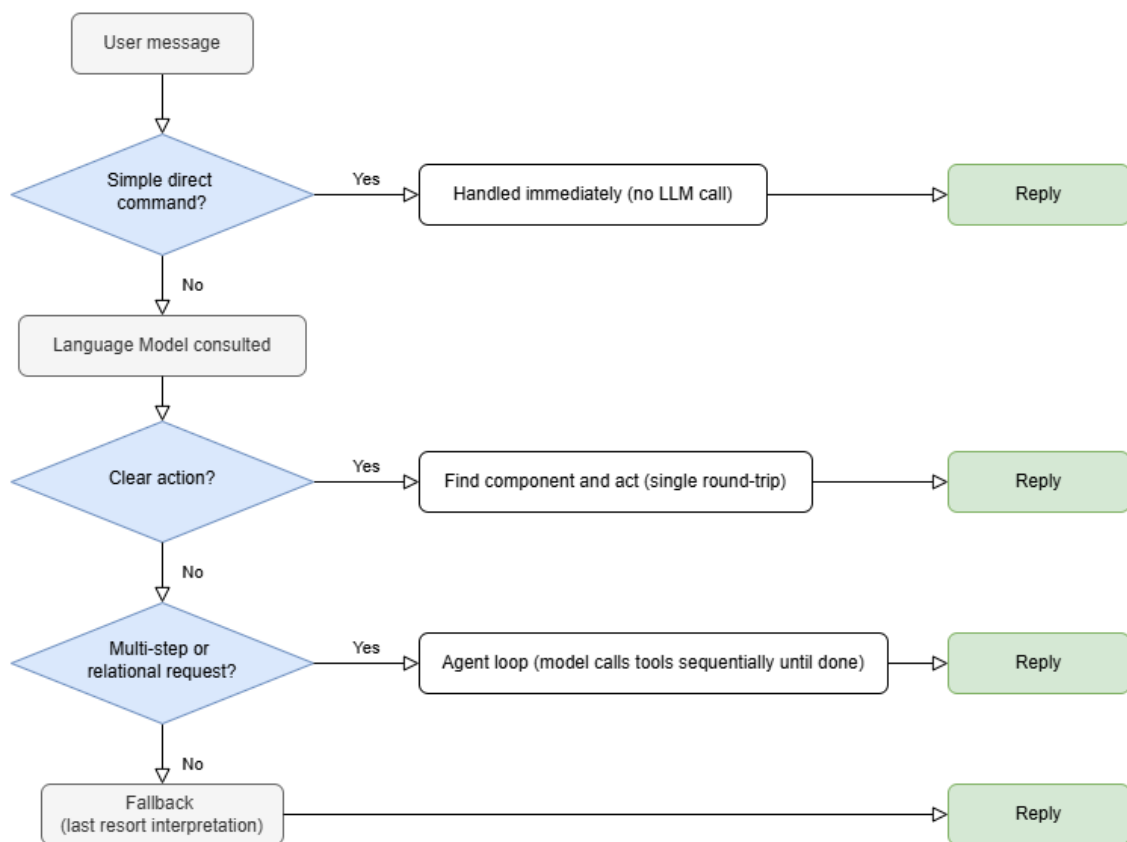


Figure 5.5: The dispatch hierarchy applied by the modifier between the language model and the backend.

than to compete with the LLM path. Only the interpretation of a free-form request depends on the language model. Resolution, dispatch, and the five-category classifier are deterministic, so a change of model affects which interpretation is tried first but not the set of operations the system can produce.

One capability follows from these mechanisms together. Because the current viewport selection is part of the model’s context, a deictic word such as “this” in a message routed to the model resolves to the currently selected component, so the user can refer to the anchor of a relation by pointing at it rather than naming it. The relational query is then handled as before, through the agent loop, with the selection serving as its reference object. Figure 5.6 shows such a query.

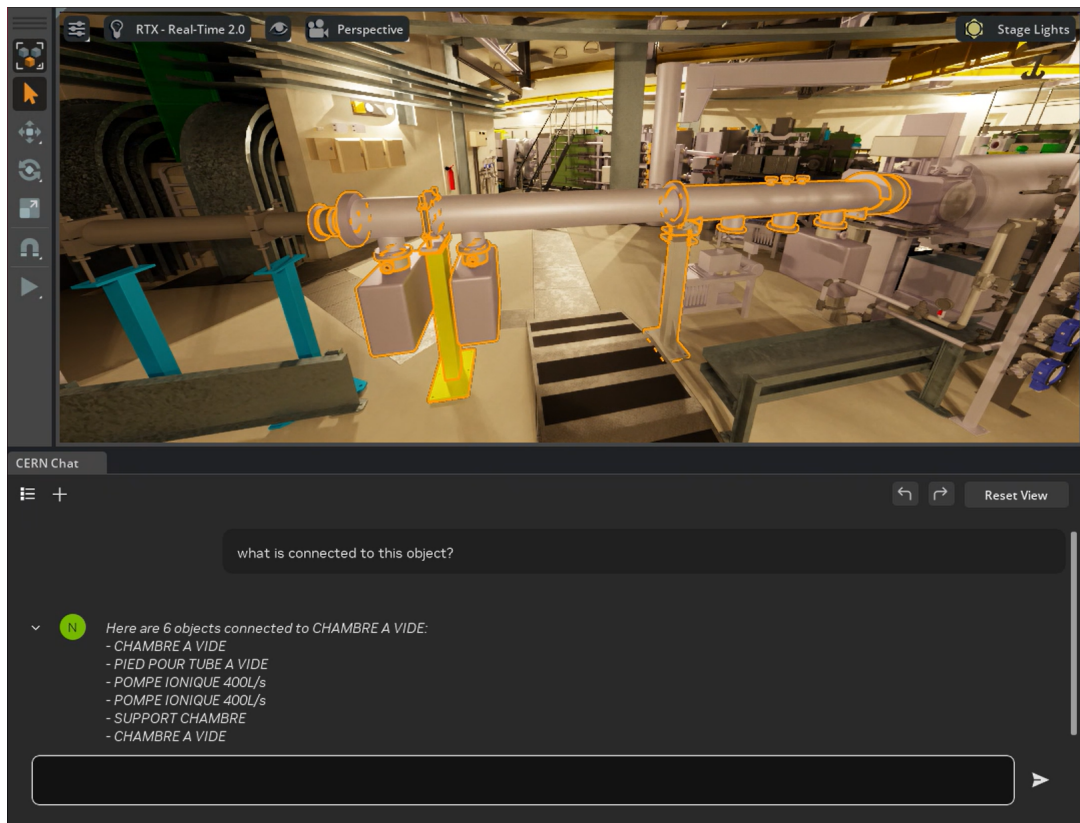


Figure 5.6: A connectivity query whose reference is the current selection. The vacuum chamber (CHAMBRE A VIDE) was selected in the viewport before the message was sent, so the deictic “this object” is bound to that selection. The agent’s reply names CHAMBRE A VIDE explicitly and lists the six connected components, which are outlined in orange in the viewport.

5.6 The Manual Interface

The two interface windows added by this work are the Info Window and the CAD Tree. The Info Window is present in both evaluation configurations, while the CAD Tree is part of the hybrid configuration only.

5.6.1 The Info Window

The Info Window surfaces the engineering metadata and any associated maintenance tasks for the currently selected component. It is rebuilt on every selection change and stays synchronized with the viewport.

A specific design decision was made for how the window resolves the component to display. The first version of the window read metadata directly from whichever prim was selected in the viewport. This produced empty or near-empty results for the most common case: when the user clicks any object in the viewport, what they actually click is a meaningless leaf mesh deep inside an assembly, not the assembly itself. Leaf meshes typically carry no metadata in CERN scenes, since the metadata is authored on the parent assembly. The Info Window appeared empty for what looked, from the user's point of view, like a successfully selected component. To correct this, the window was changed to walk up the hierarchy from the clicked prim until it finds the nearest ancestor that carries meaningful data. The displayed metadata is therefore that of the engineering-level component the user clicked on, rather than the geometric fragment they happened to hit.

The resulting window layout, shown in Fig. 5.7, includes an assembly card with the document number, version, PLM name and a task section listing any assignments retrieved from the digital twin authoring layer. The task section is the only part of the window that affects the viewport beyond simple selection feedback. When the user hovers over a task row, the corresponding component receives a color overlay: a colored outline together with a translucent fill in the same hue. In the current prototype the color is determined by the type of task, red for dismantling, yellow for maintenance, and green for installation, a practical choice for the user-study tasks. The intended direction for a production system is to assign colors by the team responsible for the task. Clicking a task row pins the overlay to the component until the user selects a different object, at which point the overlay is cleared automatically.

5.6.2 The CAD Tree

The CAD Tree exposes the scene hierarchy in a side panel and is the interface element that addresses the vocabulary gap between the language users speak and the technical identifiers that USD prim paths expose. The default Omniverse scene tree shows prims under their internal USD names, which are derived from the CAD export and bear no resemblance to the way a part is described in practice (Fig. 5.8). These names cannot be searched against in any meaningful way by a user looking for “the wire scanner in period 4”. Beyond the names themselves, the default tree also exposes the geometry meshes, material definitions, and repeated instances that make up each assembly, none of which corresponds to a part a user would search for.

A specific naming choice was made for the tree. Rather than displaying prim names, document numbers, or any of the other identifier-style attributes available on the indexed items, the tree displays each item's PLM description. This is a free-text attribute authored by engineers in PLM and contains the natural-language name that domain users actually use when referring to a component. The decision was practical rather than algorithmic: there was no need to add a fallback or a

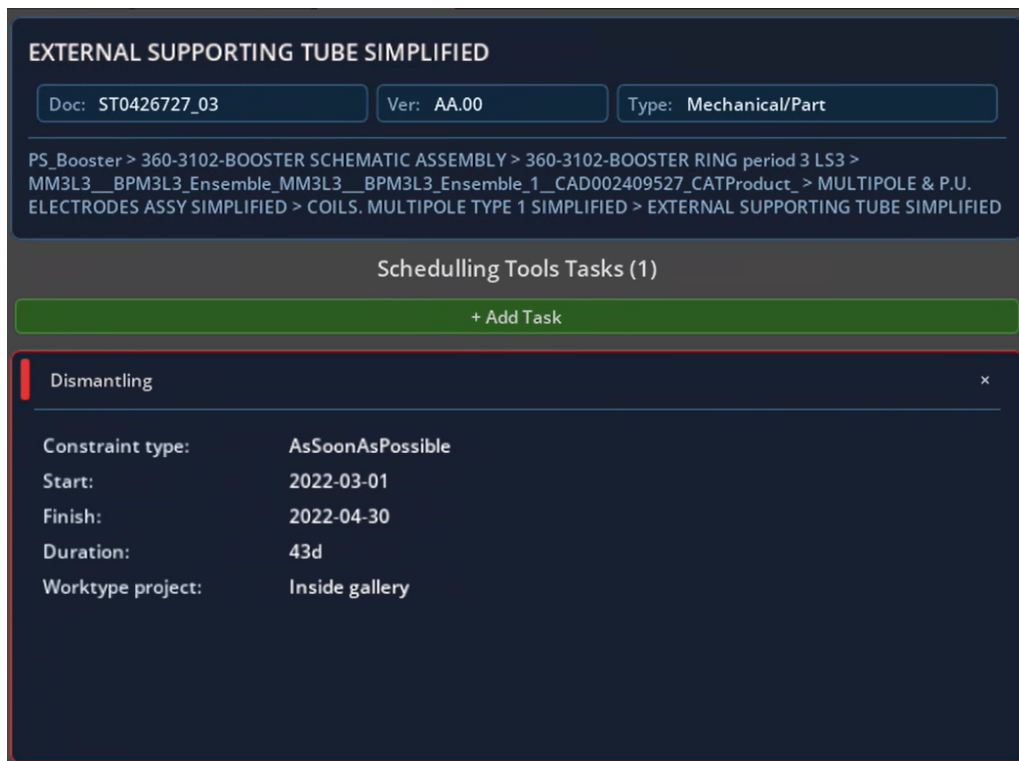


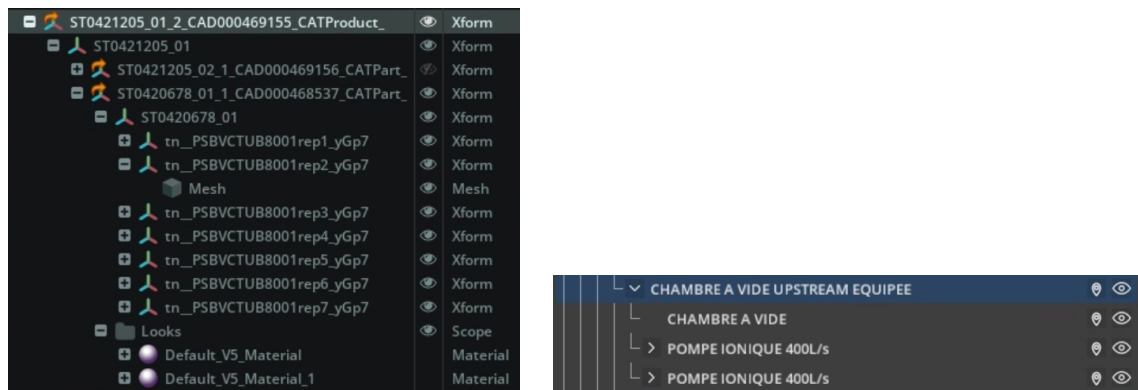
Figure 5.7: The Info Window for a selected component, showing the assembly card, scheduled tasks, and engineering metadata.

renaming layer because the scope of the tree is the same as the scope of the index, and every item in the index carries a PLM description by construction. Items without PLM data do not appear in the index, and therefore do not appear in the tree, and the tree can simply use the description directly.

Each node in the tree carries two icon buttons: move to, which centers the camera on the component and hide, which removes the component from the visible set. The tree is connected to the resolution pipeline, so when a query is resolved through the conversational agent, the tree updates to display only the matching nodes in their hierarchical context. This connection allows the user to see where a found component sits in the assembly structure rather than only as an isolated viewport selection.

5.7 External Data Integration

Beyond the geometry and the metadata embedded in the scene, the prototype needs to bring in information that lives in systems outside the 3D model and that changes independently of it. Rather than binding the tool directly to any one such system, external data is integrated through a general pattern: the tool reads from a validated intermediate layer rather than from a live source database, so that the source of truth is never written to directly, a domain expert can review data before it becomes visible in the tool, and the specifics of any one external system are isolated behind a stable



Default Omniverse stage tree

The CAD Tree

Figure 5.8: The same component shown in the default Omniverse stage tree (left) and in the CAD Tree (right). The default tree lists raw USD names alongside geometry and material nodes, whereas the CAD Tree presents each component under its PLM description and adds per-node action buttons.

interface. This keeps the ingestion mechanism independent of which external system supplies the data, so that additional sources can be integrated by adding another validated layer rather than by changing how the tool consumes external information.

Task scheduling data is the instantiation of this pattern realized in the prototype, and serves as a proof of concept for it. Tasks are retrieved from the digital-twin authoring layer, an intermediate REST API through which task-to-CAD assignments are reviewed and validated before reaching the prototype. This layer exists precisely so that the 3D tool does not write directly to the live scheduling database, and so that a domain expert can validate each assignment before it becomes visible. The prototype queries this validated layer and surfaces any associated tasks in the Info Window when a component is selected. A local name cache ensures task names remain displayable across sessions even when the upstream API cannot return a record for a previously assigned task identifier, prioritizing display consistency over immediate freshness for the name field specifically.

The scene operations themselves (camera navigation, component selection, visibility isolation) are implemented as edits to the USD session layer, leaving the base stage untouched. This was done for two reasons. First, every operation is reversible at the layer level, so the system needs no separate undo code for each one. Second, no operation can corrupt the upstream USD file, which is shared with other CERN tools and not under the prototype's exclusive control. A visibility undo stack sits on top of the session-layer edits and is exposed through a reset command, letting participants in the user study recover from accidental changes without navigating menus.

5.8 The Assembled Interface

With the conversational and manual components described, it is worth showing how they come together at run time. The two interfaces are not separate tools: they operate over the same backend

services and share a single selection, so a component found through a natural-language query becomes the selection that the Info Window describes and the CAD Tree highlights, and a component picked manually in the tree or the viewport can immediately become the subject of a follow-up conversational command. This shared state is what makes the combined environment a single hybrid interface rather than two interfaces running side by side, and it is what the central claim of this work rests on: the hypothesis is not that conversation replaces manual interaction but that the two are stronger in combination, and that combination exists only because both act on the same state.

Figure 5.9 shows the prototype running on the PS Booster model, with all of these components in a single workspace. The 3D viewport fills the main area on the left. On the right, the CAD Tree is docked above and the Info Window below it, so that the scene hierarchy and the metadata of the selected component are both visible alongside the view. The conversational chat panel is docked along the bottom, beneath the viewport. The figure also captures a complete interaction: the typed request resolves to a single component, which is outlined in the viewport, selected in the tree, and shown with its scheduled dismantling task in the Info Window.

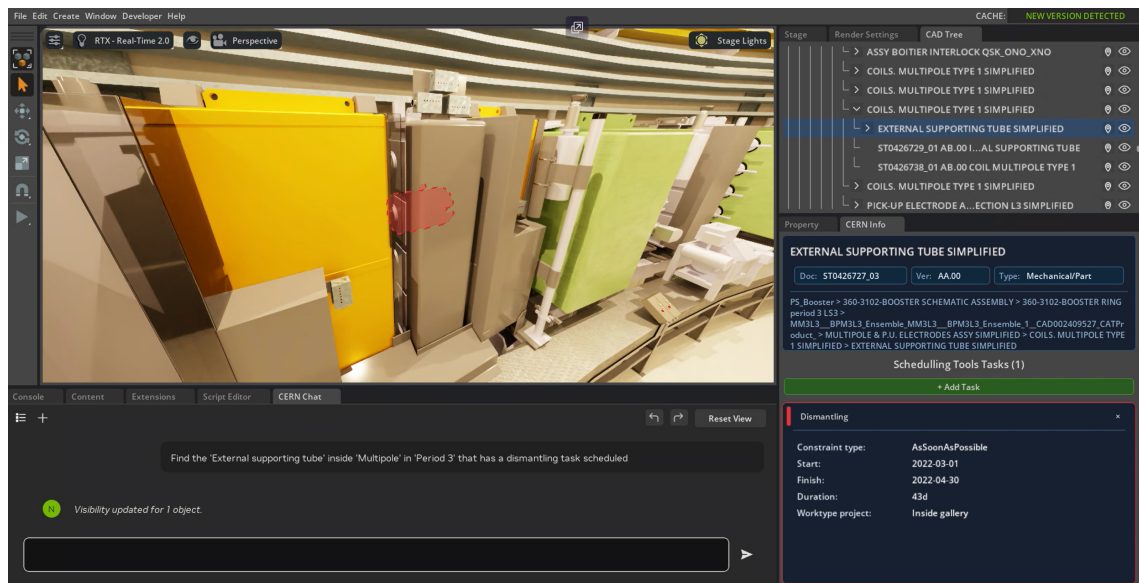


Figure 5.9: The prototype running on the PS Booster, with the viewport (left), the CAD Tree and Info Window (right), and the conversational chat panel (bottom). The chat shows a natural-language query resolved to a single component, which is highlighted in the viewport, selected in the tree, and shown with its scheduled task in the Info Window.

5.9 Development Challenges

The most disruptive challenge during the development of the prototype was a change of language model midway through development. The prototype was initially developed against a model that the ChatUSD bundle exposed in the local development environment. When the prototype was deployed to the streaming server used for the user study, that environment was running a newer

version of the Omniverse application, which had a different set of registered models. The original model was no longer available, and the closest replacement, NVIDIA Llama 3.3 Nemotron Super 49B³, had substantially different tool-binding characteristics: a different output format for tool calls, a tendency to produce reasoning text alongside the structured output, and different sensitivity to system-prompt phrasing.

The migration is also the clearest evidence for the architectural choices described in Section 5.5. Because the operation set is declared once in a JSON specification and bound to the model at runtime, and because the resolver and its five-category classification are computed deterministically in the backend, the parts of the system that define what it can do were untouched by the change of model. What had to be re-tuned was the system prompt and the modifier layer, which is the surface where the system meets the model. A different model changes which interpretation is attempted first and how a tool call is phrased, but it does not change the set of operations available, the way a name is resolved to a component, or the decision of whether a result is acted upon or held for confirmation. Had the operation set been written against the calling convention of one particular model, or had result classification been delegated to the model, the same change would have required reworking the behavior of the system rather than the wording of its prompt.

The second challenge was the multilingual authoring of PLM descriptions. The PS Booster model was assembled from CAD designs authored over several decades, during which no consistent naming convention was enforced. Component names appear in English, French and German depending on when the design was created and who authored it. This means that even within a single metadata family, text-based retrieval cannot rely on a consistent vocabulary, and a search for one phrasing of a component name will not find instances recorded under a different language. The mitigation adopted was to keep the token-based search granular: by matching individual tokens rather than full strings, a query in one language can still reach items whose description contains a subset of the same tokens, recovering many of the queries that a string-level search would have missed. The underlying limitation remains, in that a query whose tokens are entirely in one language will not retrieve a component whose description is entirely in another, and a future system would benefit from a translation or alignment layer over the PLM descriptions.

The third challenge concerned the cost of interactive use against a remote, hosted language model. The model runs on NVIDIA's shared hosted infrastructure, and the time taken by a single round-trip is variable: most queries return in a few seconds, but the latency distribution has a long tail in which a call can take considerably longer, presumably because of queuing on the shared endpoint. In the first implementation the model call ran on the same thread that drives the Omniverse interface, and while that thread was waiting for a slow call it could do nothing else, so the whole application appeared to freeze. The viewport stopped responding, the chat panel stopped accepting input, and the user was given no sign that anything was happening. In a small fraction of cases a call would stall indefinitely and the application had to be restarted. The fix was to move the model call onto a separate worker thread, so that the interface thread keeps running while the

³NVIDIA Llama 3.3 Nemotron Super 49B model card. Available at: https://build.nvidia.com/nvidia/llama-3_3-nemotron-super-49b-v1_5/modelcard

call is in progress, and to give the call a fixed deadline, so that a stalled round-trip is abandoned rather than waited on forever. After this change the application stayed responsive through any call, including the slow ones, and the indefinite-stall failure could no longer occur. This is a real constraint of building interactive applications on top of remote language-model services, and the pattern adopted here transfers directly to any similar deployment.

5.10 Summary

The prototype was built on the conversational agent included with Omniverse, redirected from creating geometry to finding components that already exist in the model. To make that search fast, a separate index is built from the loaded stage at startup and reused across sessions, holding only the objects that carry engineering metadata and ignoring the intermediate hierarchy that the CAD export produces. A three-stage resolution pipeline of exact identifier lookup, regional matching and token-based ranked search maps the language engineers use onto specific objects, and a five-category classification, computed deterministically by the resolver rather than by the model, decides whether each result is executed immediately, executed with a notification, or held for confirmation. Computing this classification deterministically is what kept the system's behavior stable when the underlying model had to be replaced partway through development. Spatial queries are handled by a separate service that moved from axis-aligned to oriented bounding boxes once the former proved too loose for the angled equipment in the tunnels, with a demotion step that descends into spread-out assemblies so that a relation such as "connected to" returns the sub-components actually in contact rather than whole neighboring assemblies, and a selection-promotion step so that a query started from a clicked mesh operates on the complete component it belongs to.

The conversational control layer connects this backend to the language model through tools defined once in a JSON specification and bound dynamically at runtime, with a modifier between the model and the backend to absorb the gap between variable model output and the typed operations the backend expects. The modifier grew in stages, each one a response to a failure observed during testing: argument sanitization for malformed parameters, a pronoun fast path for trivially interpretable commands, an auto-chain that bundles the intended action into the resolver call to avoid a second round-trip, an agent loop for multi-step requests, a guard that stops relational queries from collapsing into a single selection, and a fallback classifier that keeps the system responsive when the model produces nothing usable.

The Info Window and the CAD Tree are the visible components added by this work. The Info Window, present in both evaluation configurations, surfaces engineering metadata and scheduled tasks for the selected component, displaying the metadata of the engineering-level component the user clicked on rather than the leaf mesh fragment. The CAD Tree, available only in the hybrid configuration, exposes the scene hierarchy through the PLM descriptions engineers actually use instead of raw USD names, and reflects the current search state. Both share a single selection with the conversational agent, which is what makes the combined environment a single hybrid interface

rather than two tools side by side. The task data they surface reaches the prototype through a validated authoring layer rather than the live scheduling database, and every scene operation is written to the USD session layer so that it remains reversible and never modifies the base stage that other CERN tools share.

Two limitations are worth noting. The PLM descriptions mix English, French and German accumulated over decades of design; token-level matching lets a query in one language reach items described partly in another, but a query written entirely in one language will not retrieve a component described entirely in another, and a translation or alignment layer is left as future work. The hosted language model also has a variable response time with a pronounced long tail, which was addressed by running the call off the user-interface thread under a fixed timeout, so that the application stays responsive and a stalled call is abandoned rather than left to hang.

Chapter 6

Experimental Protocol

This chapter describes the experimental protocol used to evaluate the prototype. It begins by stating the evaluation objectives and the participants involved, then details the questionnaire design, the role of the moderator, and the apparatus used. It closes with the procedure followed in each session, the evaluation methodology, the pilot test that preceded the main study, and the threats to validity considered.

6.1 Evaluation Objectives

This chapter outlines the experimental protocol designed to evaluate the usability and effectiveness of the developed prototype for interacting with dense industrial 3D models in NVIDIA Omniverse. The primary objective was to assess whether users could locate, select, and inspect specific components within CERN’s complex engineering assemblies more effectively when assisted by a conversational interface combined with conventional desktop interaction, compared to using conventional interaction alone. The experiment involved CERN members engaging with the prototype to navigate dense 3D scenes, identify target components, and extract associated metadata, with findings validated through structured assessments.

The protocol encompassed a detailed methodology, including test procedures, participant recruitment, and evaluation methods such as task-based questionnaires, the NASA Task Load Index (NASA-TLX), and the System Usability Scale (SUS). The experiment leveraged the prototype deployed within NVIDIA Omniverse to test the tool’s compatibility with realistic maintenance-planning objectives, focusing on the effectiveness of the hybrid interaction approach. The full testing script is present in Appendix A.

The main goal of the experiment was to compare, under equivalent task scenarios, whether the AI-assisted hypothesis (H1) enabled participants to identify, select, and inspect target components in dense 3D engineering assemblies more accurately, more quickly, and with lower perceived workload than the baseline hypothesis (H0).

The experiment was performed to understand how users engaged with the prototype and how well it supported the intended interaction tasks, by obtaining answers to the following questions:

- **Question 1:** Can users locate and select a specific target component in a dense assembly more reliably when using the hybrid approach compared to the manual baseline?
- **Question 2:** Does the hybrid approach reduce the time and cognitive effort required to narrow from a visually dense scene to the correct component?

The findings directly address the dissertation’s research question, which pertains to whether a hybrid interaction approach, combining conversational natural-language input with conventional manual interaction, can effectively support the spatial and thematic selection and filtering of individual objects within complex and dense 3D industrial models.

6.2 Participants and Recruitment

The prototype was validated by CERN members, both by people familiar with the physical facility represented in the model and by people encountering them for the first time in 3D. The facility model used throughout the study is the PS Booster, introduced in Section 2.1.2. Since no participants currently use Omniverse for their daily work, both hypotheses shared the same learning curve for basic 3D navigation, making the comparison fair. Testing with people unfamiliar with the prototype was an opportunity to measure its intuitiveness. Nevertheless, testing with facility coordinators and engineers who understand the domain context was crucial to assess whether the tool supported realistic workflows.

Only CERN members were considered, as the tests needed to be conducted in person. Participant recruitment was supported by the supervisor Fernando Pedrosa, who contacted members of the relevant sections so that testing sessions could then be directly scheduled.

6.3 Ethical Considerations

The study involved human participants performing interaction tasks. Ethical handling therefore included informing participants of the study purpose and procedure, obtaining informed consent prior to participation, and ensuring that participation was voluntary and could be terminated at any time upon the participant’s request without penalty. Any logged interaction data and questionnaire responses were stored and reported in a way that preserves anonymity and confidentiality. Participants could request access to their recorded data and could request its removal from the study dataset within the limits of the study timeline and analysis process. Data were used exclusively to evaluate interaction performance and perceived usability characteristics of the proposed workflow.

6.4 Questionnaire Design

The main assessment methods consisted of a task-based questionnaire, the NASA-TLX, and a SUS questionnaire. Before the main questionnaire, demographic questions characterized the participants. After the tasks under both hypotheses, questions about preferences and broader feedback were collected.

The questionnaire started with an introduction explaining its purpose and context (Informed Consent Form), followed by a section collecting demographic details. Next, participants completed tasks under each hypothesis to assess their ability to locate, select, and inspect components. They then evaluated each hypothesis's workload and usability and provided open-ended feedback for deeper insights.

6.4.1 Pre-Test Questionnaire

Participants were asked:

1. Age range.
2. Gender.
3. Role at CERN.
4. How familiar are you with the physical facilities represented in the 3D model (PS Booster)? (5-point ordinal familiarity scale: 1 = I do not know it at all; 5 = I know the area very well).
5. How much experience do you have with 3D model manipulation and visualization software? (5-point ordinal experience scale: 1 = Not familiar at all; 5 = Extremely familiar).
6. Have you previously used NVIDIA Omniverse? (Yes / No).
7. How frequently do you use conversational AI tools (e.g., ChatGPT, Claude)? (5-point ordinal frequency scale: 1 = Never; 5 = Daily).

6.4.2 Task-Specific Questionnaire

Each task required the participant to locate a specific component in the PS Booster model and retrieve information about it from the Info Window, thereby confirming that the correct component was selected. Three tasks were used in the main study, spanning different interaction situations: T1 required disambiguation among repeated similar components; T2 required identifying the components spatially connected to a given reference component; T3 combined visibility management with disambiguation of similar components nested inside a parent assembly. A fourth task originally piloted as a warm-up component-identification task was removed after the pilot test (Section 6.9). Its outcome carried the least informational value relative to its time cost.

Tasks were framed using the types of references that domain users would realistically use when searching for a component: equipment type names, spatial designations, and task or intervention

context. The participant confirmed correct identification by reading back metadata from the Info Window.

Because each participant performed tasks under both H0 and H1, there was a risk that completing a task in the first condition made it easier in the second through memorization of the target's location. To mitigate this, two parallel task sets were defined: Set A and Set B. Both sets were structurally identical and targeted components of the same type and difficulty, but in different parts of the model. The set assigned to each hypothesis was counterbalanced across participants in combination with the hypothesis order counterbalancing. The complete task sets are listed in Section A.5.

T1 - Disambiguation of Similar Components The task required the participant to locate a target component within a specified subsystem and area. Several similar components were present in this area. A reference card showed the component type and the approximate local context. The participant had to find the correct component and report the task constraint type associated with it, as shown in the Info Window.

Steps: Navigate to the area and inspect the similar components using the reference card to help differentiate them. Select the component associated with the maintenance task and verify the task information in the Info Window.

- In H0, the participant performed the search using navigation and metadata inspection only.
- In H1, the participant could additionally use the chat and the CAD Tree to refine the search using the task context and the local scene context.

Success condition: The correct task constraint type associated with the selected component is identified.

Answer format: Multiple-choice (task constraint type).

Per-task metrics: Success, completion time.

T2 - Connected Components The task required the participant to identify all components directly connected to a specified reference component within a given assembly. A reference card showed the component in the model. The participant had to locate it and identify all components directly connected to it, reporting their names as shown in the Info Window or in the scene hierarchy.

Steps: Locate the target component using the reference card and inspect its immediate local context in the model.

- In H0, identify the target and its neighboring components using navigation alone.
- In H1, the participant could additionally use the chat and the CAD Tree to inspect the connected components.

Success condition: The correct set of connected component names is identified.

Answer format: Multiple-choice (set of connected component names).

Per-task metrics: Success, completion time.

T3 - Component Search with Visibility Management and Disambiguation The task required the participant to locate a target component inside a specified parent assembly and period. The target was partially occluded by surrounding structures, and several visually similar components were present in the same area. A reference card showed the component image, its general location within the parent assembly, and the surrounding context. The participant had to select the correct component and report the duration of the associated installation task as shown in the Info Window.

Steps: Use the reference card to interpret the target's location and visual cues. Manage the visibility of surrounding components to expose the target area, then identify and select the correct instance among the similar components present. Inspect the Info Window to determine the duration of the associated installation task.

- In H0, the participant managed visibility using navigation and the default Omniverse stage tree, then disambiguated among similar components by inspecting the raw USD properties of each candidate.
- In H1, the participant could additionally use the chat and the CAD Tree to isolate or hide surrounding structures and to narrow down the correct instance using metadata constraints, such as task association or layout name.

Success condition: The correct component is selected and the duration of the associated installation task is correctly identified.

Answer format: Multiple-choice (task duration).

Per-task metrics: Success, completion time.

6.4.3 Post-Test Questionnaire

After completing the tasks under both hypotheses, the following assessments were administered:

NASA-TLX (per hypothesis)

The standard NASA Task Load Index was administered once per hypothesis. The participant rated the workload experienced during that hypothesis across six subscales: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. Each subscale was rated on a 21-point scale. The raw TLX score (unweighted average) was used.

System Usability Scale (per hypothesis)

A standard SUS questionnaire was administered once per hypothesis. Each of the 10 statements was rated from 1 to 5, where 1 means the participant strongly disagrees and 5 means the participant strongly agrees.

Comparative preference and feedback questions

After completing both SUS forms:

1. Which interface made you feel more confident that you had selected the correct component? (Manual interface / Chat-assisted interface / No difference / It depended on the task).
2. Which interface did you find easier to use? (Manual interface / Chat-assisted interface / No difference / Not sure).
3. If both approaches were available, how would you prefer to work? (Mostly manual / Mostly conversational / Mix of both / Not sure).
4. How predictable did you find the chat-assisted interface? (5-point ordinal predictability scale: 1 = Very unpredictable; 5 = Very predictable).

At the end, four open-ended questions were asked:

1. What were the most difficult aspects of locating and selecting components in the model?
2. Were there situations where the conversational interface was particularly helpful or unhelpful?
3. Do you think a tool like this would be useful for your work at CERN? Why or why not?
4. Do you have any other comments, including recommendations for improvement or feature suggestions?

6.5 Test Moderator

The researcher was the one conducting and facilitating all usability testing sessions, with additional responsibility for scheduling. To reduce moderator-induced variability, the same researcher conducted all sessions using a standardized introduction and orientation procedure.

Before the task phase began, the moderator explained the purpose of the study, the available interaction mechanisms, and the general flow of the session. The moderator could clarify interface mechanics that were already explained during the orientation, such as how to navigate, select components, inspect the Info Window, or use the chat in H1. However, the moderator did not provide target-specific hints, indicate the correct component, recommend a preferred strategy, or suggest that one hypothesis was expected to perform better than the other. If a participant was not able to advance on a task, only previously explained concepts could be clarified. Any such intervention was recorded, and failure to answer correctly after clarification was counted as an incorrect response.

6.6 Apparatus

All experimental sessions were conducted in person at CERN between May 4th and May 26th.

A computer with a web browser and Internet access was required. The prototype was accessed through a browser-based interface that connected to a remote machine running NVIDIA Omniverse and streaming the application. This setup allowed the experiment to be performed independently of the participant's local machine, while keeping the Omniverse environment and prototype configuration consistent across sessions. A standard mouse and keyboard were required.

Before each session, the scene and the interface were verified to ensure that the required components and functionality were available. Depending on the assigned hypothesis, participants interacted either with the baseline interface alone or with the same interface extended by the conversational assistant. In both cases, selecting a component displayed its metadata in the Info Window.

6.7 Experiment Procedure

6.7.1 Procedure

The steps each participant followed are detailed below:

1. The moderator briefly introduced the study and asked the participant to read and sign the Informed Consent Form (Section A.2).
2. The participant answered a short pre-test questionnaire covering age, gender, role at CERN, familiarity with the PS Booster, experience with 3D model visualization, prior use of NVIDIA Omniverse, and frequency of use of conversational AI tools.
3. The participant was given a tool orientation. The moderator first demonstrated the basic Omniverse viewport interactions (orbit, pan, zoom, click-to-select) and showed the Info Window with metadata-on-selection. The participant was encouraged to click around and explore freely until they felt comfortable with the camera controls.
4. Participants were assigned to one of the four (hypothesis order $\times$ task set) combinations using a pre-prepared balanced assignment list.
5. First assigned hypothesis (H0 or H1, according to the counterbalancing assignment):
 - If this was H0, only the baseline Omniverse interface was available: the viewport, the Info Window, and the default Omniverse scene tree.
 - If this was H1, the chat panel and the CAD Tree were additionally visible. The moderator gave a brief orientation on what the chat understood and how to type queries, and showed and explained the CAD Tree. The participant was invited to experiment with a few example queries before starting the tasks.

- The participant then performed the three tasks of the assigned task set sequentially. For each task, the moderator read the task prompt aloud and presented the reference card. The participant performed the required actions and verbalized their answer; the moderator recorded the answer and the completion time.
6. Before the second hypothesis began, the moderator reset the scene to its baseline state. This reset included clearing the current selection, restoring the default camera, restoring default visibility for all components, clearing the chat history (if used), and resetting the CAD Tree state.
 7. Second assigned hypothesis (the remaining one). The same three tasks were performed under the other hypothesis, using the alternative task set so that the participant had not seen the specific targets before.
 8. The NASA-TLX was administered once per hypothesis (the participant rated each hypothesis separately).
 9. A standard SUS questionnaire was administered once per hypothesis.
 10. The comparative preference questions and the open-ended questions were administered once at the end of the session.

6.7.2 Completion Criteria

The experiment was deemed complete once the participant had signed the Informed Consent Form and completed all required sections of the protocol, namely the pre-test questionnaire, the task phase under both hypotheses, and the post-test questionnaire.

6.8 Evaluation Methodology

6.8.1 Research Design

The study used a within-subjects design in which each participant completed the tasks under both H0 and H1, allowing a direct paired comparison while controlling for individual differences in domain knowledge and spatial ability.

Two sources of order effects were controlled through counterbalancing. First, hypothesis presentation order was counterbalanced, so that approximately half the participants began with H0 and the other half with H1. Second, to prevent participants from memorizing target locations between rounds, two structurally identical task sets, Set A and Set B, targeted the same component types in different periods of the model. Task-set assignment was counterbalanced independently of hypothesis order, so the four possible combinations were distributed evenly across participants, as shown in Table 6.1.

Table 6.1: Counterbalancing of hypothesis order and task-set assignment across the four participant groups.

	First block	Second block
Group 1	H0 - Set A	H1 - Set B
Group 2	H0 - Set B	H1 - Set A
Group 3	H1 - Set A	H0 - Set B
Group 4	H1 - Set B	H0 - Set A

6.8.2 Objective Performance Metrics

In the experiment questionnaire, each task was scored binarily: a correct answer earned 1 point, while an incorrect answer received 0. A task was considered correct when the participant selected the correct component and provided the requested response accurately. Correct answers were predefined by the researcher during preparation using the validated ground-truth model and task annotations.

Completion time and correctness were recorded per task per hypothesis. Completion time was measured manually by the moderator using a stopwatch. Task answers, timing values, clarification events, and any anomalies observed during the session were recorded in a structured session sheet. If any part of the recording process failed during a session, the moderator recorded the issue in the same session sheet and continued the session using manual notes.

6.8.3 Subjective Perception Metrics

The main subjective measures used in the study were the NASA-TLX and the SUS questionnaire. Both instruments were administered once per hypothesis and were used to compare perceived workload and usability between H0 and H1. Comparative preference questions and open-ended questions were additionally used to gather qualitative feedback about the perceived usefulness, predictability, and overall experience of the two hypotheses. The detailed administration of these instruments is described in Section 6.4.3.

6.8.4 Data Analysis

The collected data were first summarized using descriptive statistics for each hypothesis, including task success, completion time, NASA-TLX scores, and SUS scores. The main comparison of H0 and H1 was then carried out using paired analyses appropriate for the within-subjects design. Completion time, perceived workload, and perceived usability were compared with Wilcoxon signed-rank tests, each reported with a rank-biserial effect size and a Hodges–Lehmann estimate of the median difference together with its confidence interval. Task success, which was high under both conditions, was summarized descriptively rather than tested. Comparative preference responses were summarized descriptively, and open-ended feedback was reviewed qualitatively in order to identify recurring themes related to usefulness, predictability, and interaction difficulties.

6.9 Pilot Test

A pilot test was conducted with one representative user from the EN-ACE section prior to the main study. The structure of the pilot was the same as presented above. The pilot served to:

- Validate that task prompts were clear and that ground-truth answers were unambiguous.
- Confirm that the target session duration (~50 minutes) was achievable.
- Identify usability or procedural issues with the orientation, task flow, or questionnaires.
- Verify that the interaction logging captured all required events.

The main outcome of the pilot was that the original four-task protocol exceeded the target session duration. To bring the session within the target, one task was dropped from the protocol. The dropped task was the original component-identification task (locating a component in a specified area and reporting whether any maintenance task was associated with it), which had the lowest informational value relative to its time cost. It was essentially a navigation warm-up that did not exercise the disambiguation, spatial-relation, or visibility-management aspects that the remaining three tasks target. With this change, the protocol moved to three tasks (T1 - disambiguation, T2 - connected components, T3 - visibility management with disambiguation), and pilot data was not included in the main analysis.

6.10 Threats to Validity

Several threats to validity should be considered when interpreting the results of this study, arising both from the design of the comparison and from the conditions under which it was run.

The most significant concerns the composition of the hybrid condition. The hybrid interface extended the baseline with both the conversational agent and the CAD Tree, whereas the research question concerns conversational, natural-language input in particular. Because the two were introduced together, the design cannot separate the contribution of the conversational agent from that of the CAD Tree, which on its own already replaces raw USD names with human-readable descriptions. Any improvement measured under the hybrid condition therefore reflects the combined effect of both additions. The per-task pattern offers partial reassurance, since the largest improvement appeared on the task that depended most on the conversational spatial query rather than on browsing the tree, and the qualitative feedback repeatedly singled out the chat as a source of value. Even so, isolating the two contributions would require a dedicated experiment, and the inability to do so here is a limitation of the design.

All sessions were conducted by the researcher, who also developed the prototype. A standardized introduction and a strict policy against target-specific hints were used to limit moderator influence, but the possibility of unintentional bias cannot be ruled out. In addition, each participant

took part in a single session and met the prototype for the first time, so the appeal of a new and unfamiliar way of interacting may have contributed to the positive reception of the hybrid interface, independently of its practical benefit.

Task correctness was high under both conditions, which left little room for the hybrid interface to show an advantage on that measure. Correctness is therefore the least discriminating of the outcomes, and the comparison rests mainly on completion time, perceived workload, and usability rather than on accuracy.

Completion time was measured by the moderator with a stopwatch, started once the task prompt had been read aloud. This introduces a measurement error on the order of human reaction time. The error is small relative to the differences observed on the longer tasks, but it is comparable to the difference on the shortest task, where only a few seconds separated the two conditions. Time differences on that task should be read with corresponding caution.

A further threat concerns what the measured outcomes attribute. The hybrid condition is a system in which a language model interprets every request, so any turn in which the model misreads a request, phrases a tool call that the modifier layer cannot repair, or returns a clarification where an immediate action was expected, costs the participant time that is then recorded as the performance of the interface. The study did not instrument the agent to separate these two contributions, and no measure of interpretation accuracy was collected during the sessions. Completion times under the hybrid condition are therefore the performance of the whole system, the model included, rather than of the interaction design alone. The direction of this effect is worth noting, in that interpretation failures lengthen the hybrid times and so work against the hypothesis, which makes the reported advantage conservative rather than optimistic. What the omission does prevent is any claim about how much of the remaining time is attributable to the interaction design as opposed to the model behind it.

6.11 Summary

The primary objective of the experiment was to assess whether the proposed hybrid interaction approach, combining conversational natural-language input with conventional desktop interaction, improved the reliability and efficiency of component identification and selection in dense 3D engineering assemblies, compared to a purely manual baseline. The experiment directly addressed the dissertation's research question.

Conducted in person at CERN, participants interacted with the prototype through a browser-based interface connected to a remotely hosted NVIDIA Omniverse session, navigating the PS Booster model and performing component-identification tasks under both hypotheses. The methodology adopted a within-subjects design with hypothesis order and task set counterbalanced across participants. Following the pilot test, the protocol used three tasks: a disambiguation task (T1), a connected-components task (T2), and a visibility-management-and-disambiguation task (T3).

Assessment methods comprised a task-based questionnaire with binary scoring, completion time measurement, the NASA-TLX for perceived cognitive workload, the SUS for overall usability satisfaction, comparative preference questions, and open-ended feedback. The pre-test questionnaire characterized participants in terms of role, domain familiarity, 3D-software experience, prior Omniverse use, and frequency of conversational AI use. The post-test instruments were administered once per hypothesis where applicable.

A pilot test with an EN-ACE colleague refined the protocol prior to the main study, leading to the removal of one task to bring the session within the target duration. The experiment's success criteria, based on task accuracy, completion time, and perceived workload, provided a framework for validating the hybrid interaction approach against the manual baseline.

Chapter 7

Results

This chapter presents and discusses the outcomes of the experimental evaluation of the prototype, involving 16 CERN members who completed three component-identification tasks under two interface configurations: the baseline Omniverse interface and the hybrid interface incorporating the conversational agent. The findings reveal insights into how the conversational layer affects task performance, cognitive load, and perceived usability, drawn from demographic questions, task-based measurements, the NASA-TLX questionnaire, the System Usability Scale (SUS) questionnaire, and comparative preference questions. Organized into a sample characterization, task performance analyses, cognitive load and usability scores, comparative preferences, and validity checks, the results assess the effectiveness and intuitiveness of the hybrid interaction model, informing potential enhancements for its application to maintenance planning at CERN.

7.1 Sample Characterization

The evaluation involved 16 participants drawn from different departments across CERN, covering a wide range of professional roles.

The age distribution is shown in Fig. 7.1a. Participants ranged from 27 to 50 years of age, with a mean of 36.1 and a standard deviation of 9.3. The sample was 14 male and 2 female.

Prior experience with 3D modeling and visualization software, shown in Fig. 7.1b, was spread across the whole 1-5 scale. Most participants placed themselves at the middle levels, with a smaller number reporting either very low or very high familiarity, so the sample is neither uniformly expert nor uniformly inexperienced. Only one participant had previously used NVIDIA Omniverse itself, so prior exposure to the specific platform was effectively absent across the sample. This is a relevant point when interpreting the usability results, since for almost every participant both interfaces were encountered for the first time during the session.

Familiarity with the PS Booster facility, the model used in the evaluation, is shown in Fig. 7.2. It varied across the sample, from participants who had only heard of the facility to participants who knew the area very well. This range is relevant to the interpretation of the task results, because a participant who is already familiar with the facility has a better mental model of its

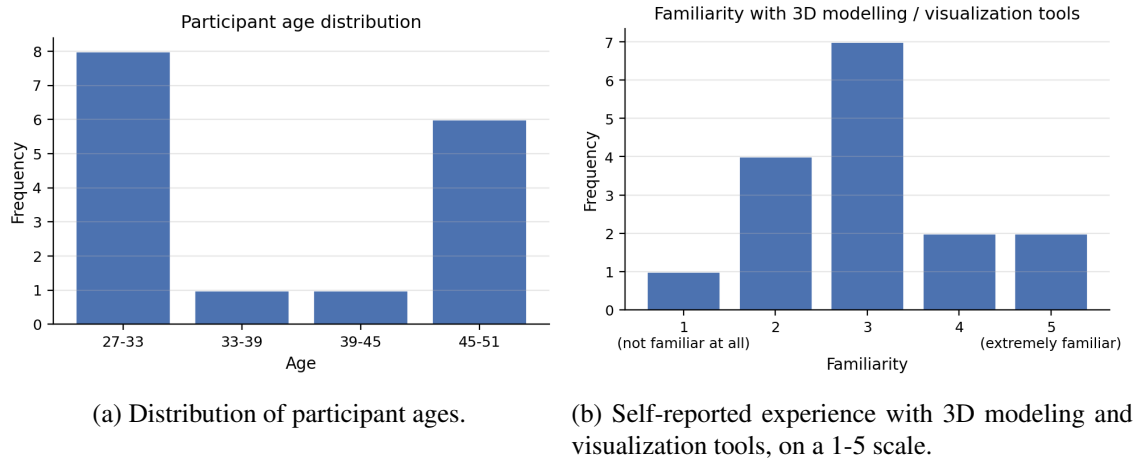


Figure 7.1: Participant demographics, including age distribution and self-reported experience with 3D modeling and visualization tools.

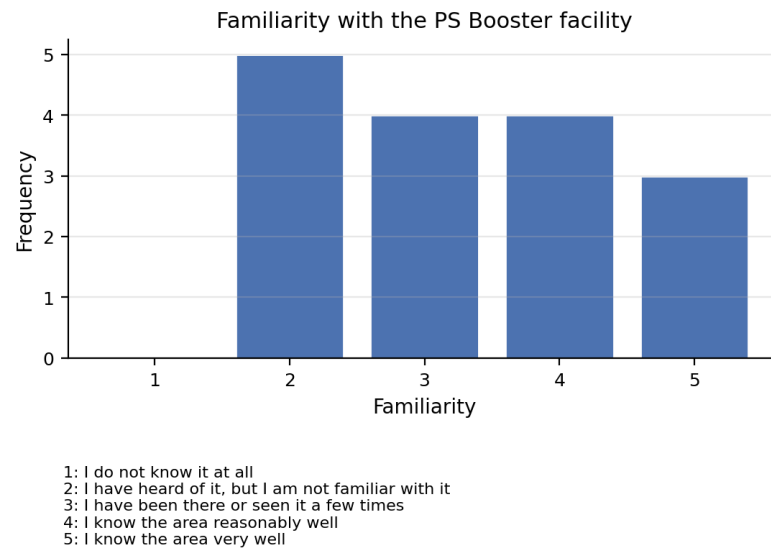


Figure 7.2: Self-reported familiarity with the PS Booster facility.

overall layout and of the components it contains, which can give an advantage when locating a component regardless of the interface used.

Finally, Fig. 7.3 shows how frequently participants reported using conversational AI tools such as ChatGPT, Claude, or Copilot. Most participants reported occasional use, with several reporting daily use and a small number reporting only rare use. The conversational interface evaluated here is therefore being judged by a sample that is broadly but not uniformly accustomed to conversational AI.

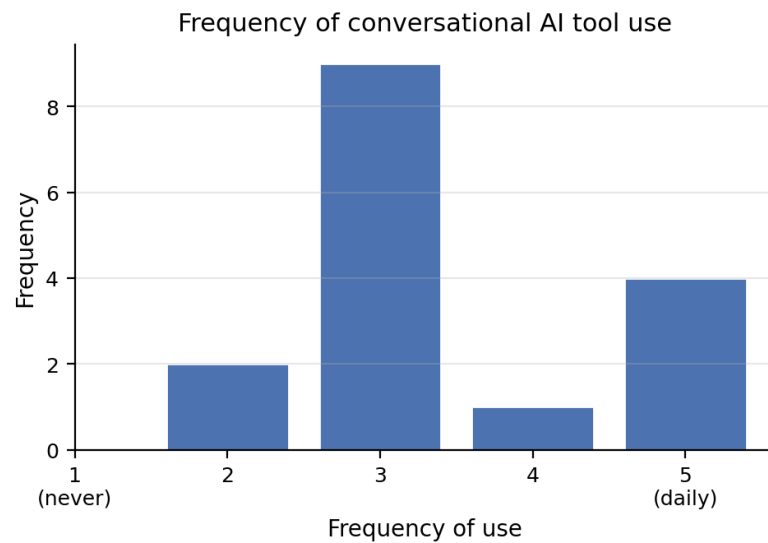


Figure 7.3: Self-reported frequency of conversational AI tool use.

Participants were assigned to one of four counterbalancing groups, which vary both the order of the two interfaces and the order of the two task sets. Four participants were assigned to each group, an even split that indicates the counterbalancing was applied as designed.

7.2 Task Performance

This section examines how participants performed the three component-identification tasks under each interface, in terms of correctness, the time taken to complete each task, and the number of hints the moderator had to provide.

7.2.1 Correctness

Each task was scored as correct or incorrect. Correctness is reported first, because the completion-time results that follow are only meaningful if the faster condition did not also produce more errors.

Figure 7.4 shows the correctness rate for each task and condition. Correctness was high throughout and very similar between the two interfaces. Under the baseline, tasks T1, T2 and T3 were answered correctly by 14, 13 and 16 of the 16 participants respectively, against 15, 16

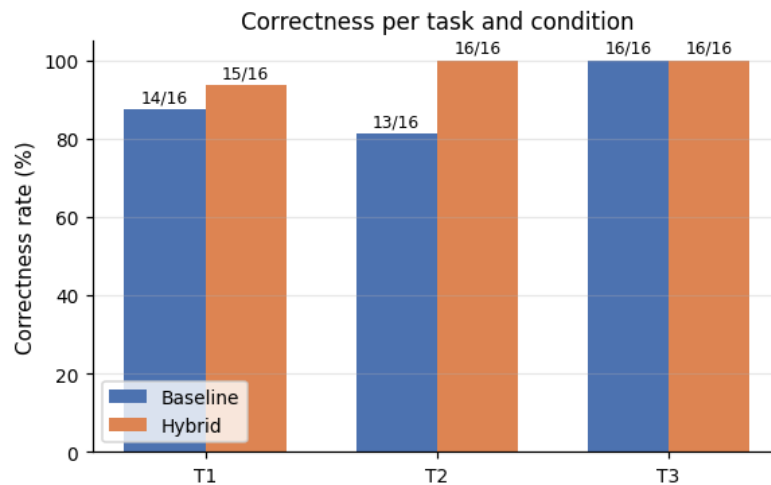


Figure 7.4: Correctness rate per task under each condition, annotated with the number of correct participants out of 16.

and 16 under the hybrid. The hybrid interface was therefore at least as accurate as the baseline on every task, and slightly more accurate on T1 and T2. Because correctness is near the ceiling under both conditions, it does not by itself discriminate between the interfaces, but instead confirms that the speed differences reported next did not come at the expense of accuracy.

7.2.2 Completion Time

Completion time was recorded for each task as the interval between the moderator finishing the task prompt and the participant giving a final answer.

Figure 7.5 compares completion times across the two conditions. The hybrid interface was faster on all three tasks. The median completion time fell from 70 seconds under the baseline to 49 seconds under the hybrid on T1, from 106 to 49 seconds on T2, and from 39 to 28 seconds on T3. The largest gain was on T2, the one task where the CAD Tree’s readable descriptions offer no help, since adjacency must be derived from the geometry. The hybrid interface was faster for 11 of the 16 participants on T1, for all 16 on T2, and for 12 of the 16 on T3. A Wilcoxon signed-rank test [68] confirmed that these differences were statistically significant on every task (T1: $W = 26$, $p = 0.029$, rank-biserial $r = 0.38$; T2: $W = 0$, $p < 0.001$, rank-biserial $r = 1.00$; T3: $W = 28$, $p = 0.039$, rank-biserial $r = 0.50$). Taking the three tasks together, total completion time was also significantly lower under the hybrid interface ($W = 6$, $p < 0.001$), with a large effect (rank-biserial $r = 0.88$) and a Hodges–Lehmann median reduction of 91.8 seconds (95% CI [51.0, 130.0]).

The contrast is sharpest on T2, the connected-components task. Here the baseline median of 106 seconds was more than double the hybrid median of 49 seconds, and not a single participant was slower with the hybrid interface. This is the task most directly served by the spatial query of the conversational interface because under the baseline, a participant has to identify the components surrounding a reference component by visual inspection or by traversing the model hierarchy by hand, whereas the hybrid interface answers the same question with a single natural-language

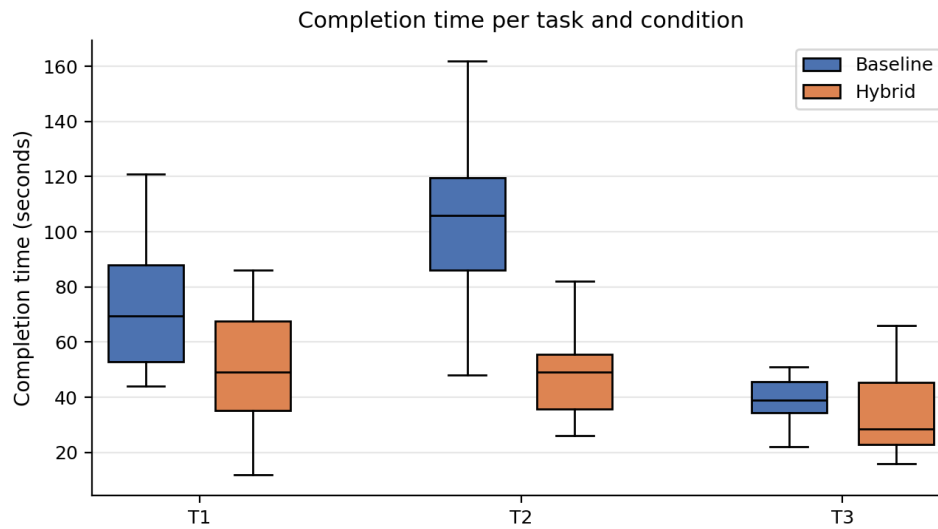


Figure 7.5: Completion time per task under each condition. Boxes show the interquartile range and median; whiskers show the range excluding outliers.

query. T1 and T3 also favored the hybrid, though with more variation between participants, as the wider and partly overlapping boxes for those tasks show.

Beyond the difference in central tendency, the two conditions also differ in the spread of completion times across participants: the baseline boxes are not only higher but also taller, with the spread especially wide on T1 and T2. Under the baseline, how quickly a participant located a component depended heavily on how comfortable they were navigating a dense 3D scene. Under the hybrid interface that dependence is reduced, because the natural-language query removes the manual-navigation step that rewards practiced spatial skill. The hybrid interface narrowed the gap between the quickest and the slowest participants, which is consistent with the idea that it lowers the barrier for users who find manual navigation hardest.

7.2.3 Hints Required

During each task the moderator could provide a hint if a participant was unable to proceed. The number of hints is a coarse, moderator-judged measure, depending on when the moderator chose to intervene and is not perfectly comparable across sessions.

The averages are low under both conditions, below one hint per task in almost every case, and do not show a consistent advantage for either interface. Under the baseline, the mean number of hints per task was 0.38 on T1, 0.69 on T2, and 0.44 on T3; under the hybrid, the corresponding means were 0.56, 0.74, and 0.31. Across all three tasks and all 16 participants, the moderator gave 24 hints under the baseline and 26 under the hybrid.

The hints were not scripted or given at fixed points, being prompted by the situation, either when a participant became stuck or when they asked whether a particular action was possible. Their character differed between the two conditions in a way that reflects the nature of each interface. Under the baseline, hints were mostly procedural, concerning the manual navigation and

selection controls, such as how to move the camera or how to reach an occluded component. Under the hybrid, they more often concerned the scope of the conversational interface, clarifying what the chat could and could not be asked to do, since participants were not always sure of its boundaries.

Given how small the numerical differences are, how few hints were given in total, and the subjective nature of the measure, no conclusion is drawn from the hint counts themselves beyond the observation that neither interface left participants frequently unable to proceed.

7.3 Cognitive Load

Perceived cognitive load was measured with the NASA-TLX questionnaire, administered once per condition. Each of the six dimensions was rated on the standard scale from 0 to 100. The unweighted Raw TLX score, computed as the mean of the six ratings, was used as the overall workload measure. This choice follows Hart [31] discussion of Raw TLX as a simpler and widely used alternative to the original weighted procedure, which avoids the additional paired-comparison step while preserving the six-dimensional workload assessment. The paired-comparison weighting procedure was therefore not administered.

Figure 7.6 shows the mean rating for each of the six TLX dimensions under both conditions. The hybrid interface lowered the mean rating on every dimension without exception. The overall Raw TLX score, the mean across the six dimensions, fell from 25.4 under the baseline to 12.1 under the hybrid, and 14 of the 16 participants reported a lower overall workload with the hybrid interface. A Wilcoxon signed-rank test confirmed that this reduction was statistically significant ($W = 3$, $p = 0.001$), with a large effect (rank-biserial $r = 0.87$) and a Hodges–Lehmann median reduction of 12.3 points (95% CI [6.2, 20.0]).

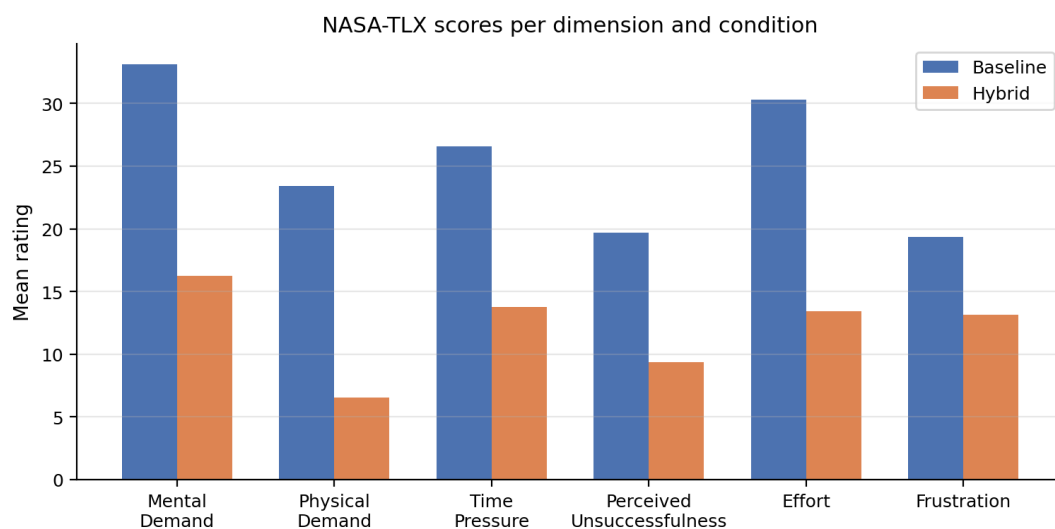


Figure 7.6: Mean Raw NASA-TLX rating by dimension and condition.

The reduction is largest on the dimensions most closely tied to the effort of finding a component. Mental demand fell from a mean of 33.1 to 16.2 and effort from 30.3 to 13.4. Physical demand, although low under both conditions, also fell substantially, from 23.4 to 6.6, plausibly because the hybrid interface replaces repeated manual camera movement with a typed query. The smallest change is on the frustration dimension, which fell only from 19.4 to 13.1. This pattern is consistent with the nature of the intervention: the hybrid interface changes how a component is found, so the dimensions tied to the mental effort and the perceived difficulty of the search improve most, whereas frustration, a more general affective response, was already moderate and changed least.

7.4 System Usability Scale

The SUS questionnaire was administered once per condition. The standard scoring procedure was applied: positively worded items contribute their rating minus one, negatively worded items contribute five minus their rating, and the sum is multiplied by 2.5 to give a score in the range [0, 100].

Figure 7.7 compares the mean SUS score between the two conditions. The mean rose from 68.3 under the baseline condition to 78.9 under the hybrid condition. A Wilcoxon signed-rank test confirmed that this increase was statistically significant ($W = 13.5$, $p = 0.008$), with a large effect favoring the hybrid interface (rank-biserial $r = 0.60$); 12 of the 16 participants rated the hybrid interface as more usable, against three who scored the baseline higher, and the Hodges–Lehmann median improvement was 7.5 points (95% CI [2.5, 20.0]). Interpreted against the acceptability ranges discussed by Bangor, Kortum, and Miller [6], the baseline mean lies close to the transition from marginal to acceptable usability, whereas the hybrid mean is clearly within the acceptable range and moves toward the higher “good” region of the scale.

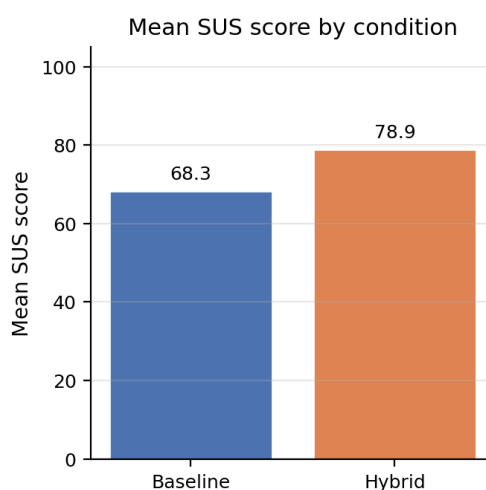


Figure 7.7: Mean SUS score under each condition.

Beyond the change in the mean, the two conditions differ in how much the scores varied between participants. The baseline SUS scores had a standard deviation of 14.5 and ranged from 30 to 85, whereas the hybrid scores had a standard deviation of only 7.1 and ranged from 70 to 95. Participants disagreed considerably about how usable the baseline interface was, but converged on a more uniformly positive assessment of the hybrid. A narrowing of this kind suggests that the hybrid interface reduced the gap between participants who adapted easily to the baseline and those who struggled with it.

The largest advantage for the hybrid interface is on the item concerning willingness to use the system frequently, followed closely by the items on ease of use and on how quickly others would learn the system. The negatively worded items, which concern complexity, inconsistency, and the need for technical support, changed comparatively little, suggesting that the hybrid interface was rated more positively mainly on how desirable and approachable it felt to use, rather than on the absence of problems.

7.5 Comparative Preference

After completing both blocks, participants answered comparative questions asking which interface gave them more confidence in their selection, which they found easier to use, and how they would prefer to work if both modalities were available. A separate question asked how predictable they found the chat-assisted interface.

When asked which interface gave them more confidence in their selection, 12 of the 16 participants (75%) chose the chat-assisted interface, two (12%) said it depended on the task, and one each chose the manual interface or reported no difference. When asked which interface was easier to use, 10 participants (62%) chose the chat-assisted interface, three (19%) reported no difference, two (12%) chose the manual interface, and one (6%) was not sure.

The third question is the most informative for the design argument of this work. The hybrid interface is not intended as a replacement for the manual interface but as the combination of a conversational agent with the existing manual tools. Fifteen of the 16 participants (94%) chose to work with a mix of both modalities, and the single remaining participant chose “mostly conversational”; not one participant chose to work mostly or entirely with the manual interface. A preference for using both modalities together is therefore not an equivocal result: it is direct evidence that the conversational layer adds value on top of the manual interface rather than competing with it.

Figure 7.8 shows how predictable participants found the chat-assisted interface. The responses concentrate at the upper end of the scale: 12 of the 16 participants (75%) gave a rating of 4 or 5, 3 gave a rating of 3, and only 1 rated the interface below the midpoint. Predictability is a known concern for conversational interfaces, where the same query can in principle produce different outputs across runs, and the responses indicate that this concern did not arise during the evaluation sessions.

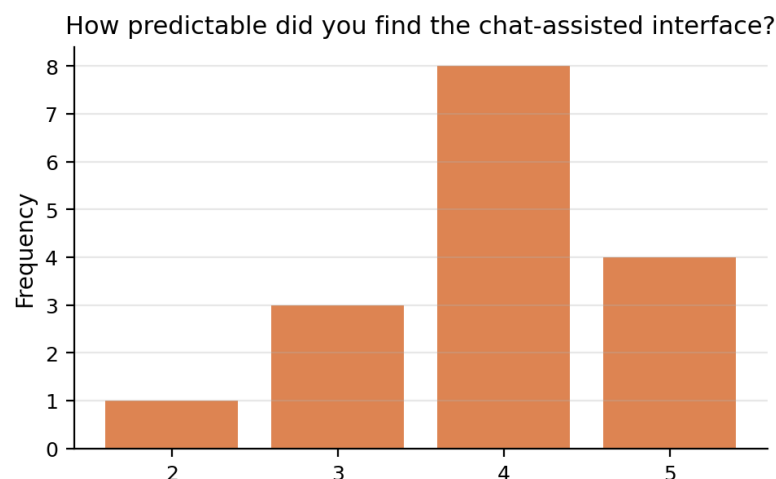


Figure 7.8: Responses to “How predictable did you find the chat-assisted interface?”.

7.6 Order and Task-Set Effects

The two task sets were designed to be of comparable difficulty, and the order in which the two interfaces were presented was counterbalanced across participants. This counterbalancing is the primary control: because interface order and task-set assignment are balanced across the sample, any residual learning effect or difference in task-set difficulty is distributed equally between the baseline and the hybrid condition and cannot systematically favor either interface. The checks below confirm that the balance held in the data. The safeguard itself comes from the design, not from the tests.

To check for a learning effect, each participant’s mean completion time in their first block was compared with their mean time in their second block, regardless of which interface each block contained. Since every participant contributes both a first-block and a second-block time, the two are paired, so a Wilcoxon signed-rank test [68] was used. The two blocks were almost identical, with means of 61.5 and 62.5 seconds ($W = 67$, $p = 0.98$, rank-biserial $r = 0.02$): a difference of about one second and a negligible effect size, giving no indication that participants improved simply through practice.

The two task sets are less cleanly matched. Under the baseline interface, the mean completion time was 81.7 seconds for the participants who saw task set A and 73.2 seconds for those who saw task set B, a gap of roughly eight seconds and larger than the difference seen between blocks. A Mann-Whitney U test [46] does not flag this as significant ($U = 36$, $p = 0.72$, rank-biserial $r = -0.13$), but with only eight participants per group the test has very little power, so the non-significant result cannot be read as evidence that the two sets are equally difficult. What keeps this residual difference from biasing the comparison is, again, the counterbalancing: each task set was paired with each interface equally often, so any difficulty gap between the sets is shared evenly between the baseline and the hybrid condition.

To make these controls visible at the level of the individual configurations rather than only in

aggregate, Table 7.1 reports the mean completion time, Raw NASA-TLX score, and SUS score for each of the four counterbalancing groups. Two observations follow from it. First, the hybrid interface outperformed the baseline on all three measures within every one of the four configurations, so the overall advantage is not produced by any single ordering or task-set pairing. Second, the breakdown reveals an order effect on the subjective measures: participants who used the baseline first rated it as more demanding and less usable (mean Raw TLX 32.2, SUS 66.2) than those who used it after the hybrid (mean Raw TLX 18.6, SUS 70.3), consistent with the baseline feeling more effortful when it is encountered without a point of comparison. Because interface order is counterbalanced, this effect is distributed equally across the two conditions and does not bias the overall comparison; it is reported here because it is the kind of effect the aggregate figures would otherwise hide, and it reinforces the value of the within-subjects, counterbalanced design.

Table 7.1: Mean outcomes for each of the four counterbalancing configurations. Each configuration lists the interface used in the first block and the interface used in the second, together with the task set paired with each.

Configuration	n	Time (s)		Raw TLX		SUS	
		Base	Hyb	Base	Hyb	Base	Hyb
Baseline A / Hybrid B	4	75.3	45.1	26.0	14.8	69.4	79.4
Baseline B / Hybrid A	4	70.2	40.8	38.3	15.4	63.1	73.1
Hybrid A / Baseline B	4	76.2	54.0	13.1	7.1	69.4	84.4
Hybrid B / Baseline A	4	88.0	46.3	24.2	11.0	71.2	78.8

Neither check, then, reveals a problem that the counterbalancing does not already address, and the per-configuration breakdown confirms that the hybrid advantage holds within each group rather than resting on the aggregate. The differences reported in this chapter can therefore be attributed to the interface rather than to the order of presentation or the choice of task set, with the caveat that a sample of sixteen participants, and only four per configuration, leaves limited power to detect a small effect of either kind.

7.7 Correlation With Demographics

This section relates the three main outcomes of the evaluation, the mean completion time across the three tasks, the overall Raw NASA-TLX score, and the SUS score, to participant characteristics that were expected to influence them: prior experience with 3D modeling and visualization tools, familiarity with the PS Booster facility, and age. Gender was not used as a grouping variable given the imbalance of the sample. The analysis is descriptive: the small number of participants per subgroup does not support formal significance testing, so the discussion focuses on whether each subgroup benefited from the hybrid interface and on the size of that benefit relative to the others.

3D experience

Participants are grouped into three categories of self-rated 3D experience: low (1 or 2, $n = 5$), medium (3, $n = 7$), and high (4 or 5, $n = 4$). Table 7.2 reports the mean outcomes for each group under both conditions. Completion time, perceived workload, and usability improved in every group. The most striking observation is that the high-experience group, which might have been expected to need the hybrid interface least, in fact benefited from it the most, starting from the heaviest perceived workload and the lowest usability rating under the baseline and ending close to the other groups.

Table 7.2: Mean outcomes by self-rated 3D experience, baseline vs. hybrid.

Group	n	Time (s)		Raw TLX		SUS	
		Base	Hybrid	Base	Hybrid	Base	Hybrid
Low (1–2)	5	79.8	50.2	26.2	17.0	71.5	76.0
Medium (3)	7	68.9	43.7	21.0	9.4	75.4	79.6
High (4–5)	4	89.5	46.9	32.3	10.6	51.9	81.2

This counter-intuitive result is best understood together with age, and is discussed below, because the self-rated experience measure does not appear to be uniformly calibrated across participants.

Age

Participants split into two clear age clusters, with a gap between the youngest seven and the oldest seven (no participant aged between 33 and 42). They are therefore grouped as younger (< 35 , $n = 9$) and older (≥ 35 , $n = 7$). Table 7.3 reports the outcomes. Both groups improved on all three measures, but the younger group, which was slower and reported lower usability under the baseline, gained considerably more in completion time, converging with the older group under the hybrid.

Table 7.3: Mean outcomes by age group, baseline vs. hybrid.

Group	n	Time (s)		Raw TLX		SUS	
		Base	Hybrid	Base	Hybrid	Base	Hybrid
Younger (< 35)	9	83.2	40.1	26.6	12.7	64.2	79.4
Older (≥ 35)	7	70.0	54.8	23.9	11.3	73.6	78.2

Age also helps explain the counter-intuitive 3D-experience result. Self-rated experience was negatively correlated with age across the sample ($r = -0.43$): the younger participants tended to rate their 3D experience higher, and the older participants lower. The two participants who rated their experience at the maximum of 5 were both under 30, and both were among the three slowest participants at baseline, whereas several participants who rated themselves only 2 were among the fastest. In other words, the high-experience group owed its heavy baseline workload

and low baseline usability less to genuine expertise than to a tendency among younger participants to over-state their experience. This is a known limitation of self-reported skill measures, where less-experienced respondents may overestimate their proficiency, and it cautions against reading the 3D-experience grouping as a clean proxy for actual skill. The practical consequence for this study is mild: because the hybrid interface improved outcomes across all experience levels and both age groups, the conclusion that it helps regardless of prior 3D experience holds whether or not the self-ratings are perfectly calibrated.

PS Booster familiarity

Participants are grouped into four categories of facility familiarity, taken directly from the questionnaire: those who had only heard of the facility ($n = 5$), seen it a few times ($n = 4$), knew it reasonably well ($n = 4$), or knew it very well ($n = 3$). Table 7.4 reports the outcomes. This grouping is the most thematically informative of the three. Under the baseline, completion time fell steadily as familiarity increased, from 100.7 seconds per task for those who had only heard of the facility to around 62 seconds for those who knew it very well. Under the hybrid, this gradient largely disappeared, so the time saved depended strongly on prior facility knowledge: roughly 50 seconds per task in the least-familiar group, against about 2 seconds in the most-familiar group. The largest usability gain (from 55.0 to 78.5) was likewise in the group that had only heard of the facility.

Table 7.4: Mean outcomes by PS Booster familiarity, baseline vs. hybrid.

Familiarity	n	Time (s)		Raw TLX	
		Base	Hybrid	Base	Hybrid
Only heard of it	5	100.7	51.3	31.5	14.2
Seen a few times	4	74.2	39.1	21.0	9.8
Knows reasonably well	4	63.2	38.2	24.4	15.0
Knows very well	3	61.9	59.8	22.5	7.8

Familiarity	n	SUS	
		Base	Hybrid
Only heard of it	5	55.0	78.5
Seen a few times	4	75.6	81.2
Knows reasonably well	4	71.9	78.1
Knows very well	3	75.8	77.5

This pattern is consistent with the role of the conversational interface as a search and locating aid: participants who already know where components are in the facility derive less benefit from a tool that helps them find them, whereas participants for whom the facility is unfamiliar are exactly those for whom the hybrid interface removes the most friction.

Summary

Across all three groupings, no subgroup was disadvantaged by the hybrid interface. The largest gains consistently came from the participants who started from the worst baseline performance, whether because they were less familiar with the facility or, as the age analysis suggests, because they were younger and slower despite higher self-ratings. This indicates that the hybrid interface raises the floor for the users who struggle most without it, rather than only making the already-fast even faster.

7.8 Qualitative Feedback

Participants were asked four open-ended questions: about the most difficult aspects of locating and selecting components, the situations in which the chat-assisted interface was particularly helpful or unhelpful, the perceived usefulness of the tool for their own work at CERN, and any further suggestions for improvement.

The feedback from participants regarding the difficulties of locating components highlights several recurring concerns: knowing where a component sits within the model, dealing with components stored under internal CAD identifiers rather than recognizable names, distinguishing between many similar or repeated components, and operating the manual navigation controls themselves. Several participants pointed at the underlying naming problem, noting that “when there are many components that are named the same, when the naming convention is not perfect” the user is forced to rely on internal IDs to disambiguate. Others emphasized the navigation barrier, describing “moving around with the default controls” and “understanding the controls at the beginning” as the hardest parts of the experience, indicating a need for either clearer onboarding or an alternative way of reaching a target component. A separate observation concerned occluded geometry, namely the need to manually hide objects to reach others.

The feedback regarding the chat-assisted interface reveals a consistent and largely positive set of impressions, with participants identifying three recurring situations in which it was particularly valuable. The first is locating a specific component directly, with one participant describing how it “finds the components directly, making navigation easier” and another reporting that it was useful “finding specific components inside the dense assembly”. The second is identifying components that are connected to a reference component, a task the conversational interface handles in a single query rather than through visual inspection: participants reported it was “really helpful not to miss objects and make mistakes when looking for connected elements”, and another singled out “knowing what is connected to a component” as one of the strongest use cases. The third situation, mentioned by several participants, is working in an unfamiliar facility, with one participant noting the interface was useful “when you don’t know anything about the model” and another that it was “really helpful when you don’t know what you are looking for”. The main reported limitation concerned components that are not uniquely named, where one participant found the chat “not

really helpful” and others preferred to confirm the result manually, indicating that disambiguation of identically named components is the area in which the conversational layer is currently weakest.

When asked whether a tool like this would be useful for their own work at CERN, all but one participant answered yes. The single exception was a matter of timing rather than principle: that participant’s experiment is currently being rebuilt, so the tool would be of limited use to them at present, though not because of any doubt about the tool itself. The stated use cases clustered around four themes that suggest the tool addresses needs across several different roles. Several participants emphasized preparation and rehearsal of interventions before carrying them out, suggesting it would help to “prepare the activities on site, instead of using the drawings” and to “prepare the manipulation of components and simulate before actually doing them”. Coactivity management was another recurring theme, with participants noting its value for “understanding the space required to work on certain components” and for “scheduling of installations or maintenance”. Training and demonstration was a third theme, with participants pointing to the training of newcomers and to “show something to a different group that is not used to work with things from my group”. The fourth theme concerned inspection of components that are difficult or unsafe to reach in person, including “cable trays and paths they go through, inaccessible ones” and “elements that are not that easy to reach in real life”. These responses indicate that the tool is perceived as useful not for one specific role but across the variety of staff involved in maintenance planning at CERN.

7.9 Summary

The evaluation involved 16 CERN participants spanning a wide range of ages, roles, and levels of prior experience. The findings can be summarized as follows.

Task performance. Correctness was high under both interfaces and showed no meaningful difference between them, confirming that neither interface compromised accuracy. Completion time, however, clearly favored the hybrid interface: it was faster on all three tasks, with the median time roughly halved on the connected-components task, where every one of the 16 participants was faster with the hybrid. The spread of completion times also narrowed under the hybrid interface, indicating that it reduced the dependence of task speed on a participant’s comfort with manual 3D navigation. The number of hints required was low under both conditions and did not distinguish the interfaces.

Cognitive load. Perceived cognitive load, measured as the overall Raw NASA-TLX score, fell from a mean of 25.4 under the baseline to 12.1 under the hybrid, and the hybrid interface lowered the mean rating on every one of the six TLX dimensions. The reduction was largest on mental demand, effort and temporal demand, and smallest on frustration: a specific, explainable pattern rather than a uniform shift.

Usability and preference. The mean SUS score rose from 68.3 under the baseline to 78.9 under the hybrid, moving the system from the lower edge of the acceptable range toward the “good” band, and the spread of scores narrowed markedly. Stated preference was decisive: 15 of

the 16 participants chose to work with a combination of both interaction modes, and none chose to work mostly with the manual interface alone.

Validity. The counterbalancing checks found no evidence of a learning effect between blocks and no evidence of an imbalance between the two task sets, so the differences reported above are not attributable to the order of presentation or to the task sets.

These findings should be read in the light of the study's limitations. The sample of 16 participants is small, so the results are best understood as a prototype evaluation that indicates a direction rather than as a definitive validation; the descriptive differences reported here are consistent and, in the case of completion time and cognitive load, substantial, but a larger sample would be needed to characterize them precisely. The evaluation also used a single facility model, the PS Booster, in a single test environment, and the hint count is a moderator-judged measure that is not perfectly comparable across sessions. Within those bounds, the evaluation supports the central claim of this work: that adding a conversational layer on top of a manual 3D interface makes maintenance-planning tasks faster and less mentally demanding without sacrificing accuracy, and that users across the range of backgrounds found among CERN staff both performed better with it and preferred to keep it alongside the manual tools. The directions in which the prototype could be extended are discussed in Chapter 8.

Chapter 8

Conclusions

This dissertation set out to make dense industrial 3D models at CERN easier to navigate and query by pairing the existing manual interface with a conversational agent. Through the implementation and evaluation of a prototype built on NVIDIA Omniverse, this work addressed the difficulty of locating and selecting components in large assemblies, where the sheer scale of the model makes manual navigation tedious. This chapter summarizes the key findings, evaluates the extent to which the objectives were met, and discusses the implications and potential future directions for this research.

8.1 Final Remarks

This dissertation set out to answer a single research question:

- To what extent does a hybrid interaction approach, combining conversational natural-language input with conventional manual interaction, improve the spatial and thematic selection and filtering of individual objects within complex and dense 3D industrial models, compared to conventional manual interaction alone?

To address it, a working prototype was designed, implemented, and evaluated, extending CERN’s conventional manual interface with a conversational agent that allows components to be located and selected through natural language alongside direct manipulation.

The approach set out to provide three capabilities, which were realized to differing extents. The first, an AI-assisted conversational interface built on a large language model for natural-language querying and context-aware selection, which formed the core of the evaluated prototype. It let users describe what they were looking for in plain language rather than by an exact identifier or name, with the agent interpreting the request, resolving references and carrying out the selection, rather than requiring them to navigate to the target by hand.

The second capability, semantic-driven selection and filtering, was also realized. The prototype draws on metadata originating from CERN’s external engineering systems to isolate components by their attributes rather than by geometric visibility alone, letting the model be queried in

terms that match how engineers describe their work and linking the geometry in the viewport to the asset information that would otherwise have to be looked up separately.

The third capability, visibility-oriented interaction techniques such as interactive exploded views, was not implemented within the scope of this work. While occlusion remains one of the main difficulties when working with dense assemblies, the prototype addresses it indirectly, through conversational and semantic selection that lets users reach a component without first exposing it manually, rather than through dedicated visibility tools.

The prototype was evaluated in a comparative study with 16 CERN members, drawn from different departments and covering a range of roles involved in maintenance planning. Under the hybrid interface, participants completed the component-identification tasks faster and reported a lower cognitive load on the NASA-TLX questionnaire, and the mean System Usability Scale score rose from 68.3 under the baseline to 78.9 under the hybrid, moving from the lower end of the “acceptable” range to approach the “good” threshold. The spread of usability scores also narrowed considerably, indicating that the hybrid interface most helped the participants who had struggled with the baseline. When asked how they would prefer to work, the large majority wanted both modalities available together. With a modest sample, these results indicate a consistent direction rather than a statistically definitive effect, but that direction holds across the measures.

Together, these results answer the research question, showing that the hybrid approach supports the spatial and thematic selection and filtering of objects in dense models more effectively than the manual interface alone. The central finding of this work is that the conversational layer adds value on top of the existing manual tools rather than replacing them, which is how participants preferred to work.

While the evaluation was conducted on a single facility model, the mechanisms it relies on are not tied to it. Confining the language model to interpretation while keeping resolution deterministic, resolving spatial relations at the level at which users reason about components, and grounding language in the engineering metadata an organization already maintains are all independent of the PS Booster, of Omniverse and of the accelerator domain. What they assume is an environment with dense geometry, a hierarchical assembly structure and identifiers that link parts to engineering records, conditions common to industrial installations well beyond CERN. Confirming that transfer empirically, on a model from a different domain, remains open.

8.2 Future Work

The findings and feedback from this work provide a foundation for further development. Several directions were identified, including:

- Build on the spatial reasoning already supported by the prototype to offer richer spatial queries, such as proximity and containment relationships between components.
- Extend the conversational agent to handle more complex queries, including multi-step and comparative requests, and to resolve ambiguous references to components more reliably.

- Implement the visibility-oriented interaction techniques left out of the current prototype, such as interactive exploded views and other occlusion-management aids, to complement the conversational and manual selection already in place.
- Evaluate the conversational agent as a component in its own right. The user study measured the system as a whole, so the accuracy of interpretation was never isolated from the usability of the interaction design.
- Deepen the integration with CERN's external engineering systems, such as IMPACT and EAM, broadening the range of metadata available for selection and keeping it synchronized with the source systems so that selections reflect current asset status and maintenance records.
- Explore additional input modalities, such as voice, given that the conversational layer is currently limited to text.

These directions represent opportunities to build on the current implementation, making it more capable, better connected to CERN's engineering data, and more robust when applied to the full scale of the accelerator complex. By addressing them, the prototype can move closer to a tool that supports the day-to-day work of inspection, maintenance planning, and experiment preparation, and can inform the design of similar interfaces for other dense industrial environments.

In conclusion, this dissertation has shown that a hybrid interaction approach, combining a conversational agent with the conventional manual interface, can make dense industrial 3D models at CERN substantially easier to explore. The evaluation indicates that the hybrid interface is faster, less demanding, and more usable than manual interaction alone, and that users value the two modalities most when they are available together. While not every capability envisaged at the outset was realized, the results support the view that conversational interaction, grounded in the semantic data CERN already maintains, is a promising addition to the manual tools used to work with these models, and a foundation worth building on.

References

- [1] Saleema Amershi et al. “Guidelines for Human-AI Interaction”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Association for Computing Machinery, 2019, pp. 1–13. ISBN: 9781450359702. DOI: [10.1145/3290605.3300233](https://doi.org/10.1145/3290605.3300233).
- [2] Ferran Argelaguet and Carlos Andujar. “A survey of 3D object selection techniques for virtual environments”. In: *Computers & Graphics* 37.3 (2013), pp. 121–136. DOI: [10.1016/j.cag.2012.12.003](https://doi.org/10.1016/j.cag.2012.12.003).
- [3] Daichi Azuma et al. “ScanQA: 3D Question Answering for Spatial Scene Understanding”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 19129–19139. DOI: [10.1109/CVPR52688.2022.01854](https://doi.org/10.1109/CVPR52688.2022.01854).
- [4] Felipe Bacim, Regis Kopper, and Doug A. Bowman. “Design and evaluation of 3D selection techniques based on progressive refinement”. In: *International Journal of Human-Computer Studies* 71.7-8 (2013), pp. 785–802. DOI: [10.1016/j.ijhcs.2013.03.003](https://doi.org/10.1016/j.ijhcs.2013.03.003).
- [5] Marc Baloup, Thomas Pietrzak, and Géry Casiez. “RayCursor: A 3D Pointing Facilitation Technique based on Raycasting”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2019, 101:1–101:12. DOI: [10.1145/3290605.3300331](https://doi.org/10.1145/3290605.3300331).
- [6] Aaron Bangor, Philip T. Kortum, and James T. Miller. “An empirical evaluation of the System Usability Scale”. In: *International Journal of Human-Computer Interaction* 24.6 (2008), pp. 574–594. DOI: [10.1080/10447310802205776](https://doi.org/10.1080/10447310802205776).
- [7] Mohammad Raihanul Bashar and Anil Ufuk Batmaz. “Virtual Task Environments Factors Explored in 3D Selection Studies”. In: *Proceedings of the 50th Graphics Interface Conference*. Association for Computing Machinery, 2024, p. 16. DOI: [10.1145/3670947.3670983](https://doi.org/10.1145/3670947.3670983).
- [8] Joanna Bergström et al. “How to Evaluate Object Selection and Manipulation in VR? Guidelines from 20 Years of Studies”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–20. DOI: [10.1145/3411764.3445193](https://doi.org/10.1145/3411764.3445193).
- [9] Lonni Besançon et al. “The State of the Art of Spatial Interfaces for 3D Visualization”. In: *Computer Graphics Forum* 40.1 (2021), pp. 293–326. DOI: [10.1111/cgf.14189](https://doi.org/10.1111/cgf.14189).
- [10] Tanya Bloch. “Connecting Research on Semantic Enrichment of BIM – Review of Approaches, Methods and Possible Applications”. In: *Journal of Information Technology in Construction (ITcon)* 27 (2022), pp. 416–440. DOI: [10.36680/j.itcon.2022.020](https://doi.org/10.36680/j.itcon.2022.020).
- [11] Doug A. Bowman et al. “New directions in 3D user interfaces”. In: *International Journal of Virtual Reality* 5.2 (2006), pp. 3–14. URL: https://www.researchgate.net/publication/220222212_New_Directions_in_3D_User_Interfaces.

- [12] Jorge Camba, Manuel Contero, and Michael Johnson. “Management of Visual Clutter in Annotated 3D CAD Models: A Comparative Study”. In: *Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments (DUXU 2014)*. Vol. 8518. Lecture Notes in Computer Science. Springer, 2014, pp. 405–416. DOI: [10.1007/978-3-319-07626-3_37](https://doi.org/10.1007/978-3-319-07626-3_37).
- [13] Yang Cao et al. “Collaborative Novel Object Discovery and Box-Guided Cross-Modal Alignment for Open-Vocabulary 3D Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47.11 (2025), pp. 10475–10489. DOI: [10.1109/TPAMI.2025.3593580](https://doi.org/10.1109/TPAMI.2025.3593580).
- [14] Jeffrey Cashion, Chadwick A. Wingrave, and Joseph J. LaViola Jr. “Optimal 3D Selection Technique Assignment Using Real-Time Contextual Analysis”. In: *Proceedings of the 2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 2013, pp. 107–110. DOI: [10.1109/3DUI.2013.6550205](https://doi.org/10.1109/3DUI.2013.6550205).
- [15] Junlong Chen, Jens Grubert, and Per Ola Kristensson. “Analyzing Multimodal Interaction Strategies for LLM-Assisted Manipulation of 3D Scenes”. In: *Proceedings of the 2025 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2025, pp. 206–216. DOI: [10.1109/VR59515.2025.00045](https://doi.org/10.1109/VR59515.2025.00045).
- [16] Sijin Chen, Xin Chen, Chi Zhang, et al. “LL3DA: Visual Interactive Instruction Tuning for Omni-3D Understanding, Reasoning, and Planning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 2584–2594. DOI: [10.1109/CVPR52733.2024.02496](https://doi.org/10.1109/CVPR52733.2024.02496).
- [17] Nguyen-Thong Dang. “A Survey and Classification of 3D Pointing Techniques”. In: *2007 IEEE International Conference on Research, Innovation and Vision for the Future*. IEEE, 2007, pp. 71–80. DOI: [10.1109/RIVF.2007.369138](https://doi.org/10.1109/RIVF.2007.369138).
- [18] J. De Jonghe. *Visualizing and Exploring CAD Data in 3D, Virtual Reality and Digital Twins*. Internal Note CERN-ACC-NOTE-2022-0075. Accessed via CERN Document Server (Restricted Access). CERN, 2024.
- [19] Hao Deng et al. “Interactive Urban Context-Aware Visualization via Multiple Disocclusion Operators”. In: *IEEE Transactions on Visualization and Computer Graphics* (2015). DOI: [10.1109/TVCG.2015.2469661](https://doi.org/10.1109/TVCG.2015.2469661).
- [20] Yannick Dennemont et al. “3D Interaction Assistance Through Context-Awareness”. In: *2012 IEEE Virtual Reality Workshops (VRW)*. IEEE, 2012, pp. 157–158. DOI: [10.1109/VR.2012.6180903](https://doi.org/10.1109/VR.2012.6180903).
- [21] Niklas Elmqvist, Ulf Assarsson, and Philippas Tsigas. “Employing Dynamic Transparency for 3D Occlusion Management: Design Issues and Evaluation”. In: *Human-Computer Interaction – INTERACT 2007*. Vol. 4662. Springer, 2007, pp. 532–545. DOI: [10.1007/978-3-540-74796-3_54](https://doi.org/10.1007/978-3-540-74796-3_54).
- [22] Niklas Elmqvist and Jean-Daniel Fekete. “Semantic pointing for object picking in complex 3D environments”. In: *Proceedings of Graphics Interface 2008*. GI '08. Windsor, Ontario, Canada: Canadian Information Processing Society, 2008, pp. 243–250. ISBN: 9781568814230. URL: <https://dl.acm.org/doi/10.5555/1375714.1375755>.
- [23] Niklas Elmqvist and Philippas Tsigas. “A Taxonomy of 3D Occlusion Management for Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.5 (2008), pp. 1095–1109. DOI: [10.1109/TVCG.2008.59](https://doi.org/10.1109/TVCG.2008.59).

- [24] Niklas Elmqvist and Philippas Tsigas. “A Taxonomy of 3D Occlusion Management Techniques”. In: *IEEE Virtual Reality Conference (VR)*. 2007, pp. 51–58. DOI: [10.1109/VR.2007.352463](https://doi.org/10.1109/VR.2007.352463).
- [25] Zamir Martins Filho, Emilio Vital Brazil, and Mario Costa Sousa. “Exploded View Diagrams of 3D Grids”. In: *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE Computer Society, 2015, pp. 242–249. DOI: [10.1109/SIBGRAPI.2015.12](https://doi.org/10.1109/SIBGRAPI.2015.12).
- [26] Lukas A. Flohr et al. “Chat or Tap? – Comparing Chatbots with ‘Classic’ Graphical User Interfaces for Mobile Interaction with Autonomous Mobility-on-Demand Systems”. In: *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction (MobileHCI ’21)*. ACM, 2021. DOI: [10.1145/3447526.3472036](https://doi.org/10.1145/3447526.3472036).
- [27] Alessandro Florio, Matthias Trapp, and Jürgen Döllner. “Semantic-driven Visualization Techniques for Interactive Exploration of 3D Indoor Models”. In: *Proceedings of the 23rd International Conference on Information Visualisation (IV)*. IEEE, 2019, pp. 24–33. DOI: [10.1109/IV.2019.00014](https://doi.org/10.1109/IV.2019.00014).
- [28] Aidan Fuller et al. “Digital Twin: Enabling Technologies, Challenges and Open Research”. In: *IEEE Access* 8 (2020). Also available as arXiv:1911.01276, pp. 108952–108971. DOI: [10.1109/ACCESS.2020.2998358](https://doi.org/10.1109/ACCESS.2020.2998358).
- [29] Daniel González-Toledo et al. “HOM3R: A 3D Viewer for Complex Hierarchical Product Models”. In: *Journal of Virtual Reality and Broadcasting* 14.3 (2017). DOI: [10.20385/1860-2037/14.2017.3](https://doi.org/10.20385/1860-2037/14.2017.3).
- [30] Tovi Grossman and Ravin Balakrishnan. “The design and evaluation of selection techniques for 3D volumetric displays”. In: *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (2006), pp. 3–12. DOI: [10.1145/1166253.1166257](https://doi.org/10.1145/1166253.1166257).
- [31] Sandra G. Hart. “NASA-Task Load Index (NASA-TLX); 20 Years Later”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50.9 (2006), pp. 904–908. DOI: [10.1177/154193120605000909](https://doi.org/10.1177/154193120605000909).
- [32] Haifeng Huang et al. “Chat-scene: bridging 3D scene and large language models with object identifiers”. In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. NIPS ’24. Vancouver, BC, Canada: Curran Associates Inc., 2024. ISBN: 9798331314385. URL: <https://dl.acm.org/doi/10.5555/3737916.3741536>.
- [33] Paulo Ivson et al. “A Systematic Review of Visualization in Building Information Modeling”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.10 (2020), pp. 3109–3127. DOI: [10.1109/TVCG.2019.2907583](https://doi.org/10.1109/TVCG.2019.2907583).
- [34] Jacek Jankowski and Martin Hachet. “Advances in Interaction with 3D Environments”. In: *Computer Graphics Forum* 34.1 (2015), pp. 152–190. DOI: [10.1111/cgf.12466](https://doi.org/10.1111/cgf.12466).
- [35] Donggang Jia et al. “VOICE: Visual Oracle for Interaction, Conversation, and Explanation”. In: *IEEE Transactions on Visualization and Computer Graphics* 31.10 (2025), pp. 8828–8843. DOI: [10.1109/TVCG.2025.3579956](https://doi.org/10.1109/TVCG.2025.3579956).
- [36] Artem Konev et al. “Fast Cutaway Visualization of Sub-terrain Tubular Networks”. In: *Computers & Graphics* 75.5 (2018), pp. 25–35. DOI: [10.1016/j.cag.2018.07.004](https://doi.org/10.1016/j.cag.2018.07.004).

- [37] Regis Kopper, Felipe Bacim, and Doug A. Bowman. “Rapid and accurate 3D selection by progressive refinement”. In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 2011, pp. 67–74. DOI: [10.1109/3DUI.2011.5759219](https://doi.org/10.1109/3DUI.2011.5759219).
- [38] Joseph J. LaViola et al. *3D User Interfaces: Theory and Practice*. 2nd ed. Boston, MA, USA: Addison–Wesley Professional, 2017. ISBN: 9780134034324. URL: <https://dl.acm.org/doi/10.5555/993837>.
- [39] Joong-Jae Lee and Jung-Min Park. “3D Mirrored Object Selection for Occluded Objects in Virtual Environments”. In: *IEEE Access* 8 (2020), pp. 200259–200274. DOI: [10.1109/ACCESS.2020.3035376](https://doi.org/10.1109/ACCESS.2020.3035376).
- [40] Wilmot Li et al. “Automated generation of interactive 3D exploded view diagrams”. In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), pp. 1–7. DOI: [10.1145/1399504.1360700](https://doi.org/10.1145/1399504.1360700).
- [41] Wilmot Li et al. “Interactive cutaway illustrations of complex 3D models”. In: *ACM Transactions on Graphics* 26.3 (2007), 31:1–31:11. DOI: [10.1145/1275808.1276416](https://doi.org/10.1145/1275808.1276416).
- [42] John F. Lucas et al. “Design and Evaluation of 3D Multiple Object Selection Techniques”. MA thesis. Virginia Polytechnic Institute and State University, 2005. URL: <https://hdl.handle.net/10919/31769>.
- [43] Peihang Luo et al. “ChatTwin: Bridging the usability gap to digital twin adoption in infrastructure operations and maintenance with a Natural Language Interface”. In: *Automation in Construction* (2025), p. 106655. DOI: [10.1016/j.autcon.2025.106655](https://doi.org/10.1016/j.autcon.2025.106655).
- [44] Katia Lupinetti et al. “A Web-Based Solution Supporting CAD Assembly Model Exploration and Analysis”. In: *SN Computer Science* 3.1 (2022), p. 19. DOI: [10.1007/s42979-021-00853-9](https://doi.org/10.1007/s42979-021-00853-9).
- [45] Xianzheng Ma, Brandon Smart, Yash Bhalgat, et al. “When LLMs step into the 3D World: A Survey and Meta-Analysis of 3D Tasks via Multi-modal Large Language Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE/CVF, 2024. DOI: [10.48550/ARXIV.2405.10255](https://doi.org/10.48550/ARXIV.2405.10255).
- [46] Henry B. Mann and Donald R. Whitney. “On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other”. In: *The Annals of Mathematical Statistics* 18.1 (1947), pp. 50–60. DOI: [10.1214/aoms/1177730491](https://doi.org/10.1214/aoms/1177730491).
- [47] Mykola Maslych et al. “Toward Intuitive Acquisition of Occluded VR Objects Through an Interactive Disocclusion Mini-map”. In: *Proceedings of the 2023 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2023, pp. 460–470. DOI: [10.1109/VR55154.2023.00061](https://doi.org/10.1109/VR55154.2023.00061).
- [48] Stephane Nzetchou et al. “Semantic enrichment approach for low-level CAD models managed in PLM context: Literature review and research prospect”. In: *Computers in Industry* 135 (2022), p. 103575. DOI: [10.1016/j.compind.2021.103575](https://doi.org/10.1016/j.compind.2021.103575).
- [49] Johanna Renny Octavia, Karin Coninx, and Chris Raymaekers. “Enhancing user interaction in virtual environments through adaptive personalized 3D interaction techniques”. In: *User Modeling, Adaptation, and Personalization*. Springer, 2010, pp. 423–426. DOI: [10.1007/978-3-642-13470-8_45](https://doi.org/10.1007/978-3-642-13470-8_45).
- [50] Johanna Renny Octavia, Chris Raymaekers, and Karin Coninx. “A Conceptual Framework for Adaptation and Personalization in Virtual Environments”. In: *Proceedings of the 2009 20th International Workshop on Database and Expert Systems Applications (DEXA Workshops)*. IEEE, 2009, pp. 284–288. DOI: [10.1109/DEXA.2009.15](https://doi.org/10.1109/DEXA.2009.15).

- [51] Johanna Renny Octavia, Chris Raymaekers, and Karin Coninx. “Adaptation in Virtual Environments: Conceptual Framework and User Models”. In: *Multimedia Tools and Applications* 54.1 (2011), pp. 121–142. DOI: [10.1007/s11042-010-0525-z](https://doi.org/10.1007/s11042-010-0525-z).
- [52] Alain Pfouga and Josip Stjepandić. “Leveraging 3D geometric knowledge in the product lifecycle based on industrial standards”. In: *Journal of Computational Design and Engineering* 5.1 (2018), pp. 54–67. DOI: [10.1016/j.jcde.2017.11.002](https://doi.org/10.1016/j.jcde.2017.11.002).
- [53] Cyprien Pindat et al. “Drilling into Complex 3D Models with Gimlenses”. In: *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*. ACM, 2013, pp. 223–230. DOI: [10.1145/2503713.2503714](https://doi.org/10.1145/2503713.2503714).
- [54] Carole Plasson et al. “A Lens-Based Extension of Raycasting for Accurate Selection in Dense 3D Environments”. In: *Human-Computer Interaction – INTERACT 2021*. Springer, 2021, pp. 501–524. DOI: [10.1007/978-3-030-85610-6_28](https://doi.org/10.1007/978-3-030-85610-6_28).
- [55] Hyeonjoong Ro, Youngho Lee, and Woontack Woo. “A Dynamic Depth-Variable Ray-Casting Interface for Object Manipulation in AR Environments”. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 2875–2880. DOI: [10.1109/SMC.2017.8123063](https://doi.org/10.1109/SMC.2017.8123063).
- [56] Jonathan C. Roberts. “State of the Art: Coordinated & Multiple Views in Exploratory Visualization”. In: *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*. 2007, pp. 61–71. DOI: [10.1109/CMV.2007.20](https://doi.org/10.1109/CMV.2007.20).
- [57] Shuai Shao et al. “An Automatic Generation Method of Exploded View Based on Projection”. In: *Manufacturing Technology* 21.5 (2021), pp. 691–699. DOI: [10.21062/mft.2021.067](https://doi.org/10.21062/mft.2021.067).
- [58] Leixian Shen et al. “Towards Natural Language Interfaces for Data Visualization: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.6 (2023), pp. 3121–3144. DOI: [10.1109/TVCG.2022.3148007](https://doi.org/10.1109/TVCG.2022.3148007).
- [59] Vladeta Stojanovic et al. “Ontology-Driven Analytics for Indoor Point Clouds”. In: *Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*. Vol. 2. 2020, pp. 537–546. DOI: [10.52842/CONF.CAADR IA.2020.2.537](https://doi.org/10.52842/CONF.CAADR IA.2020.2.537).
- [60] Vladeta Stojanovic et al. “Semantic Enrichment of Indoor Point Clouds: An Overview of Progress towards Digital Twinning”. In: *Proceedings of the 37th eCAADe and 23rd SIGraDi Joint Conference: Architecture in the Age of the 4th Industrial Revolution*. Ed. by José Pedro Sousa, Gonçalo Castro Henriques, and João Pedro Xavier. Vol. 2. Porto, Portugal: Blucher, 2019, pp. 809–818. DOI: [10.5151/proceedings-ecaadesigradi2019_051](https://doi.org/10.5151/proceedings-ecaadesigradi2019_051).
- [61] Junwei Sun and Wolfgang Stuerzlinger. “Selecting and Sliding Hidden Objects in 3D Desktop Environments”. In: *Proceedings of Graphics Interface 2019*. Canadian Information Processing Society, 2019. DOI: [10.20380/GI2019.08](https://doi.org/10.20380/GI2019.08).
- [62] Fei Tao et al. “Digital Twin in Industry: State-of-the-Art”. In: *IEEE Transactions on Industrial Informatics* 15.4 (2019), pp. 2405–2415. DOI: [10.1109/TII.2018.2873186](https://doi.org/10.1109/TII.2018.2873186).
- [63] Robert J. Teather and Wolfgang Stuerzlinger. “Factors Affecting Mouse-Based 3D Selection in Desktop VR Systems”. In: *Proceedings of the 3rd Symposium on Spatial User Interaction (SUI)*. 2015, pp. 10–19. DOI: [10.1145/2788940.2788946](https://doi.org/10.1145/2788940.2788946).
- [64] Michael Traoré, Christophe Hurter, and Alexandru Telea. “Interactive Obstruction-Free Lensing for Volumetric Data Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 1029–1039. DOI: [10.1109/TVCG.2018.2864690](https://doi.org/10.1109/TVCG.2018.2864690).

- [65] Lode Vanacken, Tovi Grossman, and Karin Coninx. “Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments”. In: *2007 IEEE Symposium on 3D User Interfaces*. IEEE, 2007, pp. 117–124. DOI: [10.1109/3DUI.2007.340783](https://doi.org/10.1109/3DUI.2007.340783).
- [66] Lode Vanacken, Tovi Grossman, and Karin Coninx. “Multimodal Selection Techniques for Dense and Occluded 3D Virtual Environments”. In: *International Journal of Human-Computer Studies* 67.3 (2009), pp. 180–203. DOI: [10.1016/j.ijhcs.2008.09.001](https://doi.org/10.1016/j.ijhcs.2008.09.001).
- [67] Xiaoyan Wang, Zeju Li, Yifan Xu, et al. “Spatial 3D-LLM: Exploring Spatial Awareness in 3D Vision-Language Models”. In: *Proceedings of the 2025 IEEE International Conference on Multimedia and Expo (ICME)*. 2025. DOI: [10.1109/ICME59968.2025.11208990](https://doi.org/10.1109/ICME59968.2025.11208990).
- [68] Frank Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. DOI: [10.2307/3001968](https://doi.org/10.2307/3001968).
- [69] Dennis Wolf et al. “Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, 2020. DOI: [10.1145/3313831.3376876](https://doi.org/10.1145/3313831.3376876).
- [70] Wenhao Wu, Xinyu Wu, and Xiaonan Dong. “Research on Virtual Simulation Disassembly and Assembly Technology for Water Turbine Units”. In: *Proceedings of the 2024 IEEE 5th International Conference on Networking, Power Systems and Information Fusion (NPSIF)*. 2024, pp. 481–485. DOI: [10.1109/NPSIF64134.2024.10883272](https://doi.org/10.1109/NPSIF64134.2024.10883272).
- [71] Difeng Yu et al. “Deriving Selection Techniques for GUIs based on the Multiple Process Model”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2025, pp. 1–16. DOI: [10.1145/3706598.3713089](https://doi.org/10.1145/3706598.3713089).
- [72] Lingyun Yu et al. “Efficient Structure-Aware Selection Techniques for 3D Point Cloud Visualizations with 2DOF Input”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2245–2254. DOI: [10.1109/TVCG.2012.217](https://doi.org/10.1109/TVCG.2012.217).
- [73] Longwen Zhang et al. “BANG: Dividing 3D Assets via Generative Exploded Dynamics”. In: *ACM Transactions on Graphics* 44.4 (2025). DOI: [10.1145/3730840](https://doi.org/10.1145/3730840).

Appendix A

User Testing Script

A.1 Introduction

Hello, and thank you very much for taking the time to help us today.

My name is João and I am a Master’s student at the Faculty of Engineering of the University of Porto. I am developing a prototype tool that extends NVIDIA Omniverse to help users interact more effectively with dense 3D models of CERN’s accelerator infrastructure.

At CERN, locating specific components in 3D models is challenging because geometric data and operational metadata, such as component identifiers, maintenance tasks, and equipment properties, are scattered across disconnected systems. A particular problem occurs when many components share the same internal identifiers because they represent instances of the same design type. Without structured metadata within the 3D environment, distinguishing between these repeated components requires tedious manual navigation and inspection of raw properties.

The prototype I am evaluating combines a conversational interface with conventional 3D navigation to bridge this gap. This study focuses specifically on how users select and interact with components in dense assemblies, and whether natural-language input improves their ability to locate and identify the correct component compared to using the baseline Omniverse interface alone.

It is important to note that the effectiveness of this approach depends on the quality and completeness of the available metadata and 3D model data. Some inaccuracies or inconsistencies may arise during the conversion from CAD formats to USD, which is outside the scope of this work.

You will work with a 3D model of the PS Booster in NVIDIA Omniverse and complete a small number of component identification and selection tasks under two different interface configurations. There are no right or wrong ways to interact with the tool: we are evaluating the prototype, not you. Please feel free to think aloud as you work, and ask questions at any time. For some parts of the tasks I may not be able to answer directly, as we want to observe how you naturally engage with the system.

A.2 Informed Consent Form

Before we begin, I would like you to read and sign an informed consent form.

A.2.1 Introduction

You are invited to participate in a research study to evaluate the usability of a prototype tool designed to support interaction with dense industrial 3D models of CERN's accelerator infrastructure. This experiment is part of a Master's Dissertation at the Faculty of Engineering of the University of Porto, conducted in collaboration with CERN. The prototype combines a conversational interface with conventional 3D navigation to help engineers locate, select, and inspect specific components within complex 3D assemblies. Your participation will help us understand whether this approach effectively supports realistic component-identification workflows.

A.2.2 Voluntary Participation

Participation in this experiment is entirely voluntary. You are free to decide whether or not to take part. If you choose to participate, you may withdraw at any time during the experiment without any consequences or need to provide a reason. To withdraw, simply inform the researcher, and any data collected from you up to that point will be deleted and not used in the study.

A.2.3 Experiment Overview

You will interact with a 3D model of the PS Booster running in NVIDIA Omniverse. You will perform a series of tasks that involve locating and selecting specific components and retrieving information about them. The tasks will be performed twice, under two different interface configurations, using two equivalent but slightly different task sets to ensure a fair comparison. After completing the tasks, you will answer questions about your experience and provide feedback on the system.

A.2.4 Data Collection

We will record your task answers, completion times, and any clarifications requested during the session. We will also collect demographic information and your feedback on the system. In the second interface configuration, the conversation history with the assistant will be logged automatically.

A.2.5 Data Usage and Confidentiality

All data collected during this experiment will be used solely for research purposes as part of this dissertation. Your personal information and responses will remain confidential and will be anonymized. Only the researcher will have access to the raw data, which will be securely stored and deleted after the study is complete.

A.2.6 Contact Information

If you have any questions about the experiment or your participation, please contact:

João Guimarães Mota, up202108677@edu.fe.up.pt

Master in Informatics and Computing Engineering, Faculdade de Engenharia da Universidade do Porto

A.2.7 Consent

I have read and understood the information provided above. I understand that my participation is voluntary, that I can withdraw at any time without consequence, and that my data will be used only for research purposes. I consent to participate in this experiment, including the collection of demographic data and the provision of feedback for system improvements.

Name: _____

Signature: _____

Date: _____

A.3 Pre-Test Questionnaire

I am going to ask you a few questions to gather some background information.

1. What is your age?
2. What is your gender?
3. What is your role at CERN?
4. How familiar are you with the physical facilities represented in the 3D model (PS Booster)?
(5-point ordinal familiarity scale: 1 = I do not know it at all; 5 = I know the area very well).
5. How much experience do you have with 3D model manipulation and visualization software?
(5-point ordinal experience scale: 1 = Not familiar at all; 5 = Extremely familiar).
6. Have you previously used NVIDIA Omniverse? (Yes / No).
7. How frequently do you use conversational AI tools (e.g., ChatGPT, Claude)? (5-point ordinal frequency scale: 1 = Never; 5 = Daily).

A.4 Tool Orientation

I will now show you how to navigate Omniverse and interact with the 3D model.

The scene displays the spatial layout of the facility and all its components. You can navigate using the mouse: hold the left button and drag to rotate the camera around the scene, use the middle button to pan laterally, and use the scroll wheel to zoom in and out. When you left-click on a component, an Info Window appears on the right side showing its metadata.

Please take a moment to click on a few components, explore the scene, and get familiar with how the interface responds to your interactions.

A.4.1 Interface Configuration H0: Baseline Omniverse

In this configuration, you interact with the standard Omniverse interface. You can navigate the 3D scene freely, click on components to select them, and inspect their metadata in the Info Window. The scene tree on the left shows the hierarchical structure of the model using technical names. A search function is available in the scene tree, but it only matches against internal technical identifiers rather than against the semantic metadata.

To locate a component, you will navigate visually, identify candidates, click to select them, and read their properties from the Info Window. This represents the baseline state of Omniverse without custom extensions.

A.4.2 Interface Configuration H1: Hybrid AI-Assisted

In this configuration, you have access to two additional tools beyond the baseline interface.

The custom hierarchy tree on the left displays the model using human-readable engineering names instead of technical internal identifiers.

The chat assistant on the right allows you to express your intent in natural language. You can use queries such as:

- “Move to [component name] in [area name]”
- “Show me the [component name] with an [task name] task”
- “Isolate the [assembly name]”
- “Reset the view”

Before we start the tasks, please experiment with a few queries to familiarize yourself with how this interface works.

A.5 Task-Specific Questionnaire

You will now complete three tasks twice, once under each interface configuration. Both configurations will use the same task structure, but different tools will be available. Your goal in each task is to locate a specific component and retrieve the information requested.

To prevent answers from the first round influencing the second through memorization, two parallel task sets are used. Both sets follow the same structure and target the same component types, but in different periods or sections of the model. The set assigned to each interface configuration is counterbalanced across participants.

A.5.1 Task Set A

A.5.1.1 Task 1: Disambiguation of Similar Components

Find the 'Flow-meter', part of the 'Straight Section', in 'Period 13' that is associated with a maintenance task. Several similar components are present in this area. The reference card shows the component type and approximate local context. Find the correct component and report the task constraint type associated with it, as shown in the Info Window.

Expected answer: [task constraint type] (multiple choice)

A.5.1.2 Task 2: Connected Components

Identify all components directly connected to this specific 'Chambre a vide' in the 'Ensemble Ligne'. The reference card shows the component in the model. Locate it and identify all components directly connected to it. Report their names as shown in the Info Window or in the scene hierarchy.

Expected answer: [set of connected component names] (multiple choice)

A.5.1.3 Task 3: Component Search with Visibility Management and Disambiguation

Find the 'Internal supporting tube' inside 'Multipole' in 'Period 3' that has an installation task scheduled. The reference card shows the component, the general location of the parent assembly and the surrounding context. The target is partially occluded by surrounding structures, and some visually similar components are present nearby. Find this component and report the duration of the installation task as shown in the Info Window.

Expected answer: [task duration] (multiple choice)

A.5.2 Task Set B

A.5.2.1 Task 1: Disambiguation of Similar Components

Find the 'Flow-meter', part of the 'Straight Section', in 'Period 5' that is associated with a maintenance task. Several similar components are present in this area. The reference card shows the component type and approximate local context. Find the correct component and report the task constraint type associated with it, as shown in the Info Window.

Expected answer: [task constraint type] (multiple choice)

A.5.2.2 Task 2: Connected Components

Identify all components directly connected to 'Tube SPS' in the 'Ensemble Ligne'. The reference card shows the component in the model. Locate it and identify all components directly connected to it. Report their names as shown in the Info Window or in the scene hierarchy.

Expected answer: [set of connected component names] (multiple choice)

A.5.2.3 Task 3: Component Search with Visibility Management and Disambiguation

Find the 'Internal supporting tube' inside 'Multipole' in 'Period 1' that has an installation task scheduled. The reference card shows the component, the general location of the parent assembly and the surrounding context. The target is partially occluded by surrounding structures, and some visually similar components are present nearby. Find this component and report the duration of the installation task as shown in the Info Window.

Expected answer: [task duration] (multiple choice)

A.6 Post-Test Questionnaire

After completing all tasks under both configurations, you will answer the following assessments and questions.

A.6.1 NASA Task Load Index

For each interface configuration, please rate the following six dimensions by marking a point on each scale. The scales range from very low on the left to very high on the right.

1. **Mental Demand:** How mentally demanding was the task?
2. **Physical Demand:** How physically demanding was the task?
3. **Temporal Demand:** How hurried or rushed was the pace of the task?
4. **Performance:** How successful were you in accomplishing what you were asked to do?
5. **Effort:** How hard did you have to work to accomplish your level of performance?
6. **Frustration:** How insecure, discouraged, irritated, stressed, and annoyed were you?

The overall workload score is computed as the average of the six ratings on a 0-100 scale.

A.6.2 System Usability Scale

For each interface configuration, please rate the following statements from 1 to 5, where 1 means "strongly disagree" and 5 means "strongly agree".

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

A.6.3 Comparative Preference

Now that you have used both interfaces, please answer the following:

1. Which interface made you feel more confident that you had selected the correct component? (Manual interface / Chat-assisted interface / No difference / It depended on the task).
2. Which interface did you find easier to use? (Manual interface / Chat-assisted interface / No difference / Not sure).
3. If both approaches were available, how would you prefer to work? (Mostly manual / Mostly conversational / Mix of both / Not sure).
4. How predictable did you find the chat-assisted interface? (5-point ordinal predictability scale: 1 = Very unpredictable; 5 = Very predictable).

A.6.4 Open Feedback

1. What were the most difficult aspects of locating and selecting components in the model?
2. Were there situations where the conversational interface was particularly helpful or unhelpful?
3. Do you think a tool like this would be useful for your work at CERN? Why or why not?
4. Do you have any other comments, including recommendations for improvement or feature suggestions?

Closing

Thank you very much for your time and thoughtful feedback. Your responses help us understand whether this approach can better support engineers working with CERN's complex 3D models. If you have any further thoughts or questions after today, please feel free to reach out.