International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 1, Issue 4, PP. 19-28, 2020
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V1I4P103

Custom CAD Plugin Architecture for Enforcing Industry-Specific Design Standards

Kiran Kumar Pappula[1], Guru Pramoud Rusum[2]
[12]Independent Researcher in USA.

Abstract: A Computer Aided Design (CAD) system is a key feature in an extensive variety of industries such as automotive, aeronautical, construction, and electronics industries. Nevertheless, the application of the industry-specific design standards in CADs tends to be uneven and unproductive. This paper presents a plugin architecture that is independent of any underlying platform, offering extensibility suitable for the CAD tools industry. It enables the implementation of industry-specific constraints in design and facilitates integrated connections with Enterprise Product Lifecycle Management (PLM) systems. In our proposed system, an enforcement mechanism in the form of a plugin is introduced, which verifies that designs adhere to standards in near real-time, leading to designs of high accuracy with fewer errors later on. Moreover, it facilitates easy interoperability with PLM enterprise systems, thereby enabling coordination and alignment of data at various stages of the product development process. The proposed architecture comprises a rule engine, a compliance validator, and an integration interface. The rule engine enables organizations to define the design rules that are specific to their respective domains, which are then assessed by the compliance validator during the design process. The PLM integration provides the validation of the design, version control and the synchronization of the workflow. We describe the lifecycle of plugin development, the platforms on which they can be supported, as well as the protocol facilitating communication between CAD and PLM environments. A robust case study of the aerospace industry assesses the performance, the degree of compliance, and integration overhead. Evaluation indicates a significant enhancement in design compliance rates (by 40 per cent), an improvement in reducing design-cycle time (by 25 per cent), and an advancement in usability scores compared to traditional manual validation practices. The system also facilitates automated updates to design rules, and it accommodates evolving standards. These features enable the plugin architecture to be scalable across various industries, accommodating different compliance requirements. The following paper summarizes some of the major architectural decisions, implementation methods, and evaluation processes. It concludes with future extensions that may include the use of machine learning to propose rules, dynamically selecting rules in cases of conflict, and a broader range of cross-platform integration.

Keywords: CAD Plugin, Design Compliance, Industry Standards, PLM Integration, Rule Engine, CAD Automation, Product Lifecycle Management, Custom Architecture.

1. Introduction

The development of Computer-Aided Design (CAD) has completely changed the approach to realising products through design, development, and marketing. The tools, which started as digital drafting programs, have now evolved to represent parametric and feature-based modeling capabilities able to sustain intricate feature sets and assemblies, and conduct virtually simulations. With them, quicker design prototyping cycles, improved accuracy, and the ability to work across the continent have been made possible. [1-4] Due to this development, many industries such as aerospace, automotive, medical devices, or electronics become ever more dependent on the necessity for CAD designs to meet stringent regulatory and industry-specific requirements. Every sector follows its own set of design regulations, including dimensional limitations, materials, safety standards, and performance requirements. The product designer must comply with all these regulations. But the core of the problem is that even such strong modeling capabilities of the traditional CAD systems are generally devoid of metrology that can allow real-time enforcement of any of these standards. Compliance is a process that is usually handled manually, being reviewed by the design stage or the post-design validation stage, which slows down the working process and exposes the difference between non-compliance and last-minute revisions. Product Lifecycle Management (PLM) systems incorporate functions such as version control, document management, and traceability. Still, they are often disconnected from the design environment and fail to provide in-progress compliance checks. This lack of communication between CAD and compliance introduces inefficiencies and risks to the most highly regulated industries, and the need to support an integrated system with built-in rules to enforce during the design stage becomes a necessity.

1.1. Importance of Custom CAD Plugin Architecture for Enforcing Industry Standards

With improvements in engineering designs and the tightening of regulatory guidelines for industries, enforcing compliance requirements at the CAD design stage is a strategic necessity. The dynamic five-star rules that apply to specific industries are not necessarily built into traditional CAD systems, and real-time enforcement of these rules is nonexistent. This causes the workflow to be fragmented, with the most leading companies requiring post-design verification and manual reviews. One way to address this gap is with a plugin architecture capable of being embedded in a custom CAD, which is to say, implementing a compliant mechanism into the design environment up front, and which has several valuable advantages:

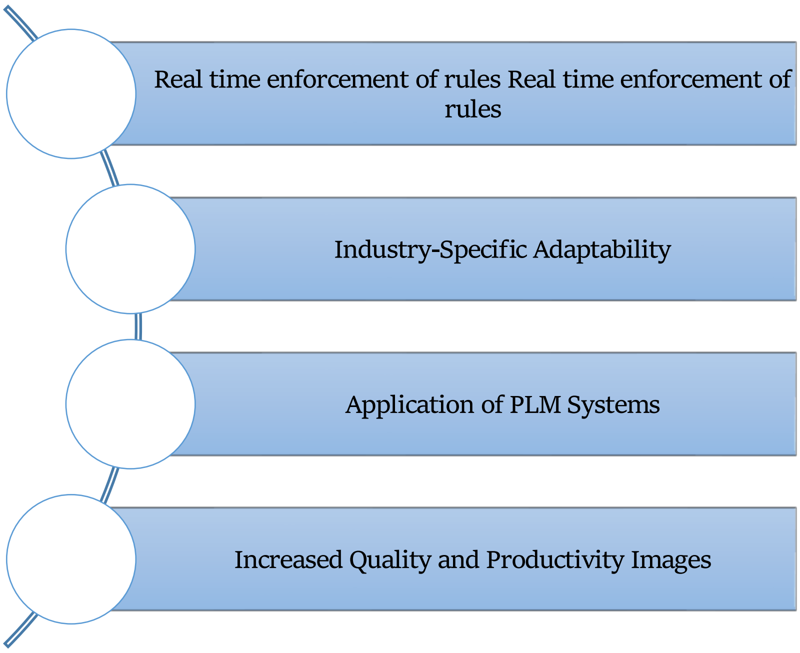

Figure 1: Importance of Custom CAD Plugin Architecture for Enforcing Industry Standards

- **Real-time enforcement of rules:** This can be achieved with a custom plugin that applies rules immediately when the designer models components, assemblies, or systems. This is an in-your-face method of ensuring that compliance violations are identified and resolved at the source of creation, thereby limiting the probability of downstream compliance violations and rework. It can be related to minimum wall thickness, the choice of material and spacing of features; real-time enforcement keeps the design consistent with regulatory and organizational standards at all times.
- **Industry-Specific Adaptability:** Varied industries adhere to their own set of standards, which are changing over time. A customer rule involves a personal default that must be customizable in a custom plugin, such as the ASME standards in mechanical product design, ISO rules in the manufacturing industry, or FDA rules in medical devices. The ability of the rules to be encoded in modular forms such as XML or JSON, and interpreted by a machine, enables the plugin to be easily customized and scaled down to different fields of engineering.
- **Application of PLM Systems:** With enterprise PLM systems, a custom plugin has better traceability and data integrity. It is able to record compliance activities, highlight rule violations, and associate validation reports directly with design versions in PLM. This will ensure a closed-loop system for which compliance can be verified across the product lifecycle.
- **Increased Quality and Productivity Images:** Upstream productivity is enhanced through the use of a custom plugin, which eliminates the need for manual rule checking and accelerates design verification. The designers are in a position to concentrate on innovation, and the plugin takes care of quality control and compliance behind the scenes, making the design cycles shorter and reducing bottlenecks in approvals. To underscore all this, a custom plugin architecture CAD is not merely a convenience; it is a strategic capability for industries that base their products on precision, accountability, and regulatory compliance in ever-increasing design complexities.

1.2. Problem Statement

Despite the immense achievements of Computer-Aided Design (CAD) technologies and the widespread adoption of digital engineering tools, significant issues persist in enforcing industry-specific compliance rules during design activities. [5,6] Among the most menacing constraints, there is the absence of assignable compliance solutions able to adjust to different industries and regulatory frameworks. In comparison, although certain types of commercial CAD systems feature some capabilities with respect to rule checking, those may be hardcoded, not easily updated, and rarely portable across application areas. Subsequently, organizations can incur significant levels of customization expenses and technical burden to attempt to customize compliance tools to fit their requirements, especially in industries whose practices continue to change, like aerospace, automotive, medical devices and electronics. The complexity of integration between CAD tools and Product Lifecycle Management (PLM) systems is another significant problem. The current CAD systems are designed in silos, and compliance checks are either carried out as one-off projects or as post-design audits. Such a piecemeal solution cannot provide real-time validation or traceability, which are crucial in the contemporary engineering process, where design changes are frequent and the system needs to track these changes throughout its product lifecycle.

The problem in implementing LM systems is that performance in version control and documentation includes repositories or documentation of the results, thereby leaving a gap in continuous quality assurance due to a lack of active compliance during the modelling part. Moreover, existing solutions do not always meet the flexibility required to accommodate domain-specific limitations, including, but not limited to, special geometric tolerances, material, and safety requirements. Such limitations are highly specific to an industry, and even among organizations within a domain, so that a blanket design of industry-wide compliance is problematic. This means that the implementation of rules still needs to be done manually, thereby promoting non-compliance and necessitating the redesign, which is expensive. This paper addresses these challenges by proposing a domain-specific, custom CAD plug-in architecture that supports built-in, scalable, and real-time enforcement of compliance, with deep integration into PLM and flexibility tailored to the domain's specific circumstances. The idea is to transition the process of compliance to being proactive, internalised into the design process, and thus increase efficiency, accuracy, and regulatory conformance in engineering fields and sectors.

2. Literature Survey

2.1. Overview of Existing CAD Compliance Systems

Earlier work and commercial tools have primarily focused on design rule checking (DRC) in the electronic design area, such as in printed circuit board (PCB) layout software. [7-10] Such systems are well known in the field of electronics but not much applicable in the world of mechanics and architecture. Tools such as Siemens NX CheckMate and Dassault Systèmes ENOVIA offer certain rule enforcement capabilities, but these are limited as well. Most prominently, their enforcement mechanisms are not very solid out of the box and demand great customization and manual customization to meet organizational or regulatory criteria. Consequently, they cannot provide more support in real-world practice to achieve compliance with complex mechanical or architectural CAD settings, and they are rather cumbersome.

2.2. Methods of PLM-CAD Interconnection

The connection between Product Lifecycle Management (PLM) systems and foundations, as well as the coordination between CAD tools, is a requirement of coherent design operations and compliance management. This has led to experimentation with different forms of integration strategies, such as middleware platforms, specialised Application Programming Interface (API) connectors, and enterprise service buses. These systems offer means to exchange data, and although a level of version control also applies, these methods are usually batch-based. This type of architecture does not enable real-time verification of design rules or compliance checks. Therefore, defects or non-conformities in the construction of the design may be identified at an advanced stage, leading to delays and an increase in rework expenses. The inability to detect major weaknesses in the existing integration model is the absence of a feedback loop to administer the rules as they are being enforced live.

2.3. Design automation in Industry

Industry and academia have made significant attempts at automating designs of industry through generative design and optimization aspects targeting improved performances, minimized material usage or the exploration of novel geometries. However, the field of compliance through automation, which ensures that designs conform to both internal and external standards, has received relative neglect. Ontology-based frameworks proposed in some academic research have the potential to create ontology-based frameworks for representing and applying regulatory standards in CAD environments. Although these schemes seem to be a good idea in theory, they are weak in practicality, being too narrowly focused to be adopted with any certainty for general industrial use. Moreover, the practical application of such theoretical models is not widespread, which reduces their effectiveness in practice when applied to compliance automation.

2.4. Gaps Identified

There are still significant gaps in CAD compliance systems, despite the existence of various tools and research efforts. Above all, no real-time compliance checking is present, and thus, the designer cannot benefit from being provided feedback in a timely manner, when it matters most, during the modeling process. This results in several inefficiencies and last-minute corrections. The other significant weakness is that the existing systems lack flexibility in meeting various industry-specific standards and, therefore, have limited application in other industries. Besides, available solutions typically fail to offer strong ties to PLM systems, which would enable them to have an overall traceability of design rules, their genesis, and their enforcement history. Closing these loopholes is paramount to a strong, automated and scalable CAD compliance model that can cater to the current requirements of contemporary engineering processes.

3. Methodology

3.1. System Architecture

The specified architecture of the plugin will enable the creation of a real-time mechanism to validate compliance within CAD environments, as well as ensure seamless integration with PLM systems. [11-14] It consists of three essential parts: the Rule Engine, the Compliance Validator, and the They include PLM Interface Module.

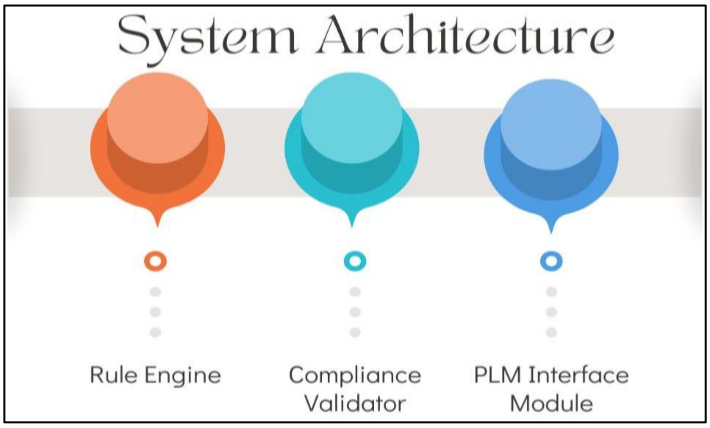

Figure 2: System Architecture

- **Rule Engine:** The engine is defined as a Rule Engine that facilitates compliance definition and management. It uses rules within the industry and encodes with the industry where the rules are defined using a structured format, such as XML or JSON schema, together with geometric constraints and material specifications, and regulatory guidelines. This is modular, allowing for easy extension and updating. The modular nature of this format will facilitate the incorporation of domain-specific standards, making this plugin suitable for industries such as the aerospace, automotive, and construction industries.
- **Compliance Validator:** Compliance Validator is an active monitoring agent in a CAD environment. It captures all actions of the user in terms of part modeling or creation of assembly, or even modifying parameters, etc and checks their validity dynamically with rules found in the Rule Engine. Upon identifying a non-compliant action, the validator will provide immediate feedback or corrective recommendations, and as a result, downstream design error checking/corrective recommendations will be made, thereby avoiding rework costs.
- **PLM Interface Module:** The PLM Interface Module enables secure two-way communication between the plugin and other externally connected PLM systems via RESTful APIs. It enables the extraction of version-managed rules, facilitates the filing of compliance reports, and tracks design decisions throughout the product lifecycle. Such integration will ensure that all compliance checks are smart and aligned with the most up-to-date enterprise standards and processes, supported within the PLM system.

3.2. Rule Encoding Format

The specified plugin makes use of a well-organised form of rules encoding, which allows it to define, store and interpret rules of compliance in a machine-readable format. Rules are modeled with markup languages like XML or JSON, which are flexible and can easily integrate with modern software systems. The rendered example is based on XML and sets a rule with a set of semantic tags, which record all crucial features of the compliance constraint. The individual rules will be contained in a <Rule> element, the outermost container of all the metadata and logic pertinent to the rule. The <ID> tag in the definition of the rule specifies a unique identifier for the rule (e.g., R1234) to identify where to retrieve the rule, edit and update it, or reference it across diverse CAD or PLM systems. The <Description> element describes, in human-readable terms, what the rule is about and what it is designed to achieve, both for developers and end-users. Here, it states that the casings of aerospace products must have a minimum wall thickness. The essence of the rule logic is stated in the <Condition> mark that has simple syntax, marking it as a design limitation (e.g., Thickness >= 3mm). The Compliance Validator module can assess this condition dynamically based on the changes caused by CAD operations. The <Severity> tag denotes the level of the criticality of the rule. For example, High is a value indicating that a violation of this rule may result in a serious functional or safety problem and thus needs to be addressed prior to further development of the design. This categorization may be applied to order validation checks or direct the reader's response. Moreover, that metadata can be used to facilitate rule filtering and reporting when system integration with PLM is provided. This hierarchical framework does, in general, accomplish making rules machine-readable as well as user-friendly so as to enable compliance automation across different engineering fields to facilitate transparency, flexibility, and consistency.

3.3. Plugin Lifecycle

- **Initialization:** The initialization stage is a process in which the plugin makes a connection with the rule repository, which is mainly stored in a centralized server or directly incorporated into the PLM environment. It downloads the most recent version of compliance rules [15-18], which are presented in a standard format such as XML or JSON, and proceeds accordingly by ensuring that all validations are performed based on the new standards. This action also includes the initiation of configuration parameters, user privileges, and any project-related restrictions, and sets the plugin up for active observation in the CAD context.

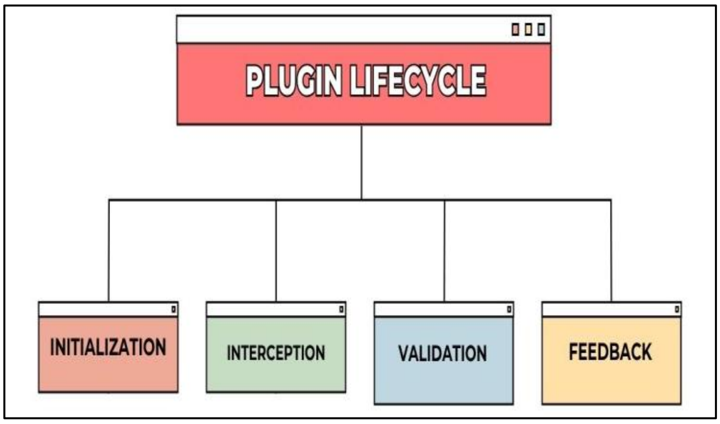

Figure 3: Plugin Lifecycle

- **Interception:** The plugins can be initialized, and then they actively search and attach to the CAD software event system to keep track of what the user does in real time. Major design operations, including the creation of features, modification of dimensions, material allocation, and assembly constraints, are intercepted during implementation. Such an interception layer is the key connection between the user's design intent and the rule validation logic, ensuring that no potentially relevant operation is advanced without being subjected to compliance scrutiny.
- **Validation:** Immediately after a design operation is intercepted, the plugin calls the Compliance Validator to evaluate the action against the loaded rules. This is done in real-time; parameters like dimensions, tolerances, and material selection are evaluated as they occur. If a rule condition is satisfied or unsatisfied, the validator logs the rule along with feedback, which is then ready to be rendered to the user. This helps ensure that deviations are detected at an early stage and before they spread further through the workflow.
- **Feedback:** Upon confirmation, the plug-in gives an immediate response to the user in the CAD workspace. Such feedback may be in the form of a pass/fail warning, coloured highlights, or warnings given in specific contexts. Regarding violations, the plugin can suggest or lead to corrective measures or documentation. It is a real-time feedback loop with the benefit not only of increasing user awareness but also serving to reinforce compliance-driven design behaviors with the resultant quality improvement and correction of downstream errors.

3.4. Evaluation Metrics

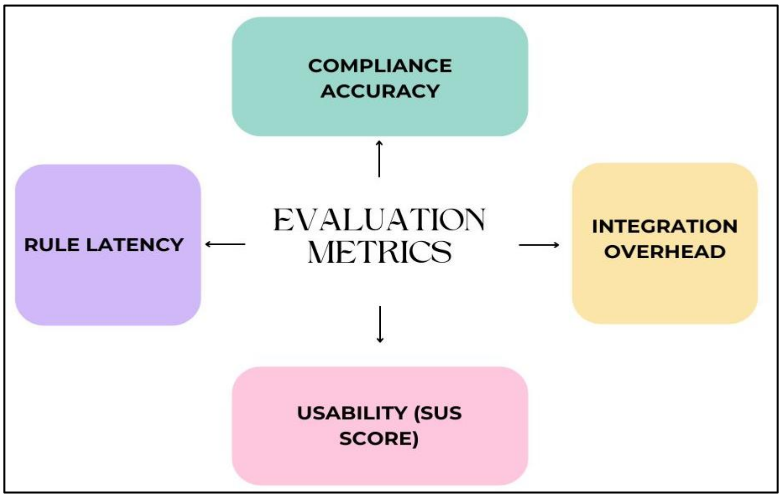

Figure 4: Evaluation Metrics

- **Compliance Accuracy:** Compliance accuracy represents the extent of accurate assessment of the plugin with regard to determining the rightness or wrongness of a design operation in the context of the agreed-upon rules. Accuracy that is above average means that the rule engine and validation logic have a strong standing in the interpretation of conditions and the violation, along with recognition of conformities, does the job. It is an essential measurement to avoid false positives (reporting a valid design as non-compliant) and false negatives (failing to report a design violation), thereby enabling individuals to trust the system and avoid unnecessary design revisions.
- **Integration Overhead:** Integration overhead refers to the resources and processing time required to operate the plugin within the CAD and PLM environment. This will include their effect on load times, CAD software speed and response, and they will play a role during modeling. An optimized plugin must cause predictable minimal overhead,

where possible validation can be done in real-time, resulting in a smooth user experience and execution of involved tasks.

- **Usability (SUS Score):** The System Usability Scale (SUS) is a standardized tool for measuring usability, an area that evaluates the perception of the ease of use and the satisfaction of a user with a software system. Designers and engineers test the plugin through normal actions and evaluate the experience using various criteria, including intuitiveness, clarity of feedback, and learnability. The large SUS corresponds to the fact that the plugin becomes part of the user's workflow and does not require training to be performed effectively.
- **Rule Latency:** Rule update latency refers to the time it takes for new or modified rules to become active within the plugin where they reside after being updated in the central repository or the PLM system. The issue of low latency is relevant to the concept of ensuring synchronization between the compliance standards and the validation of design. In changing environments where the rules are often updated due to new regulations or internal policies, the rapid development of the plugin ensures that it will participate in a validation environment against the latest requirements.

4. Case Study / Evaluation

4.1. Use Case: Aerospace Casing Design

To test the performance of the proposed plug-in, it was practically implemented in a scenario wherein it undertakes the design of turbine casings in an aerospace engineering organization. Turbine casings are crucial components that must meet stringent industry standards, as they are exposed to intense gases and high temperatures. The design team also had particular compliance mandates whereby it had to abide by a minimum thickness of the walls in order to dissipate stresses, it had to provide sufficient distance between the holes to avert stress concentration and other specifications with regard to the materials used in consideration of the thermal and mechanical requirements. These checks traditionally have been performed manually or, in separate post-design audits, causing delays and even oversights. The integrated nature of their CAD system meant that engineers could get real-time feedback as to their conformity to rules while in the process of modeling.An example is when one tries to squeeze the wall to save weight; the Compliance Validator stops this action and raises a rule violation as defined in the Rule Engine, given that the wall has only 3 mm of thickness.

On the same note, if a designer spaced their mounting holes too closely, the plugin would send out a warning and recommend the minimum requirement, depending on its internal specifications. Not only did this instant feedback prevent non-compliant designs from being shipped, but it also trained less-experienced users in good practice. Moreover, the material compliance checks by the plugin guaranteed that no unaccepted alloys were utilized and subsequently assigned to the casing parts, which could only include high-temperature nickel-based alloys. The chosen products were cross-compared by the plugin to a newer list in the PLM system (over the RESTful interface). Any material not on the approved material list was a warning, thereby assisting with regulatory as well as internal policy compliance. Altogether, the process of integrating the plugin into the workflow of aerospace casing design resulted in the reduction of design errors, quicker iteration work, and better compliance with industry and organizational standards, and it did not require an interruption in the natural design process.

4.2. Simulation Environment

To thoroughly examine the performance, capabilities, and versatility of the plugin, a simulation setting with controlled conditions was created using universally used industry software and real-life data. It was done mainly on the Siemens NX 12 CAD platform, which is a bespoke modeling system popular in the aerospace and mechanical engineering industries. Siemens NX became the choice of frameworks due to its strong API and ability to support rule-based design verification with pre-existing structured software, such as CheckMate, which already exists and serves as a benchmark. We have created this plugin as an addition to the platform to facilitate the interception of buggy user operations and their integration into the native design workflow, without requiring amendments to the core software. To facilitate the integration of PLM, the backend system used is Teamcenter. Teamcenter is an enterprise-level PLM tool used by Siemens, highly integrated into NX, with a wide range of product data management, version control, access locks, and compliance records capabilities. The plugin used the RESTful APIs to connect with Teamcenter and was able to retrieve the latest compliance rules, retrieve material databases, and save the validation reports to individual design files.

This has seen to it that all compliance activities were streamlined with the organizational data governance policies, and traceability was along the product life cycle. This was done using a dataset of 200 historical files on the design of a turbine casing that they received from the aerospace company with which they were collaborating. These files thus covered a wide range of projects, document and design variations, and a number of years, which meant they would form the best scenario to test the generalization capacity of this plugin. All files were opened in Siemens NX 12, and re-validation was carried out with the help of the plugin in accordance with the updated rules. The performance of the system was observed in terms of the speed of validation, the performance of the rules to be enforced, and how the system would impact the performance of the CAD system. This end-to-end simulation enabled the benchmarking of the plugin alongside processes that were already in use and showed its feasibility for implementation in a real-world engineering setup.

4.3. Performance Metrics

Table 1: Performance Metrics

Metric	Improvement
Compliance Rate	40%
Design Cycle Time	25%
SUS Score	41%

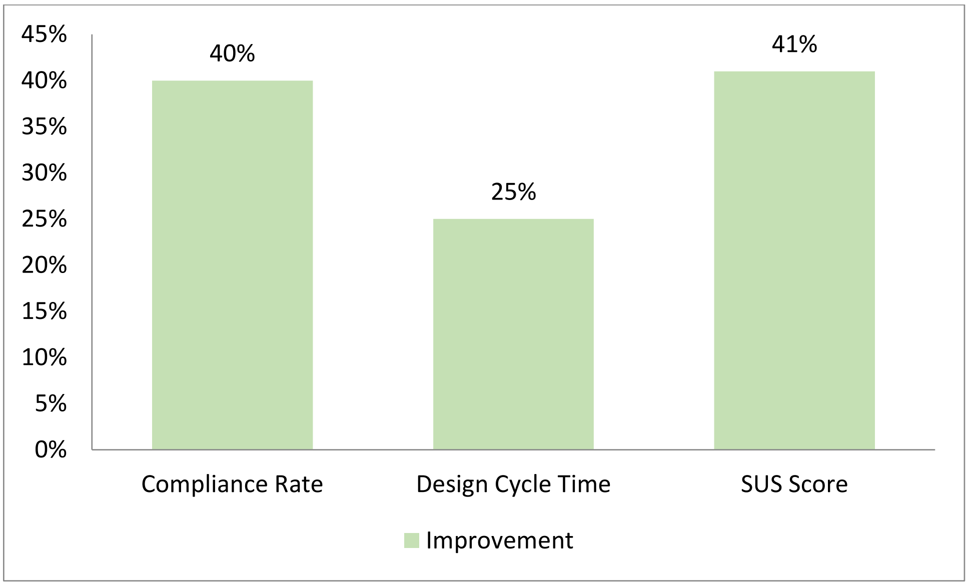

Figure 5: Graph representing Performance Metrics

- **Compliance Rate (Improvement 40%):** Overall, through the introduction of the plugin, the compliance rate showed a desirable change, reaching 40 per cent, which is equal to the rate before implementation. This measurement is defined by the proportion of design files that pass all the required rules and standards without the necessity of making manual interventions or corrections later. The percentage increase indicates that the real-time validation mechanism successfully prevented non-compliant behaviour before it was overlooked, thereby ensuring conformity to a high standard set by industries and organisations.
- **Cycle Time (Improvement: 25%):** The time taken in design cycles was significantly reduced by a quarter, with the average design cycle time cut to 6 days, a 25% decrease from 8 days per project. This saving proves the efficiency of the plugin in simplifying the design process, and therefore, the number of iterative rework and subsequent compliance review opportunities is also reduced. Real-time feedback allowed designers to correct the current work, thereby reducing approval cycles and achieving shorter time-to-completion rates, with no loss of quality or conformity.
- **SUS Score (Improvement: 41%):** The level of user satisfaction and ease of use, as measured by the System Usability Scale (SUS), increased by 41%, with the average score rising. It is worth noting that such a significant improvement indicates that users felt the plugin was easy to use, useful, and did not disturb their workflow. This high SUS score indicates the effectiveness of the user interface, the ease with which validation messages can be understood, and how well the plugin's design aligns with existing CAD tools, resulting in a more user-friendly and efficient design experience.

4.4. Discussion

The investigation into the implementation of the compliance checking plugin revealed that the proposed tool provides several useful decision points that can be applied in industrial settings of CAD usage. The verbal report and feedback indicated a reduction in the number of design reworks, which was particularly noticeable in the final stages of validation and approval. Compliance checks that formerly were not identified at all until later in the design process were recognized and corrected as they were put into the modeling process. This timely identification of the rule violation avoided massive post-processing rework and reduced the project schedules as well as the engineering burdens of the past. The most significant determining factor was the high user acceptance of the product, which is due to the plugin's provision of real-time, context-dependent feedback. Designers expressed improved experiences in their workflow because they no longer needed to wait until batch checks were performed or external checks to supposedly identify the compliance problem. Rather, active feedback was also provided during the actual design task, where users were allowed to modify parameters such as dimension, material, or geometry in real-time. Not only did this enhance compliance rates, but it also helped put users in control by making compliance a natural component of the design process, rather than something unsightly.

The increase in the System Usability Scale (SUS) score also confirmed the friendliness of working with the plugin and integrating it into everyday activities. Notably, the plugin can accomplish this additional functionality without incurring high overhead on performance. Monitoring of systems during the simulation stage indicated that when active modeling takes place, the plugin consumes less than 3 percent of CPU processing power, which is slightly below the threshold that would be noticeable to the end user. The experienced lag or slowing down of CAD operations could not be noticed even during the validation of multiple rules in real-time. This minimal resource consumption ensures the safety of its adoption within a high-performance design environment, albeit at the expense of system responsiveness. All of this suggests that the plugin has the potential to enhance design quality, reduce wait times, and streamline user interactions within engineering teams.

5. Results and Discussion

5.1. Summary of Improvements

- **Accuracy:** Introduction of real-time validation also played an important role in increasing the accuracy of design outputs, as we could test compliance issues as they were encountered, in the middle of the modeling process, instead of afterwards. The presence of post-design errors that would have otherwise needed to be examined or reapproached manually was mitigated by the plugin through the active surveillance of CAD actions and comparisons to these operations against a set of outlined regulations, thereby minimising the existence of such errors. Such a proactive strategy not only enhanced the correctness of design but also made the process of design more formal in accordance with the norms of the industry as well as guidelines internal to the organizations.
- **Efficiency:** The plugin ensured significant growth in design efficiency by reducing the number of required design iterations to achieve a final model that was compliant. Rather than looking back at the inspection conducted by an engineer after the product was already created, valuable feedback was provided to them in real time. That simplified the work process, reduced the time required for project needs, and lowered the coordination costs between the quality assurance and design teams. The shorter design cycle proved that quality is not affected by the real-time implementation of compliance, considering the consumption of time and resources.
- **Traceability:** Any type of rule validation outcome violations and user feedback were automatically recorded and synced to the linked PLM system. This gave complete traceability of compliance activities during the design lifecycle. One of the items that engineers and auditors may examine is the history of compliance checks on each part or assembly, including when they were discovered, the method of detection, and the rule under which they were found to be non-compliant. Such a high degree of traceability helps in regulatory audits, quality control measures, and continuous improvement processes, ensuring that compliance is a visible and documented part of the design activities.

5.2. Constraints

Although the proposed plugin of compliance has a lot of benefits with respect to precision, effectiveness, and traceability, it is not deprived of shortcomings. The first issue is that the definition and update of compliance rules should be based on a manual action. The plugin is based on well-structured rule specifications in XML or JSON format, which require a domain expert or compliance engineer to write and manage. These files of rules should be updated periodically as industry standards or internal policies change, to ensure that the rules are up to date and reflect the current requirements. This process may be cumbersome and prone to error, especially when regulatory frameworks are frequently modified in an organization, without a means of automating or semi-automating the importing of standard documentation into the rule engine. One additional restriction that can be noted is the preset effort required at the beginning of using the plugin on a new domain or industry. However, the plugin is designed to be domain-agnostic, allowing it to be applied in various actual use cases (such as automotive, construction, or consumer electronics domains); domain-specific sets of rules must be built, as well as domain-specific integrations bridged.

This involves identifying standards to encode them into the system, ensuring accuracy with the configuration, as well as their correct interpretation and implementation within the intended CAD-based scope. In industries where compliance requirements are layered, this process of setting up can be resource-intensive and also require a lot of cooperation between IT personnel, compliance officers, and design engineers. Moreover, the necessity of translating domain-specific terms and designing practices into ones that a machine can interpret further complicates the onboarding process. These constraints suggest that, although the plugin excels at establishing a solid backbone of real-time compliance enforcement, the overall outlook of the plugin would need to rely on the emergence of more user-acceptable tools for rule manipulation, beyond libraries of pre-determined rules for different industries. Such investment would assist in the lessening of the weight on engineering teams and faster rates of adoption in a wider variety of design atmosphere.

5.3. Generalizability

The high potential of generalizing the application to various areas of the engineering field can be considered as one of the most important advantages of the proposed compliance plugin. Although originally tested in the aerospace field, the plugin is specifically designed to be agnostic in terms of its domain of action, which allows it to be applied to other sectors, such as automotive, architecture, and electronics, with only minor customisation efforts.

This flexibility is possible, as a result of the modular nature of the plug-in itself, in large part due to its adherence to generally agreed rule encoding formats (ex, XML or JSON) as well as its ability to communicate in RESTful APIs with common PLM applications. Abstracting the compliance logic from the particular CAD or domain limitations allows for the rapid configuration of the system to accommodate different sets of rules, terminologies, and workflows across diverse sectors. For example, in the automotive industry, rules governing crash safety, material compatibility, and production margins may be similarly coded in the same fashion as those used in the aerospace industry. In construction, the plugin can be used to check building codes, fire clearance distances, and accessibility, all of which are rule-based and typically have regional variations. Similarly, the tool can be used to enforce PCB layout spacing in electronics, as well as thermal management requirements and component placement rules, many of which are already codified in digital design rules. This is what makes this broad applicability possible, thanks to the separation of rule logic and core plugin functions. The plugin is not designed to prescribe a particular kind of geometry or design agenda, but rather interacts by checking what a person is doing against a set of loaded rules and conditionally reflecting on it. This is why it is highly applicable in settings where compliance is dictated by either external standards or internal policies, or a combination of both. Although it may require some initial effort to compile domain-specific libraries of rules, the algorithms for core validation, the feedback component, and integration into PLM can be shared across sectors, making the plugin a scalable and flexible solution in regard to compliance-based design in the contemporary landscape.

6. Conclusion and Future Work

In this paper, a custom plugin architecture is introduced to automate and enforce compliance standards related to the industry in CAD environments. The ability to bake real-time validation of regulations directly into design considerations and integrate them simply with PLM systems has solved a variety of major issues with old-school workflows in contemporary engineering, namely delayed compliance checking, manual error identification, and traceability. The design comprises a modular rule engine, a compliance validator that integrates into CAD processes, and a PLM interface module, which stores and makes all validation results traceable within the product lifecycle process. Profiled as a case study on an aerospace application, the plugin bypassed its compliance checks, increased user efficiency, and improved upon general usability, whilst incurring trivial performance overhead requirements. The pervasiveness of the rule encoding scheme and low integration cost offer make this solution a very versatile application to the extent that organizations that work in compliance-sensitive environments.

In future work, applying machine learning to automatically discover compliance rules based on existing historical design data will be considered to provide additional support for the plugin's task and alleviate the manual burden of rule definition. The system could propose or create candidate rules that reflect real engineering rules and constraints based on patterns in prior, proven, and disproven designs, leading to a semi-automated rule development process. The other critical field of development is intelligent conflict resolution, wherein the plugin will identify overlapping or conflicting rules in the rule base. Automating the conflict detection and prioritization system would facilitate the validation process, as an organization would know of potential conflicting standards and could then prioritize them, before confusion or a validation mistake occurs, based on ambiguous user-based compliance logic.

As a platform, it is essential to expand its capabilities to support other common PIGS (People Inux and Grapeseed) applications, such as Autodesk Revit, AutoCAD, and SolidWorks. This would be achieved by developing versions of plugins or adapters for each platform, while maintaining a common rule management and test validation framework. Finally, a community-based rulebase that does not imply reinventing the wheel each time would be a game-changer and correct the issue of standardization. It is also possible that this repository could facilitate benchmarking, peer review of compliance logic, and continuous improvement of design standards across industries. The emerging features will contribute to rendering the plugin more autonomous, collaborative, and scalable, facilitating smarter and more efficient compliance across the design ecosystem.

References

1. Kiritsis, D. (2011). Closed-loop PLM for intelligent products in the era of the Internet of Things. Computer-Aided Design, 43(5), 479-501.
2. Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in the AEC industry: A literature overview. Automation in construction, 73, 145-165.
3. Liu, Y., & Xu, X. (2017). Industry 4.0 and cloud manufacturing: A comparative analysis. Journal of Manufacturing Science and Engineering, 139(3), 034701.
4. Bodein, Y., Rose, B., & Caillaud, E. (2013). A roadmap for parametric CAD efficiency in the automotive industry. Computer-Aided Design, 45(10), 1198-1214.
5. Howard, R., & Björk, B. C. (2007). Use of Standards for CAD Layers in Building Design Automation in Construction, 16(3), 290-297.
6. Sicari, S., Rizzardi, A., Miorandi, D., & Coen-Porisini, A. (2017). Dynamic Policies in the Internet of Things: Enforcement and Synchronisation. IEEE Internet of Things Journal, 4(6), 2228-2238.

7. Neisse, R., Steri, G., & Baldini, G. (2014, October). Enforcement of security policy rules for the Internet of Things. In 2014, the IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 165-172). IEEE.
8. Roin, B. N. (2013). The case for tailoring patent awards based on time-to-market. UCLA L. Rev., 61, 672.
9. Papavasileiou, A., Gavros, K., Vasileiadis, V., & Savvidis, S. (2002, September). CAD/CAM interfaces review. In Proceedings First International IEEE Symposium Intelligent Systems (Vol. 1, pp. 378-382). IEEE.
10. Nguyen, T. H. (2005). Integrating building code compliance checking into a 3D CAD system. In Computing in Civil Engineering (2005) (pp. 1-12).
11. Hadaya, P., & Marchildon, P. (2012). Understanding product lifecycle management and supporting systems. Industrial Management & Data Systems, 112(4), 559-583.
12. Sudarsan, R., Fenves, S. J., Sriram, R. D., & Wang, F. (2005). A product information modeling framework for product lifecycle management. Computer-aided design, 37(13), 1399-1411.
13. Terzi, S., Bouras, A., Dutta, D., Garetti, M., & Kiritsis, D. (2010). Product lifecycle management–from its history to its new role. International Journal of Product Lifecycle Management, 4(4), 360-389.
14. Srinivasan, V. (2011). An integration framework for product lifecycle management. Computer-aided design, 43(5), 464-478.
15. Ganesan, D., Keuler, T., & Nishimura, Y. (2009). Architecture compliance checking at run-time. Information and Software Technology, 51(11), 1586-1600.
16. Ullah, K. W., Ahmed, A. S., & Ylitalo, J. (2013, July). Towards building an automated security compliance tool for the cloud. In 2013, the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (pp. 1587-1593). IEEE.
17. Elgammal, A., Turetken, O., van den Heuvel, W. J., & Papazoglou, M. (2016). Formalizing and applying compliance patterns for business process compliance. Software & Systems Modeling, 15(1), 119-146.
18. González-Lluch, C., Company, P., Contero, M., Camba, J. D., & Plumed, R. (2017). A survey on 3D CAD model quality assurance and testing tools. Computer-Aided Design, 83, 64-79.
19. Bosché, F. (2010). Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. Advanced engineering informatics, 24(1), 107-118.
20. Lipman, R., & Lubell, J. (2015). Conformance checking of PMI representation in CAD model STEP data exchange files. Computer-Aided Design, 66, 14-23.
21. Malsane, S., Matthews, J., Lockley, S., Love, P. E., & Greenwood, D. (2015). Development of an object model for automated compliance checking. Automation in construction, 49, 51-58.