A11yUI-DSL: A Conceptual Framework for Generative Accessible User Interfaces

Abstract

Recent advances in generative user interface (GenUI) tools demonstrate the potential of artificial intelligence to rapidly produce high-fidelity UI prototypes from natural language prompts. While these systems accelerate design workflows, they fall short on accessibility: most outputs achieve only baseline compliance with standards such as WCAG, often resulting in homogenized, uninspired interfaces that fail to address diverse user needs (Guriţă & Vatavu, 2025). Meanwhile, research on adaptive UIs and model-driven engineering has shown that accessibility can be systematically supported through formal models and transformation rules (Zouhaier, BenDalyHlaoui, & Ben Ayed, 2023; Peissner, Häbe, Janssen, & Sellner, 2012). However, these approaches lack integration with contemporary generative workflows.

This paper introduces *A11yUI-DSL*, a conceptual framework for a Domain-Specific Language (DSL) for Generative Accessible User Interfaces. A11yUI-DSL makes accessibility heuristics first-class constructs, enabling designers to declare accessibility intent (e.g., keyboard navigation, contrast ratios, multimodal inputs) in a structured, declarative syntax. These specifications can be mapped to generative AI prompts, instantiated in accessible design systems, and validated against heuristic-based constraints. We illustrate the framework with example DSL snippets for common design scenarios (e.g., login screens, dashboards, conversational UIs), demonstrating how declarative constraints can balance accessibility enforcement with creative design diversity.

The contribution of this work is twofold: (1) a conceptual language model that embeds accessibility heuristics directly into generative design pipelines, and (2) a research agenda outlining design space tensions, open challenges, and evaluation strategies. By positioning accessibility as a generative constraint rather than a post hoc check, A11yUI-DSL reframes the role of DSLs in future human–AI collaboration for inclusive design.

Introduction

Advances in generative artificial intelligence (AI) have begun to transform the design of digital interfaces. A new class of generative user interface (GenUI) systems. These range from text-to-UI diffusion models to large language model–based prompt interpreters that can rapidly produce high-fidelity prototypes from natural language descriptions (Bleichner & Hermansson, 2023; Chen, Knearem, & Li, 2025). These systems promise to accelerate workflows for designers, reduce prototyping costs, and enable non-experts to generate usable interface drafts. However, despite their creative potential, such tools have revealed critical gaps in accessibility and inclusivity.

Accessibility is a foundational principle of human–computer interaction (HCI). It is enshrined in legal frameworks such as Section 508 of the Rehabilitation Act in the United States and the European Union's Web Accessibility Directive, and operationalized through international standards such as the Web Content Accessibility Guidelines (WCAG) and ISO 9241-171 (Miñón, Moreno, Martínez, & Abascal, 2014). Yet most generative design tools address accessibility only superficially. Recent evaluations show that AI-generated UIs often achieve baseline compliance (e.g., appropriate color contrast ratios) but produce homogenous layouts that fail to meet the diverse needs of disabled users (Guriţă & Vatavu, 2025). Practitioners describe this phenomenon as "accessibility handcuffs". These are designs that are formally compliant but aesthetically limited and creatively restrictive (Guriţă & Vatavu, 2025; Regan, 2004).

The consequences of this gap are significant. Designers remain skeptical of generative AI because of its poor support for nuanced accessibility requirements, including cognitive load, multimodal alternatives, and adaptive personalization (Chen et al., 2025). Moreover, because generative models are trained on existing interfaces, many of which themselves fail accessibility standards (WebAIM, 2023), the outputs risk reinforcing exclusionary patterns at scale. Against GenUI critiques highlight how overreliance on opaque AI systems may further obscure accountability for accessible design, leaving disabled users at risk of systemic marginalization (Okopnyi et al., 2024).

Previous research in adaptive and model-driven UI frameworks provides partial solutions. Systems such as SUPPLE (Gajos, Weld, & Wobbrock, 2008), MyUI (Peissner, Häbe, Janssen, & Sellner, 2012), and the AUIAC framework (Zouhaier, BenDalyHlaoui, & Ben Ayed, 2023) demonstrate how interfaces can adapt to user profiles, devices, and environmental contexts. However, these approaches are typically rule-bound, pattern-driven, and limited to adapting existing designs. They do not engage with the generative potential of modern AI, nor do they provide designers with mechanisms to declaratively embed accessibility requirements into the creative process. In short, while adaptive UIs can personalize, they do not generate accessible design space.

We argue that accessibility must be positioned not as a post hoc validation layer but as a first-class generative constraint. To advance this vision, this paper introduces A11yUI-DSL, a conceptual framework for a domain-specific language (DSL) that embeds accessibility heuristics directly into generative workflows. Unlike natural-language prompting, which is imprecise and difficult to validate, a DSL offers structured, declarative syntax through which designers can explicitly state accessibility intentions (e.g., "ensure keyboard navigation," "provide multimodal input options," "maintain minimum 4.5:1 contrast ratio"). These specifications can then be mapped into generative model prompts, instantiated in accessible design systems, and validated against heuristic-driven rules.

By elevating accessibility heuristics to the level of language constructs, A11yUI-DSL reframes how inclusive design can operate in generative contexts. Rather than treating accessibility guidelines as restrictive checklists, the DSL approach allows them to function as generative constraints that coexist with creative variation. For example, a simple login screen specification in A11yUI-DSL could yield multiple compliant yet stylistically distinct variants. Some optimized for low-vision users, others for motor impairments, and still others for general audiences. This approach aligns with broader calls in HCI for balancing accessibility compliance with design creativity (Regan, 2004; Guriţă & Vatavu, 2025).

The contributions of this paper are threefold. First, we conceptualize the A11yUI-DSL syntax and semantics, grounding them in established accessibility heuristics (including WCAG, ISO standards, and custom heuristic systems used in professional training). Second, we propose a pipeline model demonstrating how A11yUI-DSL specifications can be translated into generative AI prompts, instantiated within accessible design systems, and validated against heuristic rules. Third, we outline a research agenda that identifies key challenges, including DSL usability for non-programmers, integration with existing generative tools, and balancing constraint with creativity, while sketching pathways for future empirical evaluation and tool-building.

In doing so, this paper contributes to both HCI and software language engineering. For HCI, it extends ongoing debates about the role of AI in accessibility by offering a structured, language-based approach to inclusive generative design. For software language engineering, it demonstrates how DSL principles (Kosar, Martínez López, Barrientos, & Mernik, 2008; Karsai et al., 2009) can be adapted beyond traditional software productivity gains to address accessibility as a design-critical domain. Most importantly, it establishes a conceptual foundation upon which future systems can be built and evaluated, moving toward a vision where accessibility is not merely an evaluative afterthought but a generative condition of design itself.

2. Background & Related Work

2.1 Generative UI Tools

Recent advances in generative AI have produced a new wave of tools capable of creating user interface mock-ups from high-level prompts. Tools such as UIzard, Galileo AI, and Figma's First Draft combine natural language input with model-driven generation pipelines to produce high-fidelity wireframes and prototypes (Bleichner & Hermansson, 2023; Chen, Knearem, & Li, 2025). These systems have demonstrated clear value for rapid ideation, especially for non-designers and cross-functional teams.

However, empirical studies reveal limitations. UX practitioners report that GenUI tools are useful for producing a "good first draft," but they often fail to meet professional standards in the "last mile" of refinement (Chen et al., 2025). Furthermore, while these tools can accelerate early design exploration, they lack robust mechanisms for encoding domain-specific constraints such as accessibility, resulting in outputs that are often visually polished but exclusionary (Okopnyi et al., 2024; Bleichner & Hermansson, 2023).

2.2 Accessibility in HCI and Generative AI

Accessibility has long been a cornerstone of HCI, with frameworks such as WCAG 2.1, ISO 9241-171, and national accessibility mandates shaping interface evaluation (Miñón, Moreno, Martínez, & Abascal, 2014). Yet the intersection of accessibility and generative AI is underexplored. Recent evaluations of AI-generated interfaces found that while they frequently achieve baseline WCAG compliance (e.g., color contrast), they tend to replicate homogenized patterns that restrict creativity and fail to support diverse user needs (Guriţă & Vatavu, 2025).

This tension is not new. Designers have historically perceived accessibility as a limitation on creativity, equating compliance with restrictive or aesthetically uninspired designs (Regan, 2004). Generative AI risks reinforcing this perception by producing outputs that appear compliant but offer limited creative variation. Scholars caution that without explicit mechanisms for embedding accessibility requirements, generative models will continue to mirror biases and gaps in their training data, which often consists of inaccessible web and app interfaces (Guriţă & Vatavu, 2025; WebAIM, 2023).

2.3 Adaptive and Model-Driven UI Frameworks

Parallel research in adaptive and model-driven UI frameworks has demonstrated how accessibility can be systematically embedded into interface generation. Systems such as SUPPLE (Gajos, Weld, & Wobbrock, 2008), MyUI (Peissner, Häbe, Janssen, & Sellner, 2012), and the Adaptive User Interface to Accessibility Context (AUIAC) framework (Zouhaier, BenDalyHlaoui, & Ben Ayed, 2023) use rule-based transformations to personalize UIs for different user profiles, devices, and environmental contexts. These approaches exemplify the potential of model-driven engineering for accessibility, enabling automatic adaptation across sensory, cognitive, and motor impairments.

Yet limitations remain. Adaptive systems are constrained by predefined rule sets and pattern repositories. While they can adapt existing designs, they rarely create new ones, limiting their ability to support diverse creative solutions (Peissner et al., 2012). Moreover, studies show that designers often resist these systems due to their opacity and perceived loss of control over visual outcomes (Peissner et al., 2012; Zouhaier et al., 2023). In contrast, generative AI promises to expand design space rather than constrain it, but it lacks the structured accessibility mechanisms that adaptive frameworks provide.

2.4 Domain-Specific Languages (DSLs) and Usability

Domain-specific languages (DSLs) are specialized notations designed to capture the concepts and abstractions of a specific domain (Visser, 2008). DSLs have been shown to improve productivity, reduce boilerplate, and bridge the gap between domain experts and technical implementation (Kosar, Martínez López, Barrientos, & Mernik, 2008). Crucially, DSLs offer declarative precision: they allow users to specify *what* should be achieved, leaving the *how* to automated compilers or interpreters.

The success of a DSL depends not only on its expressive power but also on its usability. Research in software language engineering emphasizes that DSLs should align with users' cognitive models, support error prevention, and remain simple enough for non-specialists to adopt (Barišić, Amaral, & Goulão, 2012; Poltronieri, Zorzo, Bernardino, & Campos, 2018). Usability evaluation frameworks such as Usa-DSL highlight the importance of treating DSLs as human–computer interaction artifacts in their own right, subject to the same usability testing as interfaces (Poltronieri et al., 2018).

Despite these advances, the potential of DSLs for accessibility remains largely untapped. Previous attempts have focused on extending existing user interface description languages (UIDLs) with accessibility requirements (Miñón et al., 2014), but these remain technical rather than generative. To date, no DSL has been designed specifically to embed accessibility heuristics into generative AI design workflows. This gap is what A11yUI-DSL seeks to address.

3. Conceptual Framework: A11yUI-DSL

The central contribution of this paper is the conceptualization of *A11yUI-DSL*, a domain-specific language designed to embed accessibility heuristics into generative design workflows. Unlike natural language prompts, which are often vague, inconsistent, and difficult to validate, A11yUI-DSL offers a structured, declarative syntax through which designers can specify accessibility requirements. The framework rests on four interlocking principles: declarative accessibility, heuristic grounding, generative mapping, and balancing creativity with compliance.

3.1 Design Principles

1. **Declarative Accessibility**
   A11yUI-DSL adopts a declarative paradigm, allowing designers to express *what*

accessibility properties an interface must satisfy rather than *how those properties should be implemented*. For example, a designer can specify that a button must be keyboard-accessible, without dictating the exact HTML or ARIA markup. This separation of concerns ensures that accessibility intent is preserved even as generative systems produce stylistic or layout variations.

2. **Heuristic-Grounded Semantics**
   Each DSL construct is directly linked to accessibility heuristics and guidelines. The semantic layer ensures that constraints such as "minimum contrast ratio 4.5:1" or "label required for all input fields" are encoded as first-class rules. These heuristics may derive from standards (WCAG, ISO 9241-171) and extended heuristic systems used in practitioner training, ensuring that the DSL covers both compliance-oriented and user-centered accessibility considerations.

3. **Generative Mapping**
   DSL specifications serve as structured input for generative workflows. This mapping has two forms:

   - **Prompt-level translation**: A11yUI-DSL statements can be converted into structured generative prompts for text-to-UI models or large language models, ensuring that accessibility intent shapes generated outputs.

   - **Component-level instantiation**: DSL constructs can be mapped to accessible components in design systems (e.g., Material, Carbon, Fluent), guaranteeing baseline accessibility compliance in generated artifacts.

4. **Balancing Creativity and Compliance**
   A key principle of A11yUI-DSL is to avoid the "accessibility handcuffs" identified in recent research (Guriţă & Vatavu, 2025). While constraints are enforced, the DSL allows multiple stylistic solutions to emerge within accessible boundaries. For example, specifying "color-blind safe palette" does not dictate a single color scheme but rather ensures that all generated variants remain distinguishable for users with color vision deficiencies.

3.2 Syntax and Semantics

A11yUI-DSL adopts a lightweight, human-readable syntax resembling YAML or pseudo-code. Constructs are organized around *screens*, *components*, and *constraints*.

Example: Login Screen

```
Screen Login
  Component TextField "Username"
    Constraints: labelRequired true, minContrast 4.5, altInput voice
  Component TextField "Password"
    Constraints: labelRequired true, maskInput true
  Component Button "Sign In"
    Constraints: keyboardAccessible true, focusVisible true
Layout: linear, responsive, adaptFontSize true
```

- **Screen**: Top-level container representing a view.
- **Component**: UI element with semantic role (e.g., text field, button).
- **Constraints**: Accessibility rules applied to the component.
- **Layout**: Defines structural properties with adaptive features (e.g., responsive scaling, dynamic font size).

Semantics

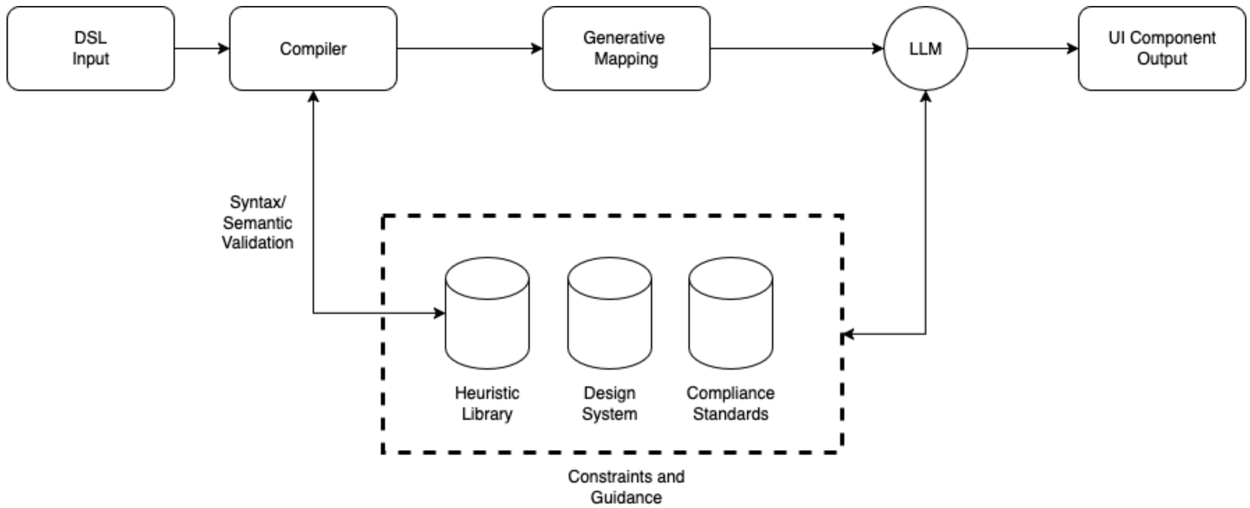
- `labelRequired true` ensures that all input fields include descriptive labels.
- `minContrast 4.5` enforces WCAG AA contrast compliance.
- `altInput voice` specifies an alternative input modality (speech recognition).
- `keyboardAccessible true` and `focusVisible true` ensure operability via keyboard.

Each constraint maps to a formal rule in the heuristic engine, making specifications machine-checkable and generatively enforceable.

3.3 Pipeline Model

The A11yUI-DSL workflow can be conceptualized as a layered pipeline:

1. **DSL Input**: Designers author specifications in A11yUI-DSL.
2. **Compiler/Interpreter**: Parses the DSL and validates syntax/semantics against the heuristic library.
3. **Generative Mapping**:
    - Converts DSL constraints into structured prompts for AI models (e.g., "Generate a login screen with input fields that include labels, a minimum 4.5:1 contrast ratio, and support for voice input").
    - Instantiates accessible components from a design system library.
4. **Accessibility Validator**: Automatically checks generated outputs against heuristics and WCAG standards, flagging violations for regeneration or designer review.
5. **UI Output**: Produces compliant design variants that balance accessibility and creativity.



This pipeline positions accessibility as an upstream constraint rather than a downstream correction.

3.4 Comparison with Natural Language Prompting

Current generative workflows rely heavily on natural language prompting, where designers issue commands such as *"Create a modern login screen with good contrast and easy navigation."* While intuitive, this approach suffers from ambiguity: models may interpret "good contrast" differently, and there is no guarantee of validation.

By contrast, A11yUI-DSL replaces ambiguity with structured declarations. Designers specify `minContrast 4.5`, which maps unambiguously to WCAG AA compliance. Furthermore, DSL-based prompts are machine-checkable and reusable, ensuring consistency across projects.

3.5 Summary

A11yUI-DSL thus represents a shift in how accessibility can be embedded into generative design. By combining declarative syntax, heuristic semantics, and generative mapping, it provides a language-based framework that treats accessibility not as a checklist but as a generative design driver.

4. Illustrative Use Cases

To demonstrate the potential of A11yUI-DSL, we present three use cases that highlight how declarative accessibility constraints can shape generative workflows. Each case contrasts a typical natural-language prompt with its equivalent in A11yUI-DSL, illustrating how structured specifications can yield accessible yet creative design outcomes.

4.1 Case 1: Login Screen

Natural Language Prompt

> "Design a clean login screen with a modern look and good accessibility."

Limitations

Although generative models may interpret this prompt to include high-contrast colors and large input fields, there is no guarantee that all accessibility requirements will be satisfied. Critical features such as input labels, keyboard navigation, and multimodal alternatives are often omitted (Bleichner & Hermansson, 2023).

A11yUI-DSL Specification

```
Screen Login
  Component TextField "Username"
    Constraints: labelRequired true, minContrast 4.5, altInput voice
  Component TextField "Password"
    Constraints: labelRequired true, maskInput true
  Component Button "Sign In"
    Constraints: keyboardAccessible true, focusVisible true
Layout: linear, responsive, adaptFontSize true
```

Resulting Output

- Multiple variants generated, each stylistically distinct but all including labeled input fields, contrast-compliant colors, keyboard operability, and voice input as an alternative modality.
- Accessibility is ensured at the specification level, freeing designers to explore aesthetics without risking exclusion.

4.2 Case 2: Data Dashboard

Natural Language Prompt

> "Generate a dashboard with graphs and widgets that are accessible to all users."

Limitations

Generative tools often produce visually appealing dashboards but neglect accessibility for users with low vision or color vision deficiency. Graphs may use red–green palettes, small labels, or low-contrast legends, violating accessibility guidelines (Guriţă & Vatavu, 2025).

A11yUI-DSL Specification

```
Screen Dashboard
  Component Chart "Sales Overview"
    Constraints: colorBlindSafe true, scalableText true, legendPosition top
  Component Table "Monthly Data"
    Constraints: headerRowRequired true, keyboardNavigable true
  Component Button "Export Data"
    Constraints: focusVisible true, minContrast 7.0
Layout: grid, responsive, highZoomSupport true
```

Resulting Output

- Variants include line and bar charts rendered with color-blind safe palettes and zoomable text.
- Chart components are rendered with a table representing the same data. (Tables provide a better accessible experience than charts).
- Tabular data is structured with headers and keyboard navigation.
- Buttons maintain AAA-level contrast ratios.
- Designers can select among stylistic options while knowing all generated dashboards meet accessibility heuristics.

4.3 Case 3: Conversational UI

Natural Language Prompt

> "Create a chat interface that is easy to use and accessible."

Limitations

Outputs typically resemble generic messaging apps with minimal consideration for multimodal interaction. For users with motor or cognitive impairments, these designs may be unusable without voice input, adaptive layouts, or simplified structures (Fadhil, 2018).

A11yUI-DSL Specification

```
Screen Chat
  Component MessageThread
    Constraints: scalableText true, contrastAdaptive true
  Component InputBox
    Constraints: altInput voice, keyboardAccessible true, labelRequired true
  Component Button "Send"
    Constraints: focusVisible true, sizeLarge true
Layout: adaptive, multimodalSupport true
```

Resulting Output

- Multiple chat interfaces generated, each supporting text and speech input, adaptive layouts for small or large screens, and high-contrast message bubbles.
- Users can scale text size dynamically, and input controls remain accessible through both keyboard and voice.
- Designers can evaluate stylistic variants while accessibility is enforced by default.

4.4 Synthesis of Use Cases

Across these scenarios, a consistent pattern emerges: natural-language prompts are too vague to guarantee accessibility, whereas A11yUI-DSL specifications produce outputs that are both compliant and diverse. By shifting accessibility from a downstream evaluation to an upstream generative constraint, A11yUI-DSL enables inclusivity without stifling creativity.

5. Design Space and Open Challenges

The conceptualization of A11yUI-DSL opens a design space at the intersection of accessibility, generative design, and domain-specific languages. While promising, this space raises several open challenges that must be addressed before such a system can be fully realized. We identify five central tensions.

5.1 Expressivity vs. Simplicity

A recurring challenge in DSL design is balancing the expressivity of the language with its ease of use (Karsai et al., 2009; Kosar et al., 2008). A highly expressive DSL may allow detailed specifications of accessibility constraints (e.g., multimodal interaction rules, dynamic adaptation thresholds), but such complexity risks alienating designers who lack technical training. Conversely, a simpler DSL may be easier to learn but fail to capture the nuance of accessibility heuristics. Future work must explore how to scaffold language adoption, perhaps through tiered syntax levels (basic vs. advanced) or visual abstractions for non-programmers.

5.2 Constraint Rigidity vs. Creative Freedom

One of the primary critiques of AI-generated UIs is that they often achieve accessibility compliance at the cost of creative diversity, a phenomenon termed “accessibility handcuffs” (Guriţă & Vatavu, 2025). DSL-driven constraints risk reinforcing this problem if they are too rigid, producing repetitive or uninspired designs. A key challenge is to support multi-objective optimization: generating outputs that satisfy accessibility rules while still allowing for stylistic variation. Approaches such as genetic algorithms for UI generation (Troiano & Birtolo, 2012) may offer pathways for balancing these objectives.

5.3 Integration with Generative Systems

For A11yUI-DSL to function in practice, its specifications must be effectively translated into generative workflows. This integration involves two technical hurdles: (1) mapping DSL constraints to natural language prompts or structured conditioning for generative models, and (2) instantiating DSL constructs in accessible component libraries (e.g., Material, Carbon, Fluent). Given that generative systems vary in representation, from pixel-based image models to code-generating transformers (Chen et al., 2025), a robust translation layer will be required. Research is needed on interoperability standards that link declarative accessibility rules with diverse generative architectures.

5.4 Evaluation Metrics and Methods

Accessibility evaluation in generative contexts remains underdeveloped. Current GenUI evaluations focus on superficial compliance, with limited alignment between AI evaluators, automated tools, and human testers (Guriţă & Vatavu, 2025; Miñón et al., 2014). A key challenge is to develop comprehensive evaluation metrics that assess not only compliance but also usability, adaptability, and creative diversity. Furthermore, A11yUI-DSL itself must be evaluated as an HCI artifact, using established frameworks for DSL usability testing (Barišić, Amaral, & Goulão, 2012; Poltronieri et al., 2018).

5.5 Usability of the DSL Itself

Finally, A11yUI-DSL must be usable by a wide range of stakeholders, from accessibility specialists to UX designers and developers. Prior studies highlight that DSLs often fail when they are too technical or misaligned with users' mental models (Visser, 2008; Barišić et al., 2012). Ensuring usability will require iterative co-design with practitioners, lightweight syntax that reduces cognitive load, and validation tools that provide immediate feedback on DSL correctness. Embedding usability evaluation methods into the DSL development process is therefore critical.

5.6 Summary

Together, these challenges highlight the complexity of embedding accessibility into generative workflows. Addressing them requires interdisciplinary research that spans HCI, AI, accessibility, and software language engineering. A11yUI-DSL is thus best understood as a provocation and roadmap: a conceptual proposal that surfaces the tensions and opportunities at this emerging intersection, and that invites future empirical and technical work to explore solutions.

6. Implications & Roadmap

The conceptual framework of A11yUI-DSL has implications for both research and practice. By embedding accessibility heuristics into generative workflows through a structured language, A11yUI-DSL reframes accessibility not as an afterthought but as a driver of design generation. This reorientation provides opportunities to expand inclusivity, advance methodological approaches, and guide the next phase of tool development.

6.1 Implications for Research

For researchers in human–computer interaction, accessibility, and software language engineering, A11yUI-DSL provides a new conceptual model for studying the interplay between accessibility constraints and generative design. It offers a formal mechanism to operationalize accessibility heuristics in design generation, thereby enabling controlled experiments on trade-offs between compliance, creativity, and usability. Moreover, it extends prior work in adaptive and model-driven UIs by proposing a generative approach, shifting the focus from rule-based adaptation to language-driven creation. Research communities can use A11yUI-DSL as a foundation for exploring multi-objective optimization, heuristic validation pipelines, and DSL usability evaluation frameworks.

6.2 Implications for Practice

For practitioners, A11yUI-DSL offers a pathway to integrate accessibility into everyday generative workflows. UX designers and developers could leverage DSL constructs to specify accessibility requirements with precision, ensuring that generated outputs meet standards while still offering stylistic diversity. Accessibility specialists could use A11yUI-DSL to formalize heuristics and enforce organizational compliance policies. Product teams might adopt the language as a bridge between design intent and implementation, reducing miscommunication across roles. In each case, the DSL enables accessibility-first design without significantly increasing workload or technical overhead.

6.3 Roadmap for Future Work

The next phase of work involves moving from conceptualization to implementation and empirical validation. Four directions are particularly salient:

1. **Prototype Development**: Build a minimal A11yUI-DSL compiler or interpreter that supports a subset of constructs, mapping them to generative AI prompts and accessible component libraries.

2. **Integration with Design Systems**: Connect DSL specifications to existing accessible design frameworks such as Material, Carbon, or Fluent, enabling seamless generation of compliant components.

3. **Empirical Evaluation**: Conduct studies with UX designers and users with disabilities to compare DSL-driven generative workflows with natural language prompting. Evaluation criteria should include accessibility compliance, design diversity, usability, and perceived creative freedom.

4. **Iterative Refinement**: Use insights from prototype deployment and evaluation to refine the DSL's syntax, semantics, and usability. This iterative process can draw on established methodologies in software language engineering, such as incremental language development and participatory evaluation.

6.4 Positioning as an Agenda-Setting Contribution

A11yUI-DSL should be understood as both a framework and a provocation. It demonstrates the feasibility of embedding accessibility heuristics as generative constraints while highlighting the challenges that must be addressed to make such systems usable and effective. By providing a roadmap for future work, this paper invites interdisciplinary collaboration across AI, HCI, accessibility, and programming languages to realize the vision of inclusive generative design.

7. Conclusion

Generative AI has introduced new possibilities for user interface design, enabling rapid prototyping and democratized access to design tools. Yet as current research and practice reveal, these systems often fall short on accessibility, producing interfaces that are formally compliant but creatively limited or, in many cases, exclusionary. The risk is clear: without explicit mechanisms for embedding accessibility into generative workflows, AI-driven design may replicate existing inequities at scale.

This paper has proposed *A11yUI-DSL*, a conceptual framework for a domain-specific language that embeds accessibility heuristics directly into generative user interface design. By elevating accessibility to a first-class generative constraint, A11yUI-DSL reframes accessibility not as a downstream evaluation but as an upstream driver of inclusive design. Through design principles, illustrative syntax, pipeline modeling, and use cases, we have outlined how the DSL approach can offer both compliance and creative flexibility.

The contributions of this work are threefold. First, it articulates a conceptual model for embedding accessibility heuristics into generative workflows, addressing a critical gap in current

GenUI research. Second, it demonstrates, through concrete use cases, how structured declarative specifications can outperform vague natural language prompts in producing accessible and diverse outputs. Third, it identifies a research agenda, highlighting design tensions and open challenges that must be resolved through interdisciplinary collaboration.

A11yUI-DSL is not presented as a finished system but as an agenda-setting proposal. Its value lies in defining a research and design space where accessibility, generative AI, and domain-specific languages converge. Future work will require prototyping, integration with design systems, and empirical evaluation involving both practitioners and users with disabilities. In doing so, we can move closer to a vision where generative systems do not merely replicate accessible patterns but create them by design.

By positioning accessibility as a generative condition rather than a corrective check, A11yUI-DSL seeks to expand both the inclusivity and creativity of future design practices. The challenge now is to transform this conceptual framework into actionable tools and methods that ensure generative AI contributes to, not detracts from, the goal of universal access.

References

Barišić, A., Amaral, V., & Goulão, M. (2012, September). *Usability evaluation of domain-specific languages*. In 2012 Eighth International Conference on the Quality of Information and Communications Technology (QUATIC) (pp. 342–347). IEEE. https://doi.org/10.1109/QUATIC.2012.63

Bleichner, J., & Hermansson, P. (2023). *Generative UI design with diffusion models: Exploring automated interface creation and human-computer interaction*. arXiv preprint arXiv:2305.XXXX.

Chen, X., Knearem, T., & Li, Z. (2025). *The GenUI Study: Exploring the design of generative UI tools to support UX practitioners and beyond*. Proceedings of the ACM on Human-Computer Interaction, 9(CSCW). https://doi.org/10.1145/XXXXXXX

Fadhil, A. (2018). *Domain-specific design patterns: Designing for conversational user interfaces*. In Proceedings of the International Conference on Conversational UX Design. arXiv preprint arXiv:1802.09055.

Gajos, K. Z., Weld, D. S., & Wobbrock, J. O. (2008). *Automatically generating personalized user interfaces with SUPPLE*. Artificial Intelligence, 174(12–13), 910–950. https://doi.org/10.1016/j.artint.2010.05.002

Guriţă, A., & Vatavu, R.-D. (2025). *How well can generative AI design and evaluate user interfaces?* Multimedia Tools and Applications, 84(3), 4451–4478. https://doi.org/10.1007/s11042-023-14390-5

Karsai, G., Krahn, H., Pinkernell, C., Rumpe, B., Schindler, M., & Völkel, S. (2009). *Design guidelines for domain-specific languages*. In Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM'09). Orlando, FL.

Kosar, T., Martínez López, P. E., Barrientos, P. A., & Mernik, M. (2008). *A preliminary study on various implementation approaches for domain-specific languages*. Information and Software Technology, 50(5), 390–405. https://doi.org/10.1016/j.infsof.2007.04.002

Miñón, R., Moreno, L., Martínez, P., & Abascal, J. (2014). *Development of accessible user interfaces for ambient intelligence systems*. Interacting with Computers, 26(5), 406–421. https://doi.org/10.1093/iwc/iwt050

Okopnyi, P., Turner, F., Brehmer, M., & Westendorf, L. (2024). *Against generative UI*. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '24). ACM.

Peissner, M., Häbe, D., Janssen, D., & Sellner, T. (2012, June). *MyUI: Generating accessible user interfaces from multimodal design patterns*. In Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (pp. 81–90). ACM. https://doi.org/10.1145/2305484.2305500

Poltronieri, I., Zorzo, A. F., Bernardino, M., & Campos, M. de B. (2018). *Usability evaluation framework for domain-specific language: A focus group study*. In Proceedings of the 2018 ACM Symposium on Applied Computing (pp. 1629–1636). ACM. https://doi.org/10.1145/3167132.3167348

Regan, B. (2004). *Accessibility and creativity: Finding balance in interface design*. Journal of Usability Studies, 2(1), 45–53.

Visser, E. (2008). *WebDSL: A case study in domain-specific language engineering*. In R. Lämmel, J. Visser, & J. Saraiva (Eds.), Generative and Transformational Techniques in Software Engineering II (pp. 291–373). Springer. https://doi.org/10.1007/978-3-540-88643-3_7

WebAIM. (2023). *The WebAIM Million: An annual accessibility analysis of the top 1,000,000 home pages*. Center for Persons with Disabilities, Utah State University. https://webaim.org/projects/million/

Zouhaier, K., BenDalyHlaoui, Y., & Ben Ayed, L. (2023). *Adaptive user interfaces to accessibility context: A framework for user-centered accessibility*. Universal Access in the Information Society, 22(2), 201–218. https://doi.org/10.1007/s10209-022-00884-7