

I-RPG: An Interactive and Collaborative Framework for Repository Planning Graph Construction

AI Research Assistant

September 22, 2025

Abstract

Recent advancements in large language models (LLMs) have enabled the automated generation of entire software repositories. The Repository Planning Graph (RPG) has emerged as a key data structure for this process, providing a unified representation of a repository’s functionality, file structure, and data flows. However, the fully automated construction of RPGs can lead to plans that are misaligned with user intent, lack domain-specific nuance, or are difficult to debug and refine. To address these limitations, we propose the Interactive Repository Planning Graph (I-RPG), a framework that integrates human developers into the RPG construction process. I-RPG provides a visual interface for developers to inspect, modify, and collaborate on the RPG in real-time. It also includes an LLM-powered suggestion engine that assists developers by proposing relevant additions, identifying potential inconsistencies, and automating repetitive tasks. We argue that this human-in-the-loop approach will lead to the creation of more accurate, robust, and customized software repositories.

1 Introduction

The automated generation of software repositories from high-level specifications is a long-standing goal of software engineering. The recent success of large language models (LLMs) has brought this goal closer to reality. Frameworks like ZeroRepo, which uses a Repository Planning Graph (RPG) to guide the generation process, have demonstrated the ability to create complex, multi-file projects from scratch.

The RPG is a powerful abstraction that represents a repository’s plan in a structured, hierarchical manner. However, the process of constructing the RPG is typically a “black box,” where the LLM makes all the planning decisions without human intervention. This can lead to several challenges:

- **Misalignment with User Intent:** The LLM may misinterpret the user’s high-level specification, leading to an RPG that does not accurately reflect their desired outcome.
- **Lack of Domain-Specific Knowledge:** The LLM may not have the necessary domain-specific knowledge to create an optimal plan for a specialized repository (e.g., in scientific computing or embedded systems).
- **Difficulty in Debugging and Refinement:** If the generated code is incorrect, it can be difficult to trace the error back to a flaw in the RPG.

To address these challenges, we propose the Interactive Repository Planning Graph (I-RPG), a framework that transforms RPG construction from a fully automated process into a collaborative, human-AI partnership.

2 Related Work

Our work builds directly on the foundation laid by Luo et al. in "RPG: A Repository Planning Graph for Unified and Scalable Codebase Generation" (<http://arxiv.org/pdf/2509.16198v1>). We adopt the core RPG data structure but focus on the process of its construction.

Our approach is also inspired by the broader field of human-in-the-loop AI, which seeks to combine human and machine intelligence to solve complex problems. This paradigm has been successfully applied in areas such as machine translation, data labeling, and creative design.

3 The I-RPG Framework

The I-RPG framework is designed to facilitate real-time collaboration between human developers and an LLM-powered assistant during the repository planning phase. The framework consists of three main components: a visualizer, a user interaction toolkit, and an LLM suggestion engine.

3.1 Mathematical Formalism

We formally define an RPG as a directed acyclic graph (DAG) $G = (V, E)$, where:

- V is a set of vertices (nodes) representing the components of the repository (e.g., modules, files, classes, functions). Each vertex $v \in V$ has a type $T(v)$ and a set of attributes $A(v)$.
- E is a set of edges representing the relationships between the components. An edge $e = (u, v) \in E$ indicates a dependency or data flow from node u to node v .

3.2 The I-RPG Visualizer

The visualizer provides an interactive, graphical representation of the RPG. It allows developers to explore the graph, inspect the attributes of each node and edge, and understand the overall structure of the planned repository.

3.3 User Interaction Toolkit

The user interaction toolkit provides a set of operations that developers can use to modify the RPG. These operations include:

- **Node Operations:**

- `add_node(type, attributes)`: Adds a new node to the graph.
- `remove_node(v)`: Removes a node and its associated edges.
- `update_node(v, new_attributes)`: Modifies the attributes of a node.

- **Edge Operations:**

- `add_edge(u, v)`: Adds a new edge between two nodes.
- `remove_edge(e)`: Removes an edge.

These operations can be expressed as transformations on the graph G . For example, the `add_node` operation can be defined as:

$$G' = (V \cup \{v_{new}\}, E)$$

where v_{new} is the new node.

3.4 LLM Suggestion Engine

The LLM suggestion engine acts as an intelligent assistant, providing real-time feedback and suggestions to the developer. The engine can perform several tasks:

- **Graph Completion:** Given a partially constructed RPG, the engine can suggest new nodes and edges to complete the plan. This can be formulated as a function $S(G_{partial}) = G_{suggestion}$, where $G_{suggestion}$ is a subgraph that can be merged with $G_{partial}$.
- **Inconsistency Detection:** The engine can analyze the RPG for potential inconsistencies, such as circular dependencies or mismatched data flows. This can be represented as a validation function $f_{validate} : G \rightarrow \{true, false\}$.
- **Automated Refinement:** The engine can suggest refactorings to the RPG that would improve its modularity or efficiency.

4 Evaluation

To evaluate the effectiveness of the I-RPG framework, we propose a user study with two groups of developers. The first group will be asked to create a software repository using a traditional, non-interactive code generation tool. The second group will be asked to create the same repository using the I-RPG framework.

We will collect the following metrics for each group:

- **Task Completion Time:** The time it takes for the developers to create a functional repository.
- **Code Quality:** The quality of the generated code, as measured by metrics such as code complexity, test coverage, and bug density.
- **User Satisfaction:** The developers' subjective satisfaction with the development process.

We hypothesize that the I-RPG group will produce higher-quality code in less time and report higher levels of satisfaction.

5 Conclusion

The I-RPG framework represents a shift from fully automated code generation to a more collaborative, human-centric approach. By allowing developers to interact with and refine the repository plan in real-time, we believe that I-RPG will enable the creation of more robust, accurate, and customized software. This work opens up several avenues for future research, including the development of more sophisticated LLM suggestion engines and the integration of I-RPG with existing IDEs and development workflows.