Process-Aware RAG for Smart City Infrastructure Maintenance Documentation

Ada Bagozi, Devis Bianchini, Massimiliano Garda, Michele Melchiori, Anisa Rula
Dept. of Information Engineering, University of Brescia
Via Branze, 38 – 25123 Brescia, Italy
Emails: {ada.bagozi, devis.bianchini, massimiliano.garda, michele.melchiori, anisa.rula}@unibs.it

***Abstract*—Smart-city infrastructures rely on large collections of technical manuals, maintenance procedures, troubleshooting guidelines, and safety protocols. Such documentation contains procedural knowledge that is essential for operating smart buildings, energy systems, water networks, mobility assets, and public-service facilities. Retrieval-Augmented Generation (RAG) can make this knowledge accessible through natural-language interfaces, but standard chunk-based RAG is poorly suited to procedural content: chunking may break step ordering, prerequisites, branches, and safety warnings. This extended abstract refocuses a previous Process-Aware RAG (PA-RAG) research vision on smart-city infrastructure maintenance. We outline a conceptual architecture that extracts procedures from technical documentation into a knowledge graph and supports conversational querying through an on-premise Small Language Model (SLM), limiting the exposure of sensitive operational information.**

***Index Terms*—Retrieval-Augmented Generation, Smart City Infrastructure, Technical Documentation, Procedural Knowledge, Knowledge Graphs, Small Language Models**

I. Introduction

Smart cities are increasingly composed of interconnected cyber-physical infrastructures: smart buildings, energy and water systems, mobility assets, environmental monitoring devices, and public-service facilities. Their maintenance and operation depend on technical documentation that is often large, heterogeneous, and difficult to navigate under time pressure. Municipal technicians, facility managers, utility operators, infrastructure maintainers, and emergency or incident-response teams may need to answer operational questions such as: *"Which checks are required before restarting this pumping station?"*, or *"What procedure should be followed when a smart-building HVAC unit reports fault code F12?"*

RAG [1] is a natural candidate for supporting such users, since it grounds language-model responses in domain-specific document collections. However, technical procedures are not independent text fragments. Their meaning depends on sequential order, decision conditions, prerequisites, safety warnings, tools, components, and relationships with other procedures. Standard RAG pipelines, based on chunking and dense vector similarity, may retrieve a later step without its required preconditions or fail to surface a relevant variant or sub-procedure. This is a structural mismatch between flat retrieval and procedural knowledge.

The smart-city setting also motivates security-aware deployment. Technical manuals, configuration details, maintenance workflows, and operator queries may reveal sensitive operational information about critical urban assets. Exposing such information to external cloud services can increase governance risks and enlarge the cyber-physical attack surface. While cloud LLMs may be suitable for public or low-risk documentation, PA-RAG considers on-premise SLM-based deployment as a practical option for scenarios involving critical infrastructure and operational data.

This extended abstract builds on our previous PA-RAG research vision for technical documentation [2] and contextualizes it to smart-city infrastructure maintenance. Its contribution is threefold: (i) it motivates process-aware retrieval for smart-city maintenance documentation; (ii) it outlines a knowledge-graph-based architecture for preserving procedural structure and source traceability; and (iii) it identifies representative users, use cases, and open challenges for future prototyping.

II. Background

Recent work at the intersection of Large Language Models (LLMs) and Business Process Management (BPM) shows that generative models can support the extraction and interpretation of process knowledge. Kampik et al. [3] frame Large Process Models as a direction for combining generative AI with formal process logic. Franzoi et al. [4] show how LLMs can generate process knowledge from heterogeneous enterprise content, highlighting the relevance of unstructured documentation as a source of operational processes. In parallel, GraphRAG approaches [5] demonstrate that graph structures can support retrieval beyond isolated text chunks by surfacing coherent subgraphs. PA-RAG connects these directions by focusing on procedural technical documentation and on retrieval mechanisms that preserve process logic.

III. PA-RAG for Smart Infrastructure Documentation

Fig. 1 summarizes the proposed PA-RAG conceptual architecture. The system is organized into two phases. In the *knowledge construction* phase, PDF manuals and maintenance documents are processed by an LLM/SLM extractor to identify procedure boundaries, ordered steps, tools, target components, and cross-references. The extracted elements are instantiated in a procedural knowledge graph grounded in a variant of

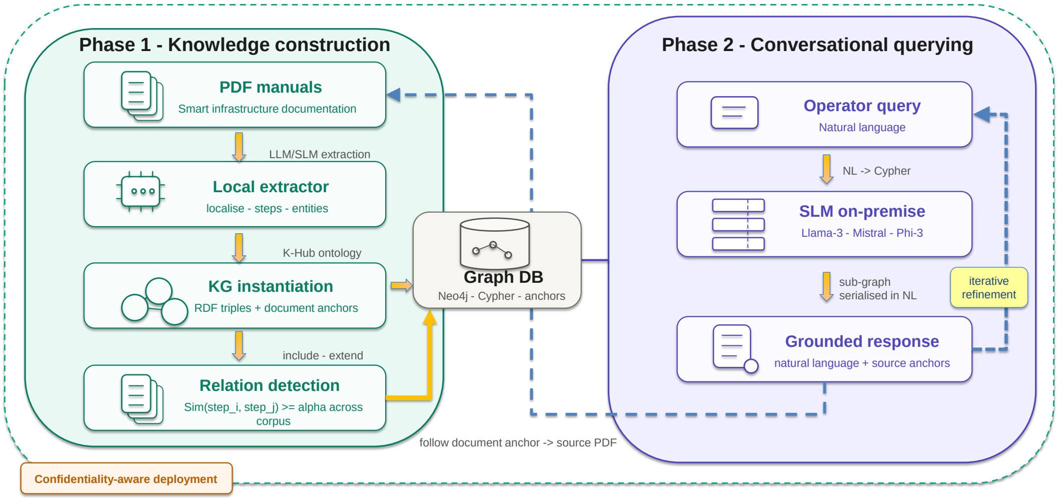

Fig. 1. Conceptual PA-RAG architecture for smart-city infrastructure maintenance documentation.

the K-Hub procedure ontology[1]. Procedures are represented as plans, steps are connected through ordering relations such as `nextStep`, and each node is associated with a document anchor, e.g., document identifier, page, and section.

A second construction step identifies inter-procedural relationships. The relation `include` links a procedure to another one whose steps are subsumed by it, while `extend` captures structurally analogous variants with the same step-level organization. Such relations capture general maintenance interventions and their variants across different equipment, locations, safety conditions, and mobility constraints.

In the *conversational querying* phase, a user submits a natural-language request. The SLM translates it into a graph query, for instance Cypher over Neo4j, and retrieves a structurally coherent subgraph rather than a set of independent chunks. The retrieved subgraph can include the requested procedure, its prerequisites, related variants, and source anchors. The SLM then produces a natural-language response grounded in the graph, while the user can follow anchors to inspect the original manual documentation.

IV. Potential Users and Use Cases

Potential users include municipal technical offices, smart-building facility managers, utility operators, maintenance contractors, control-room personnel, and emergency-response teams. These users typically need operational answers that are both fast and traceable, especially when procedures concern public services or safety-critical infrastructure.

A first use case concerns *smart-building maintenance*. A facility manager receives an HVAC fault code and asks which checks must be performed before restarting the unit. PA-RAG retrieves the relevant fault-resolution procedure, preserves the required order of checks, highlights safety warnings, and surfaces model-specific variants. A second use case concerns *urban utility operations*. A water-network operator asks which procedures for a pumping station include the isolation and pressure-release steps required before replacing a valve. The graph traversal can identify procedures that subsume the required intervention and return source anchors for auditing and verification.

[1]The Knowledge Hub Ontology – Procedure Module, Cefriel, release 1.0.0, Dec. 2022. Available: https://github.com/cefriel/k-hub-ontology

These examples illustrate why process awareness matters: the relevant answer is not only a passage of text, but a structured set of steps, constraints, variants, and links to source documentation.

V. Open Challenges and Outlook

Several challenges remain open: multimodal extraction from diagrams, flowcharts, maps, and annotated images; reconstruction of procedures spanning multiple pages or documents; domain adaptation of on-premise SLMs for NL-to-Cypher and graph-grounded generation; and evaluation beyond standard RAG metrics, with emphasis on procedural completeness.

Future work will include the design of a PA-RAG prototype on a representative corpus of smart-city infrastructure maintenance manuals, with particular attention to security-aware deployment and to the validation of answers against procedural constraints.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems (NeurIPS 2020)*, vol. 33, 2020, pp. 9459–9474.
[2] A. Bagozi, D. Bianchini, M. Garda, M. Melchiori, and A. Rula, “Towards process-aware rag for technical documentation: An overview and research vision,” in *Proceedings of the 2nd International Conference on AI Revolutions (AIR 2026)*. Bursa, Türkiye: IEEE, June 10–12 2026.
[3] T. Kampik, C. Warmuth, A. Rebmann *et al.*, “Large process models: Business process management in the age of generative AI,” *KI - Künstliche Intelligenz*, vol. 38, no. 1, pp. 1–12, 2024.
[4] S. Franzoi, M. Delwaulle, J. Dyong, J. Schaffner, M. Burger, and J. vom Brocke, “Using large language models to generate process knowledge from enterprise content,” in *Proceedings of the NLP4BPM Workshop, co-located with the 22nd International Conference on Business Process Management (BPM 2024)*, Kraków, Poland, 2024.
[5] D. Edge, H. Trinh, N. Cheng, J. Bradley *et al.*, “From local to global: A GraphRAG approach to query-focused summarization,” Feb. 2024. [Online]. Available: https://arxiv.org/abs/2404.16130