COEVOLVE

Knowledge Graph for Traceability in Basic Software Porting

Aimen Medjek[1], Andreas Kappel[1], Florian Riedl[2], Amer Kajmaković[1], Tobias Scheipel[2], Marcel Baunach[2], Gerhard Wirrer[3], Alexander Mattausch[4], Judith Schaeffer[5]

Pro2Future GmbH[1], TUG-ITI[2], Infineon Technologies AG[3], Elektrobit Automotive GmbH[4], Aumovio GmbH[5]

[1] Sandgasse 34, 8010 Graz (AUT)
[2] Inffeldgasse 16/1, 8010 Graz (AUT)
[3] Am Campeon 1-15, 85579 Neubiberg (GER)
[4] Am Wolfsmantel 46, 91058 Erlangen (GER)
[5] Guerickestraße 7, 60488 Frankfurt am Main (GER)

MOTIVATION & GOALS

Challenge: Porting Basic Software to new HW architecture requires preserving **traceability** across SW requirements, HW specifications, and source code. Relational databases struggle to represent the dense, cross-cutting relationships between these artifacts.

Goal: Build a **Knowledge Graph (KG)** that consolidates MCAL and AUTOSAR requirements, HW specifications, and source-code artifacts into a single traceable structure that an **AI-assisted porting pipeline** can query. On top of this KG, we use **GraphRAG** rather than plain RAG, so retrieval captures dependencies across artifact types instead of isolated text chunks.

Project FactBox

Project Name	COEVOLVE
Project ID	MFP II Prod.1
Duration	44 Months

Area 4.1
Cognitive Products

Project Lead
DI Dr. Amer Kajmaković

APPROACH

We ingest and embed AUTOSAR, MCAL, and hardware artifacts into a **graph database** with typed requirement, source-code, and hardware **nodes** connected with **edges** *(e.g., implements, depends-on)*. We also embed the **Code Property Graph** of the origin HW to link source-code structure to this context. **GraphRAG** then augments semantic search by **pulling neighboring graph nodes**, e.g. giving driver code and register definitions together, before generation.

CONTRIBUTION

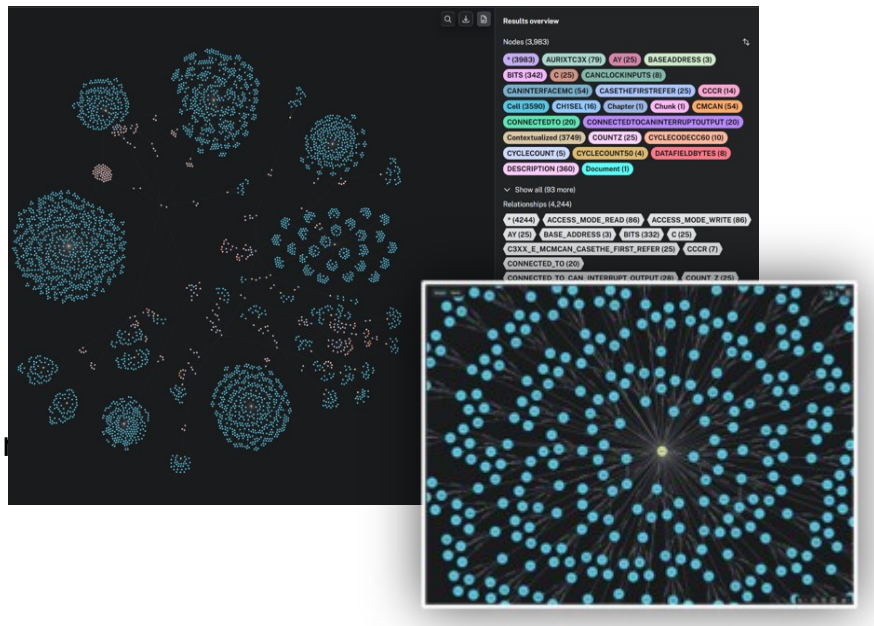
Scientific contribution
- Proposed KG-based traceability approach for automotive porting.
- Typed schema across requirement, source-code, hardware.
- GraphRAG adds graph-neighbor retrieval to semantic search.
- LLM sees driver logic + relevant HW architecture, not just text.

Economic contribution
- Less manual cross-referencing of SRS, MCAL, and hardware docs.
- Speeds up Steps 3 and 4 of the porting workflow.
- Reusable ontology cuts future re-engineering cost.

SYSTEM ARCHITECTURE

Prototype Knowledge Graph built in Neo4j from ingested requirements, source code, and hardware specs.

The HW-specification and AUTOSAR/MCAL requirement subgraphs are constructed with a **deterministic extraction** pipeline, while the origin source code's Code Property Graph is built through **static analysis**. Each subgraph follows its own **ontology**, so **GraphRAG** retrieval is adapted per subgraph rather than applied uniformly across the full KG.



Contact: DI Dr. Amer Kajmaković, Pro2Future GmbH, amer.kajmakovic@pro2future.at, +43 664 888 9210
Acknowledgement: This work was supported by Pro²Future II (FFG, 911655), Infineon Technologies AG, Elektrobit Automotive GmbH, and Aumovio GmbH.