HAK-GAL Suite: Technical Report and Future Outlook

Date: 2025-07-13
Prepared by: AI-Assistant (as scientific collaborator)

---

1. Executive Summary

The HAK-GAL (Hybrid Assertion Knowledge & Grounded Assertion Logic) Suite is a hybrid AI framework designed to provide verifiable reasoning by combining RAG-based document retrieval, formal logical provers, and a structured knowledge base. To date, the system has reached a stable Alpha stage, featuring: - A modular front-end with four interactive panels (RAG Context, Interaction, Query Builder, Profile) and live system metrics.
- A back-end API (Flask) orchestrating an LLM ensemble (DeepSeek, Mistral, Gemini), Lark-based parsing, advanced RAG indexing, and integration of formal provers (Z3 SMT, Functional Constraint, Pattern-Matcher, Wolfram|Alpha).
- A two-stage Hybrid Parser achieving 100 % classification and formula-generation accuracy in critical tests.
- An initial domain ontology (34 concepts, 95 property mappings) enabling semantic validation.
- A semantic knowledge base with 43 enriched facts and integration into the retrieval pipeline.

Despite these successes, key challenges remain in entity extraction, syntax validity, ontology ranking, and retrieval precision. This report documents the current technical state and outlines a roadmap for performance optimization, feature completion, and preparation for production deployment.

---

2. System Architecture

2.1 Front-end

- **Framework:** React (Tailwind CSS) with modular components.
- **Layout:** Three-column grid: Knowledge Base, Interaction Panel, Advanced Control Panel.
- **Panels:**
- **Knowledge Base:** Displays permanent facts, learning suggestions, and data sources.
- **Interaction Panel:** Chat-style command entry and response display, with support for `ask`, `learn`, `explain`, etc.
- **Advanced Control Panel:** Tabs for RAG Context, Orchestrator configuration, Query Builder, and Profile.
- **Metrics Display:** Total facts, cache hit rate, average query latency.
- **Theme Support:** Light/Dark toggles; planned Easy/Scientific mode switch.

2.2 Back-end

- **API Server:** Flask, serving on `127.0.0.1:5001` (development mode).
- **LLM Ensemble:** DeepSeek, Mistral, Gemini via unified provider interface.
- **Parsing Engine:** Lark-based grammar for logical formulas; optional hybrid parser with regex and NLP models.
- **RAG Pipeline:** 41-chunk index of data sources; sentence-transformers embeddings (all-MiniLM-L6-v2).

- **Formal Provers:**
- Functional Constraint Prover
- Z3 SMT Solver
- Pattern Matcher (DSL-based heuristics)
- Wolfram|Alpha Oracle
- **Knowledge Graph:** JSON persistence for permanent facts; optional vector index caching (Faiss).
- **Timeout Protection:** 30 s per command with category-specific limits.

2.3 Data Sources and Persistence

- **Source Files:** `test_document.txt`, `HAK_GAL_Wissensfakten.txt`, `geodb.txt` for initial ingest.
- **Indexing:** Chunking, embedding, and storing in vector store; persistent KG saved as JSON.
- **Ontology:** `hak_gal_ontology.json` (OWL-based conversion) persisted and lazy-loaded.

---

3. Module Status and Validation

3.1 Hybrid Parser (Step 2.1)

- **Functional Tests:** 6 critical queries; 100 % classification and formula accuracy.
- **Entity Extraction:** 66.7 % accuracy for multi-entity cases; pending improvement.
- **Syntax Validity:** 55 % grammar compliance; post-processing fixer in development.

3.2 Ontology Integration (Step 2.2)

- **Concept Coverage:** 34 classes, 95 property mappings loaded successfully.
- **Semantic QA:** 4 test cases; 100 % functional retrieval with semantic confidence 0.83–0.87.
- **Predicate Inference:** Multiple candidates returned; ranking mechanism required.

3.3 Knowledge Base Integration (Step 2.3)

- **Facts Extracted:** 43 enriched facts from three source files.
- **Retrieval Accuracy:** Top-3 relevant facts for standard queries; precision declines (< 50 %) for niche queries.
- **Performance:** Average query time ~300 ms; cache hit rate ~78 %.

---

4. Performance Metrics

Metric	Value	Target (Prod)
Average Query Latency	301 ms	< 200 ms
Cache Hit Rate	78.2 %	> 90 %
Classification Accuracy	100 %	≥ 95 %
Formula Generation Accuracy	100 %	≥ 95 %
Syntax Validity	55 %	≥ 80 %

Metric	Value	Target (Prod)
Entity Extraction Accuracy	66.7 %	≥ 90 %
Prover Success Rate (avg)	89 %	≥ 95 %

---

5. Key Challenges

1. **Entity Recognition:** Missed compounds and non-canonical terms.
2. **Syntax Compliance:** Dropped punctuation and spacing errors.
3. **Ontology Ranking:** Overproduction of predicate candidates.
4. **Retrieval Precision:** Low similarity scores for edge-case facts.
5. **Startup Performance:** Double initialization via Flask debug, model load on CPU.

---

6. Roadmap and Future Outlook

6.1 Short-Term (Next 4 Weeks)

- **Performance Tuning:**
- Persist RAG vector index (Faiss)
- Lazy-load embeddings, disable Flask debug reload
- Reduce front-end polling frequency
- **Parser Enhancements:**
- Implement regex post-processing fixes for syntax
- Expand regex patterns and retrain entity recognizer
- **Ontology Improvements:**
- Build ranking/scoring for inferred predicates
- Add synonyms and multilingual labels for instance lookup

6.2 Mid-Term (1–3 Months)

- **Easy/Scientific Mode:** Implement mode switch with tailored UI/UX.
- **CI/CD Integration:** Automate end-to-end tests (Hybrid Parser, Ontology, KB).
- **Monitoring & Alerts:** Integrate Prometheus and Grafana dashboards for production metrics.
- **Security Hardening:** Code signing, containerization (Docker), and AV false-positive mitigation.

6.3 Long-Term (6+ Months)

- **Production Deployment:** Migrate to Gunicorn/Uvicorn behind Nginx, HTTPS, scaling via Kubernetes.
- **Advanced Features:**
- Distributed KB updates, collaborative editing
- Real-time streaming of RAG chunks
- Integration of additional provers (e.g., Lean, Coq)
- **Academic Publication:** Document architecture and validation results for conference/journal submission.

---

7. Conclusion

The HAK-GAL Suite has achieved a robust alpha implementation, validating core concepts of hybrid retrieval and formal reasoning. Immediate focus on performance optimization, parser robustness, and semantic ranking will prepare the platform for production use. A phased roadmap ensures gradual maturation toward a scalable, trust-worthy AI toolbox for both students and researchers.