

RunInk Project Summary

Project Overview

RunInk is a command-line tool for parsing contract, conf, dsl, and herd files to build and execute Directed Acyclic Graphs (DAGs). This implementation provides a skeleton structure that can be extended to implement the full functionality.

Project Structure

```
runink_demo/
├── README.md
├── SUMMARY.md
├── examples/
│   ├── files/
│   └── run_example.sh
└── src/
    ├── cmd/
    │   ├── execute.go
    │   ├── root.go
    │   ├── run.go
    │   └── run_test.go
    ├── internal/
    │   ├── engine/
    │   │   ├── dag.go
    │   │   ├── graph.go
    │   │   └── parser.go
    │   └── parser/
    │       ├── conf.go
    │       ├── contract.go
    │       ├── dsl.go
    │       └── herd.go
    ├── go.mod
    ├── go.sum
    └── main.go
```

Components

- 1. Command Line Interface (CLI)**
 - Built using the Cobra CLI framework
 - Provides a `run` command with flags for input files
 - Supports verbose output
- 2. Parser Package**
 - Skeleton implementations for parsing different file types:
 - Contract files (`.contract`)
 - Configuration files (`.conf`)
 - Domain Specific Language files (`.dsl`)
 - Herd files (`.herd`)
- 3. Engine Package**
 - DAG configuration structure
 - Graph implementation for representing the DAG
 - Execution logic for running the DAG

Next Steps

The current implementation provides a skeleton structure that can be extended to implement the full functionality. The following steps are recommended for further development:

1. **Implement File Parsers**
 - Develop parsers for each file type
 - Define data structures to represent the parsed content
2. **Implement DAG Builder**
 - Create logic to build the DAG based on the parsed files
 - Define node relationships and dependencies
3. **Implement DAG Execution**
 - Implement topological sort for execution order
 - Add error handling and recovery mechanisms
4. **Add Testing**
 - Create unit tests for each component
 - Add integration tests for the entire workflow
5. **Improve Documentation**
 - Add detailed API documentation
 - Create user guides and examples

Usage

```
# Build the tool
cd ~/runink_demo/src
go build -o runink

# Run with required files
./runink run --contract file.contract --conf file.conf --dsl file.dsl

# Run with optional herd file
./runink run --contract file.contract --conf file.conf --dsl file.dsl --herd file.herd

# Run with verbose output
./runink run --contract file.contract --conf file.conf --dsl file.dsl --verbose
```

Example

An example script is provided in `examples/run_example.sh` that demonstrates how to use the tool with the sample files.