

Conceptual Architecture for a Simple Sudoku Game

Overview of Components and Connectors

The architecture for the Sudoku game will be divided into three main components:

1. the user interface
2. the verification component
3. the data model/data store

give me a picture or diagram to go with

Each of these three components will have its own type of connectors to match: *text.*

1. event connectors to handle user input
2. procedure connectors to handle logic
3. data access connectors to handle state/model requests

You can probably copy a lunar (under example)

another picture would help
The user interface will display a graphical representation of the Sudoku puzzle. It will respond to user manipulation via event connectors. A user will select an available square and type a number to fill in that square. When a user wishes to validate an input or set of inputs, the user will click the Verify/Solve button.

The verification component will handle the logic to check if each user move is viable to complete the puzzle. It will use procedure connectors, in particular method calls. After the user presses the Verify/Solve button, the verification component will check the correctness of the user's input. Each square will change its presentation if its input is incorrect.

The data store will contain the information representing the Sudoku puzzle. It will be changed with data access connectors. This component will hold the various types of available Sudoku puzzles, from easy to difficult.

→ Is portability a major issue, or a side-effect?

Basic Game Design Principles

The Sudoku game is designed to be played by a single user on a single machine.

The game will be built using Java 1.7, so it can be played on machines with any operating system that has a viable Java virtual machine implementation.

Other items:

1. Rationale for decisions
2. Identify cross-cutting/arch. significant concerns.
3. Show an architecture that solves a problem and has Conceptual Integrity

LOGICAL ARCHITECTURE DOCUMENT

Lori Gildersleeve
Cody Nelson
Matthew Olivetti
Jennifer Martin

This I would
consider to be
Conceptual Architecture—
with a little more detail
than a C.A. even

- L.A.
- Show Interfaces
 - Specs for Interfaces
 - Component Specs
 - Component Usage guides
 - Design
 - interactions
 - protocol specs and protocols
 - connectors & mechanisms