MVIS

Mine Vehicle Inspection Simulator

Ben Lucchesi
Samir Patel
Eric Rasmussen

This work has been partially funded by:

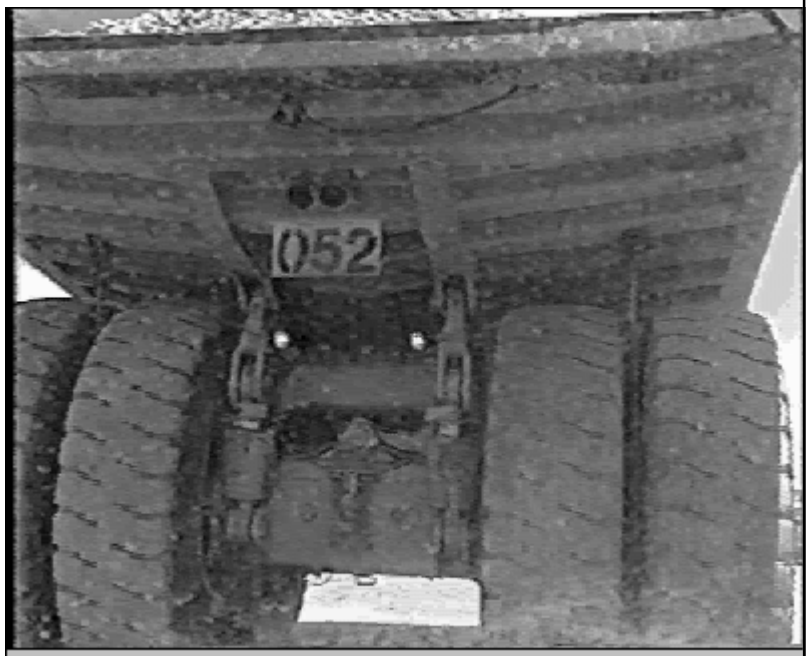
Newmont Gold Co.
Echo Bay Minerals
Nevada Division of Industrial Relations
Mine Safety Health Administration

Special Thanks

- Dr. Fred Harris
  - UNR Department of Computer Science
  - project sponsor, technical advisor
- Dr. Pierre Mousset-Jones
  - UNR Mackay School of Mine
  - project sponsor, technical advisor
- Dr. Mohammed Fayad
  - UNR Department of Computer Science
  - senior projects coordinator

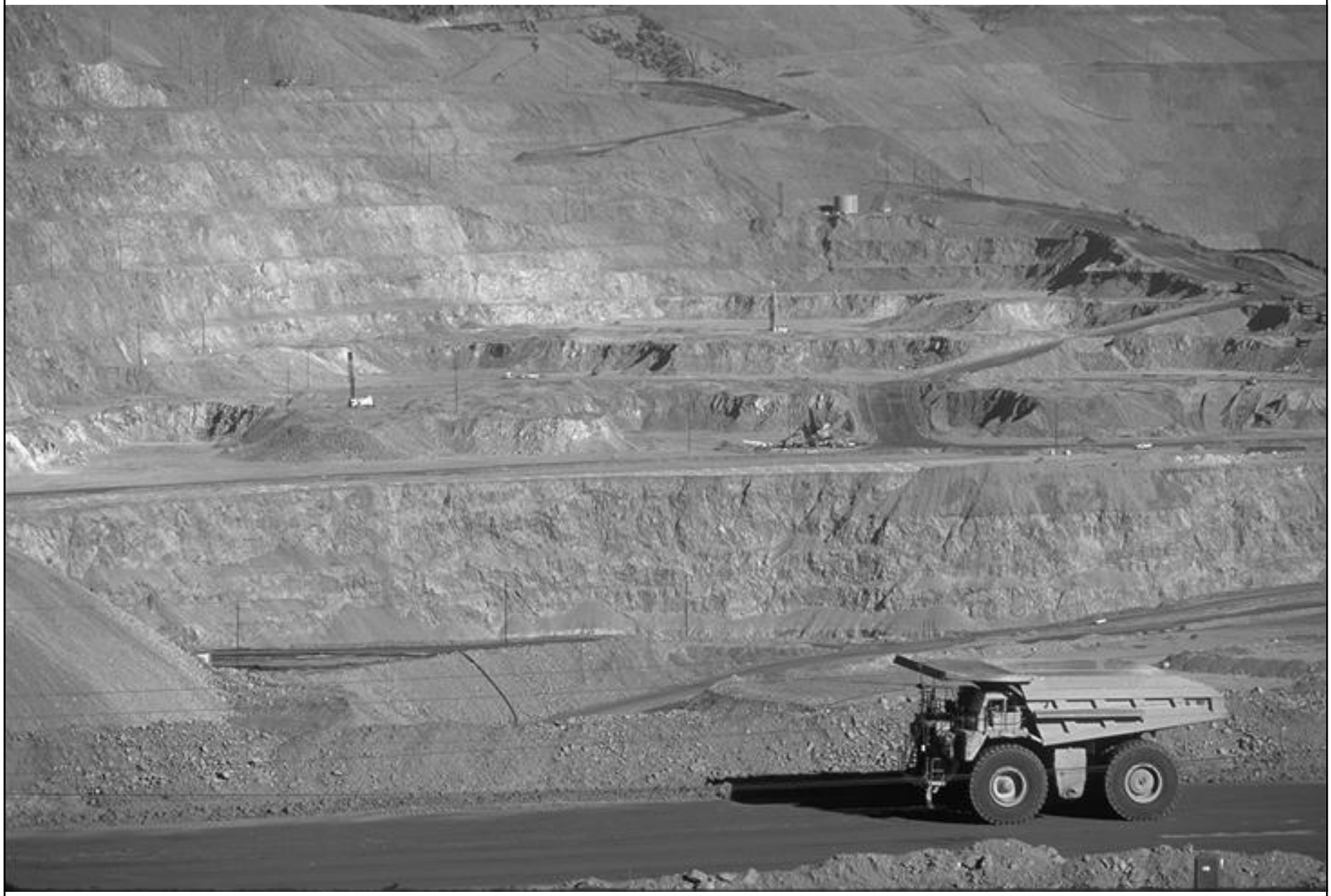
Introduction: Powered Haulage used in surface mining industry

- Common types of vehicles
  - Trackhoes
  - Backhoes
  - Off-Highway Haul Trucks
  - Front Loaders
- Uses for
  - Digging Ore
  - moving topsoil and overburden
  - Transporting Ore to crushers
  - Watering haul roads



gold.avi

EX2500
HITACHI



Pre-Operational Vehicle Inspection

- When?
  - performed by every driver before shift begins
- Why?
  - identify common problems with vehicle
    - preventative measure reduces vehicle damage
    - identify safety hazards
- Benefits:
  - increased productivity due to less down time
  - increased vehicle life-span
  - reduced workplace accidents and worker injury

Training Vehicle Operators: Expensive but Necessary

- Downside to training
  - man hours for training
    - qualified trainer available
    - employee not working while training
  - equipment for training
    - truck taken out of production
- Upside to training
  - increased productivity
    - less downtime due to equipment failure
    - less downtime due to employee injury

Current Training Methods

- Written Material
  - pamphlets
  - training manuals
- Training Videos
- Seminars with trainer
- Live inspection of truck with trainer

Deficiency of these Training methods

- Books and videos are static learning material
- Memorization of process without experience

… A new training method is needed

- People learn by doing = Interactive training
- Training aid is able to simulate non-static many situations
- Non-static + Interactive training = Experience

Project Goal

- Provide an alternative method of training through simulation
  - focus less on memorizing and more on doing
- Make the simulation interactive
  - make the trainee perform and as well as evaluate performance
- Make the simulation configurable
  - don’t give the trainee the benefit of memorizing the training simulation - change it every time
  - help the trainee understand how their training will apply to the real world through experience

Project Overview

- Virtual Reality (VR) Simulation
  - 3D graphics to simulate Pre-Operational Vehicle Inspection
    - 3D image of vehicle in a scene
    - input device to move naturally in scene
- Interactive
  - trainee practices inspecting a vehicle in virtual reality
- Configurable
  - type vehicle can be changed
  - configure vehicle on a per-part basis
- Evaluate trainee's performance
  - what parts were/weren't inspected?
  - what action was performed to inspect a part?

Introduction into Virtual Reality VR

- What is Virtual reality
- What does Virtual reality try to accomplish
- What is required to accomplish Virtual Reality

The goal of Virtual Reality

- To completely immerse the user into this virtual world
- To make this virtual world as real as possible
- To make this world extremely interactive

The requirements to create Virtual Reality

- Hardware requirements
  - Hardware for the generation of the virtual world
  - Hardware for the interaction with the virtual world
- Software requirements
  - Software for the generation of the virtual world
  - Software for the interaction with the virtual world

The hardware requirements

- For the graphics generation of the virtual world
  - Graphics accelerator card
- For the graphics interaction with the virtual world
  - Input
    - 6 degree of freedom input device, SpaceOrb
  - Output
    - Flat Display

The software requirements

- For the graphics generation of the virtual world
  - OpenInventor API
  - OpenGL API
- For the graphics interaction with the virtual world
  - Microsoft win32 API
- Reasons for choosing OpenInventor
  - It is windowing and hardware system independent
  - Focuses on scene management rather than scene drawing

How Is the Simulation Used?

- Scenes and scenarios
- The inspection process
- Trainee performance evaluation

Scenes and Scenarios

- What is a scene?
  - Defines the vehicle used in the simulation
  - Defined by a single file (.scn)
- What is a scenario?
  - Defines the configuration of the vehicle
  - Multiple scenarios defined by a single file (.scr)

The Inspection Process

- Scene file is opened
  - Scenario is either selected from a list
- Inspection is started
  - Trainee inspects parts
  - Trainee records actions taken
- Inspection is stopped
- Inspection results are evaluated

Trainee Performance Evaluation

- Certified trainer reviews printed inspection results
  - List of damaged or defective parts
  - Action(s) performed during the inspection

Designing the MVIS

- Design Philosophy
- Object interaction

Design Philosophy

- Include no windowing specific code in primitive classes
- Why?
  - simplify object design
  - keep as much code Windowing system independent as possible

Layered approach

Primitive Objects

API used: OpenInventor

65%

Windowing/primitive glue

API used: OpenInventor

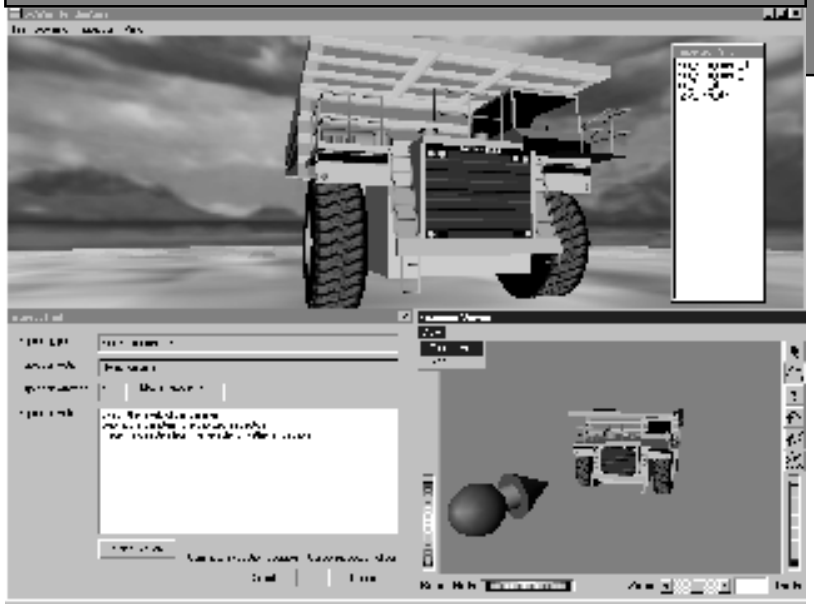
10%

Main program & Windowing

API used: Win32 or other

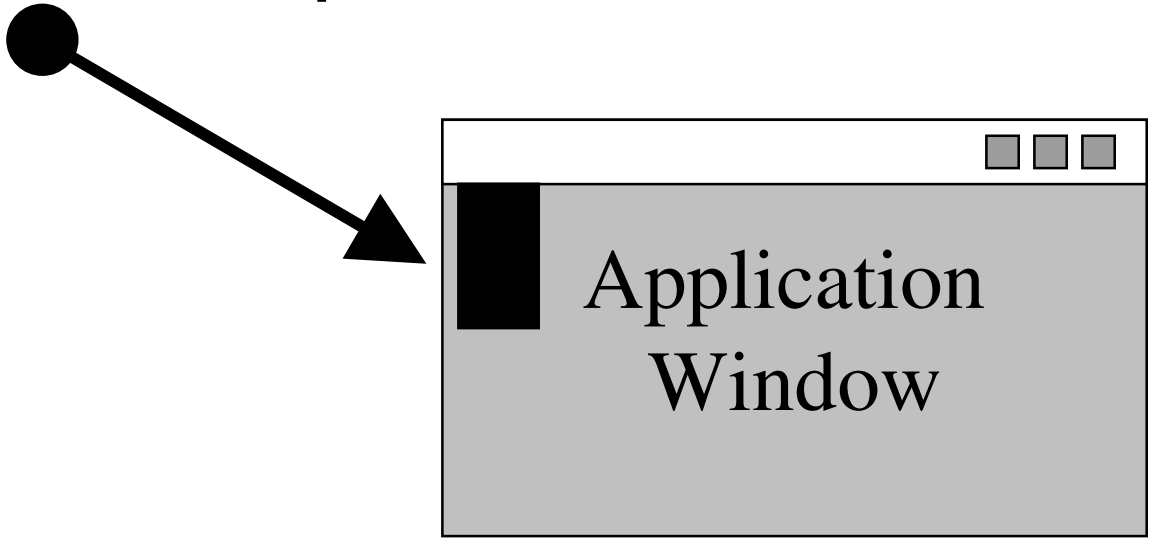
25%

Final Application

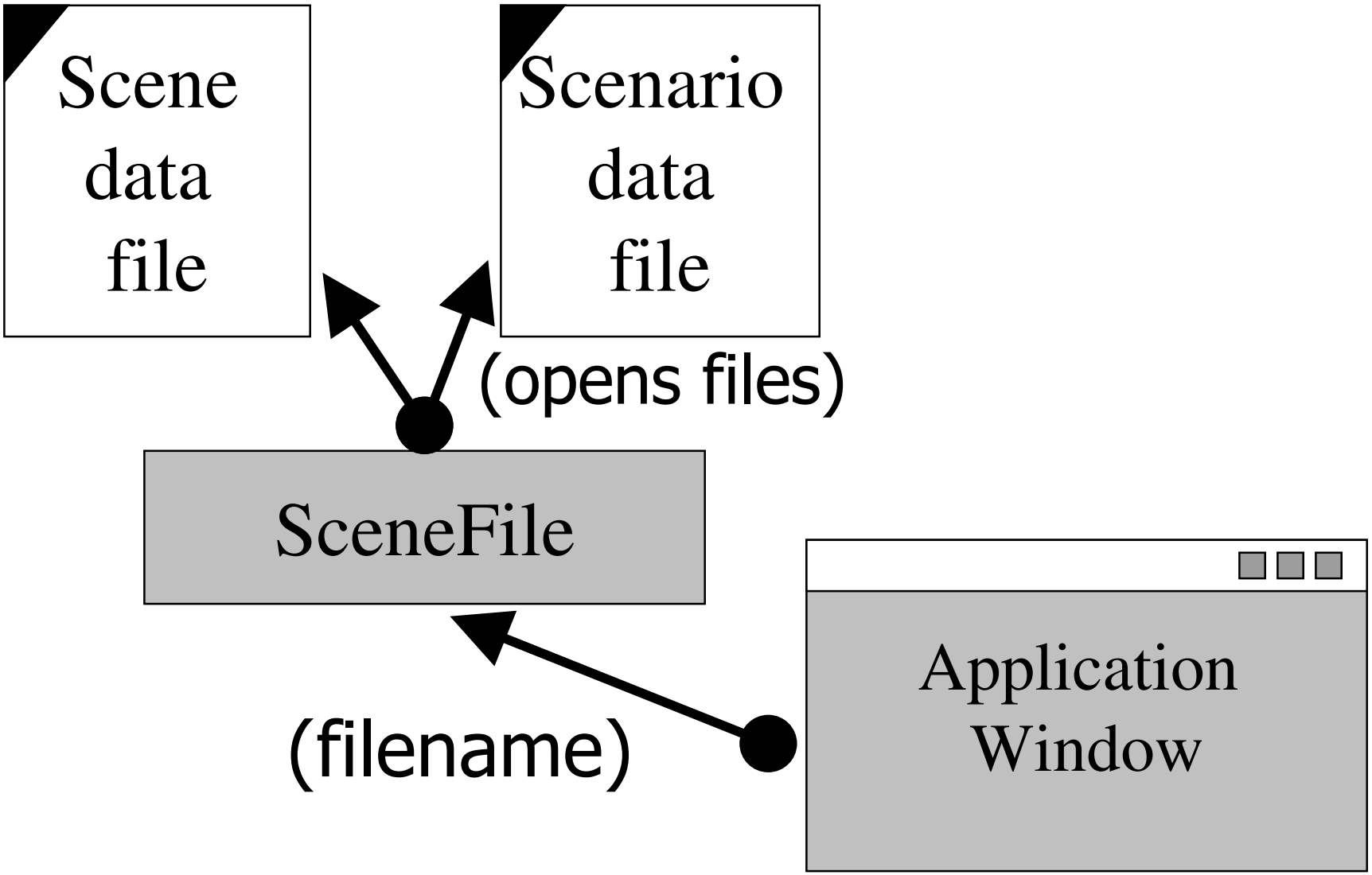


Object interaction: opening a file

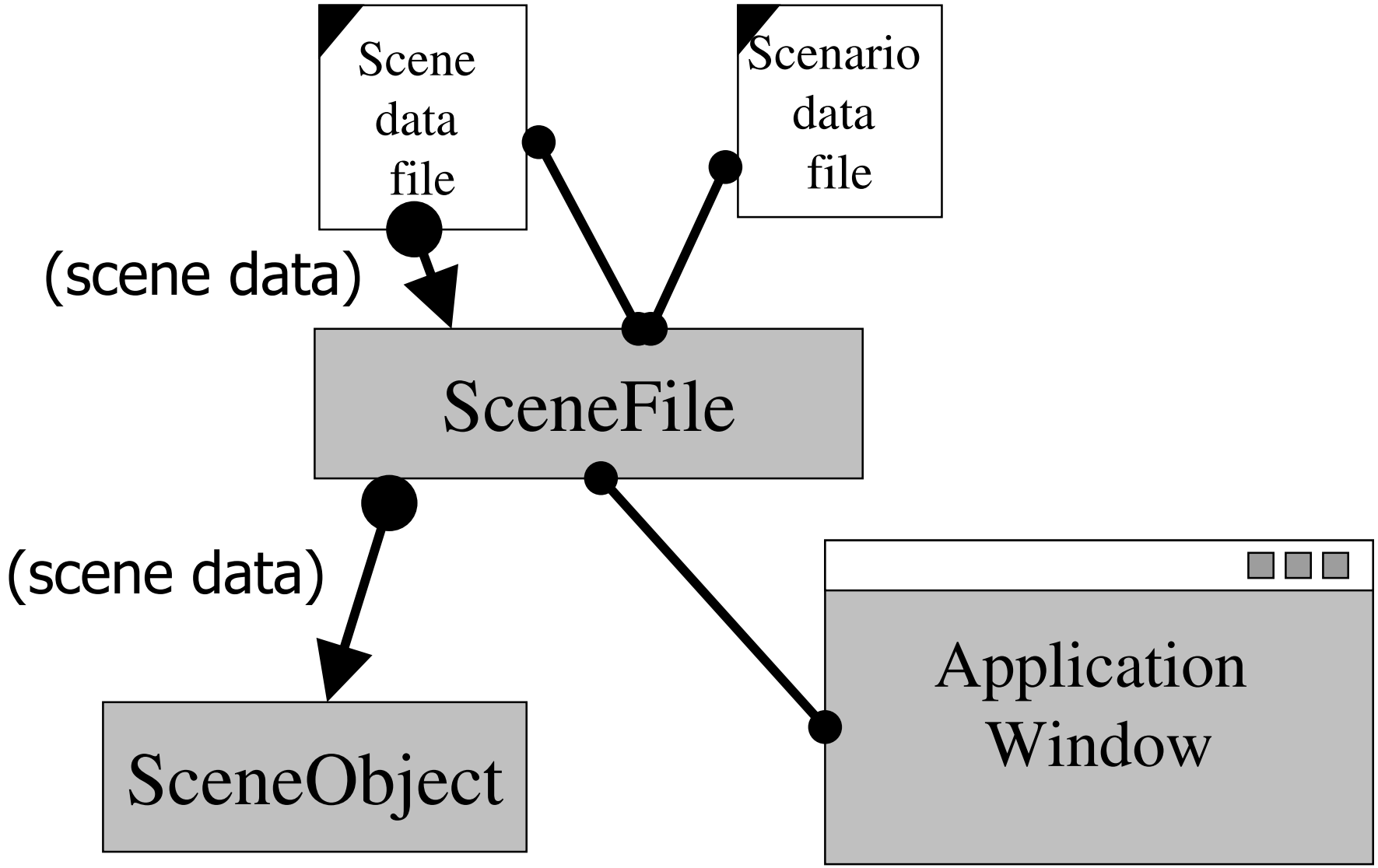
User selects file open from menu



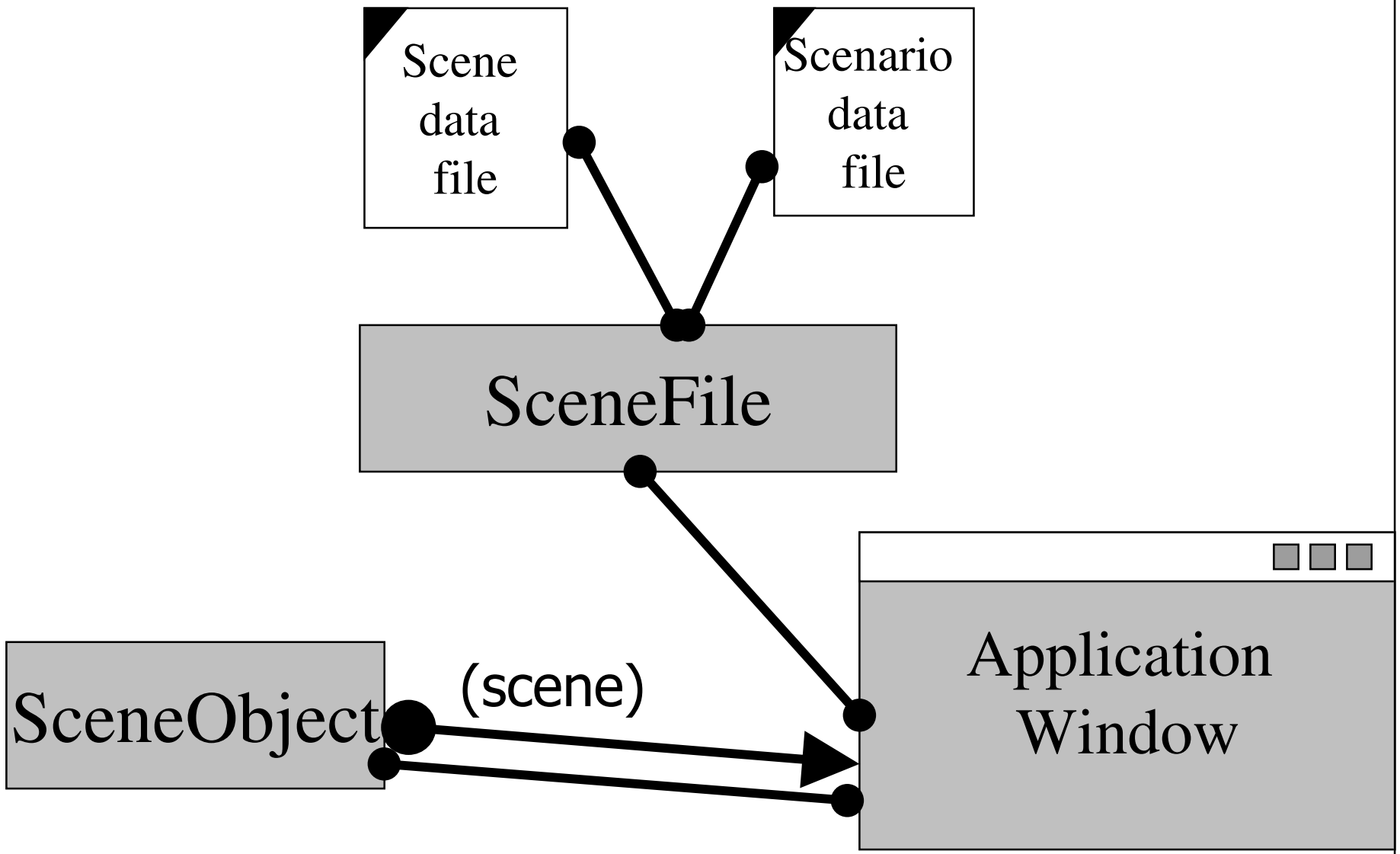
SceneFile Object created



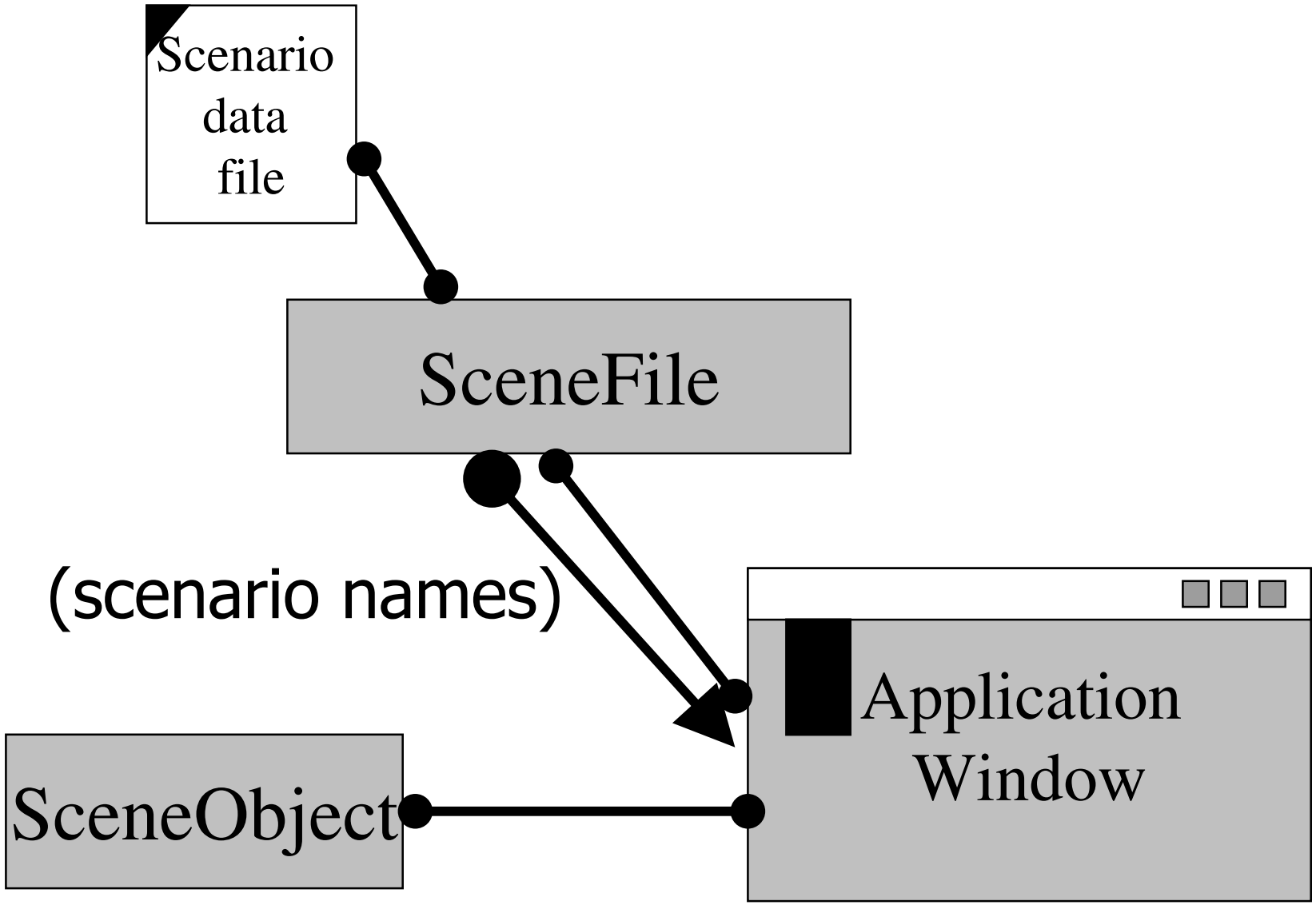
Create SceneObject



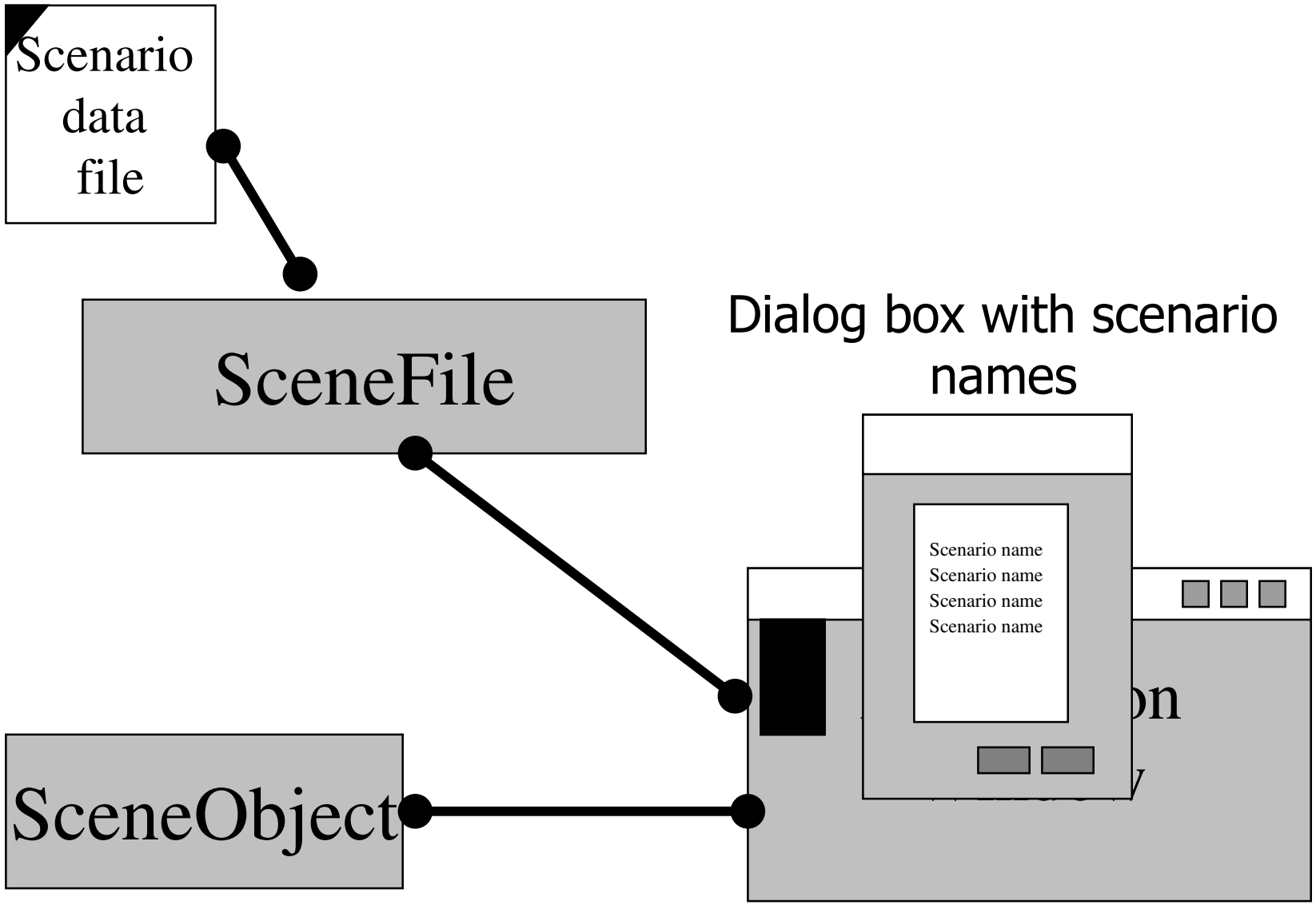
Viewing scene

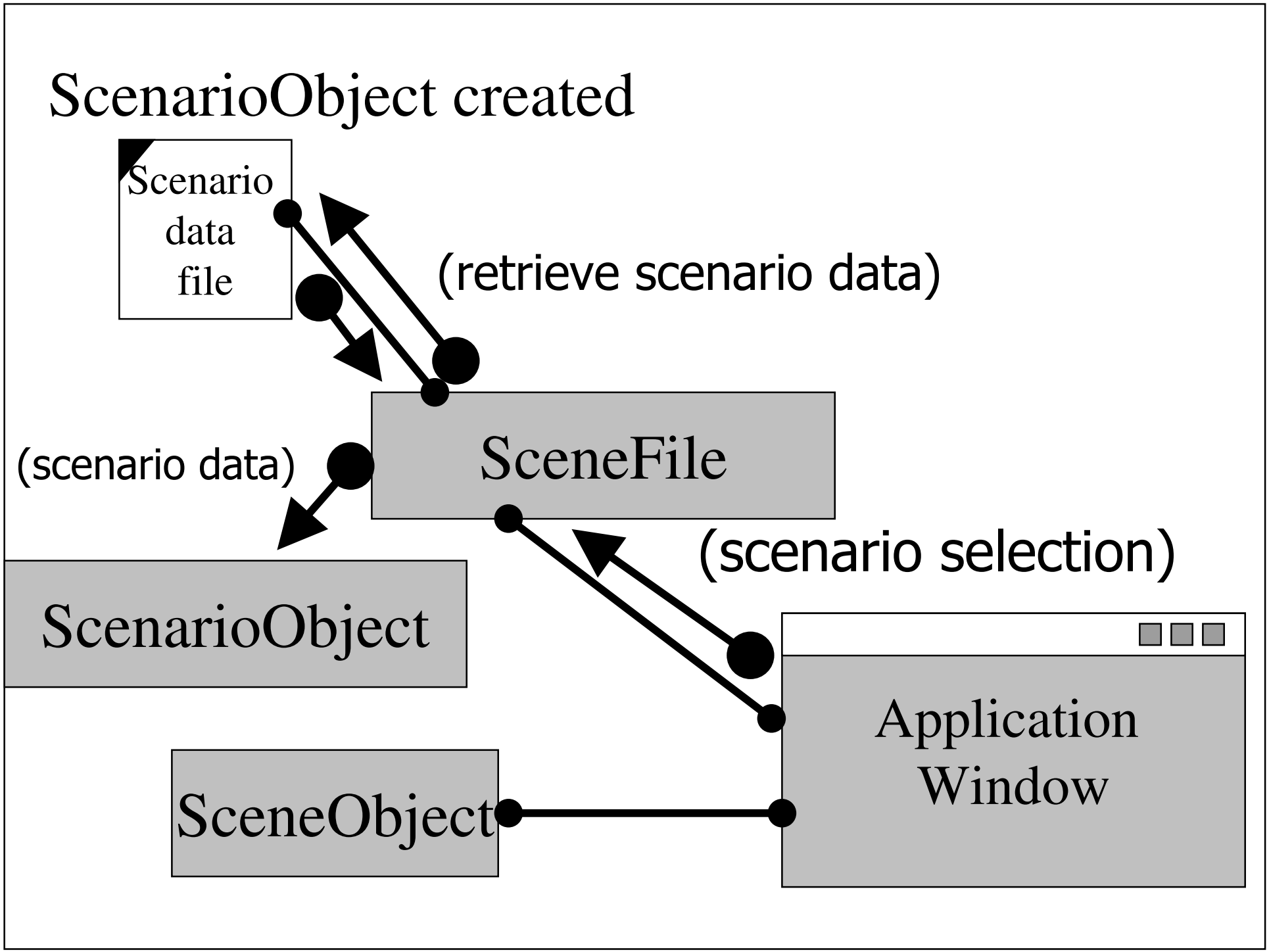


selecting scenario

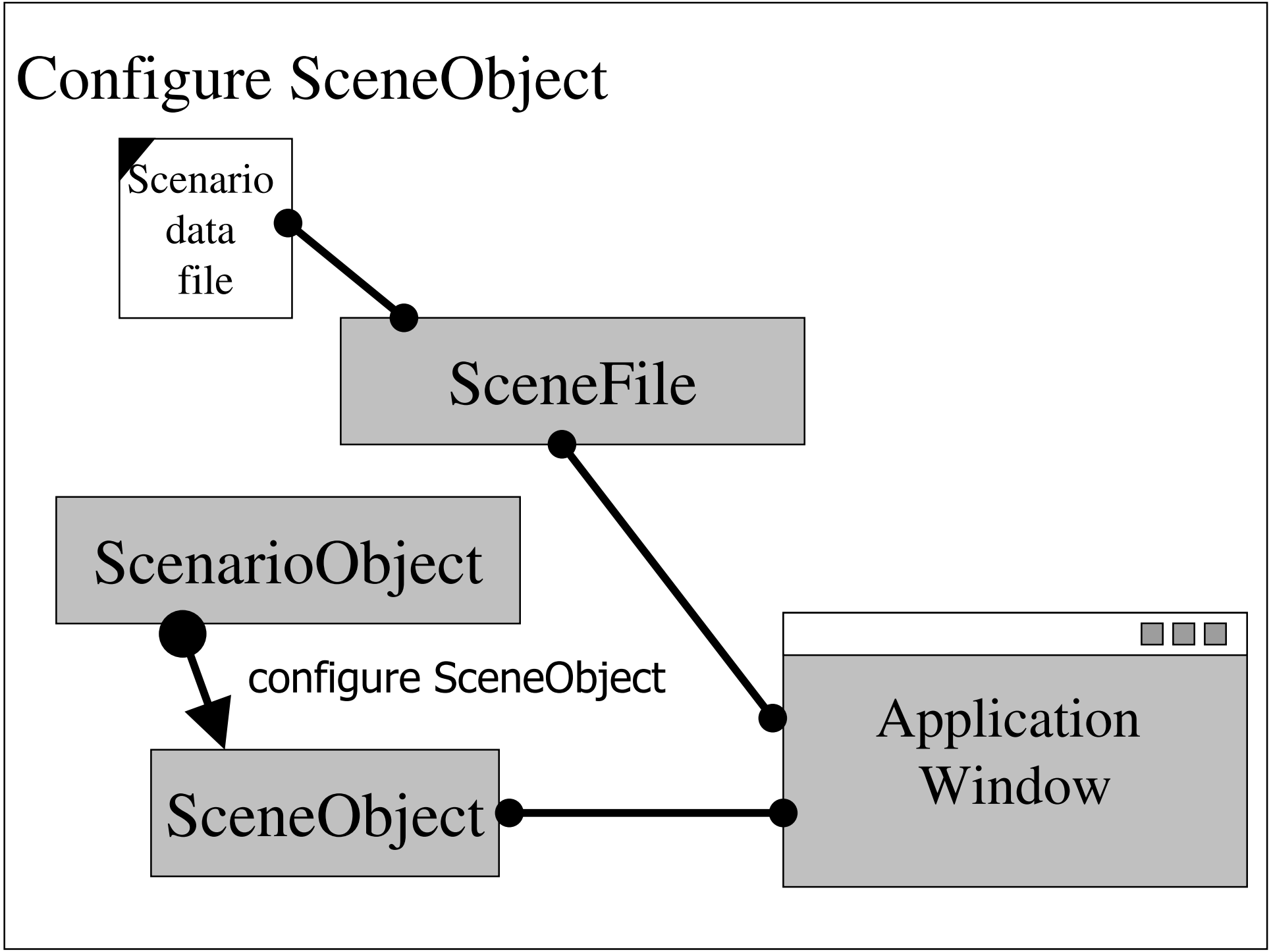


Dialog displays scenario names

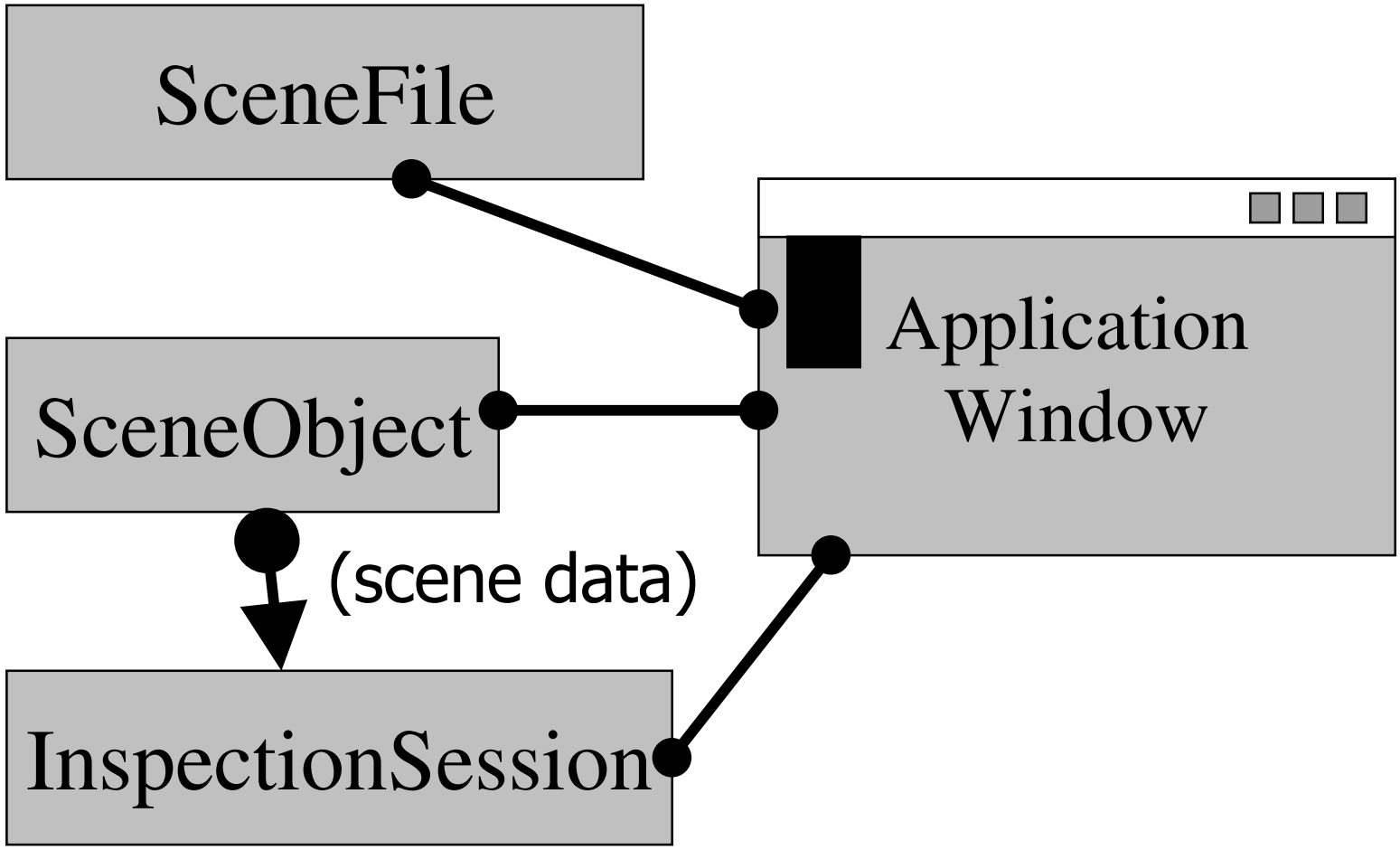



ScenarioObject created
Scenario data file
(retrieve scenario data)
(scenario data)
SceneFile
(scenario selection)
ScenarioObject
Application Window
SceneObject

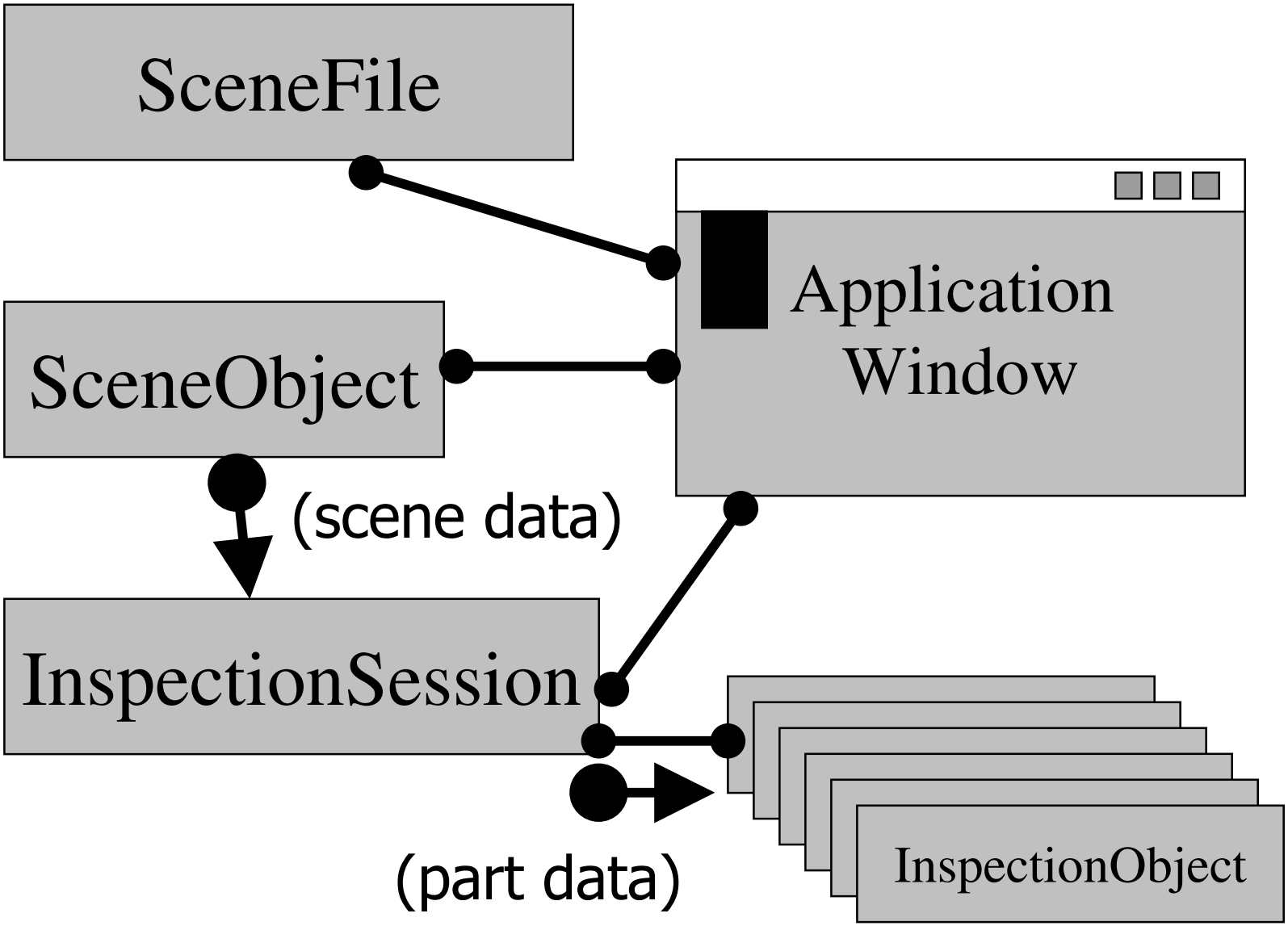
Configure SceneObject



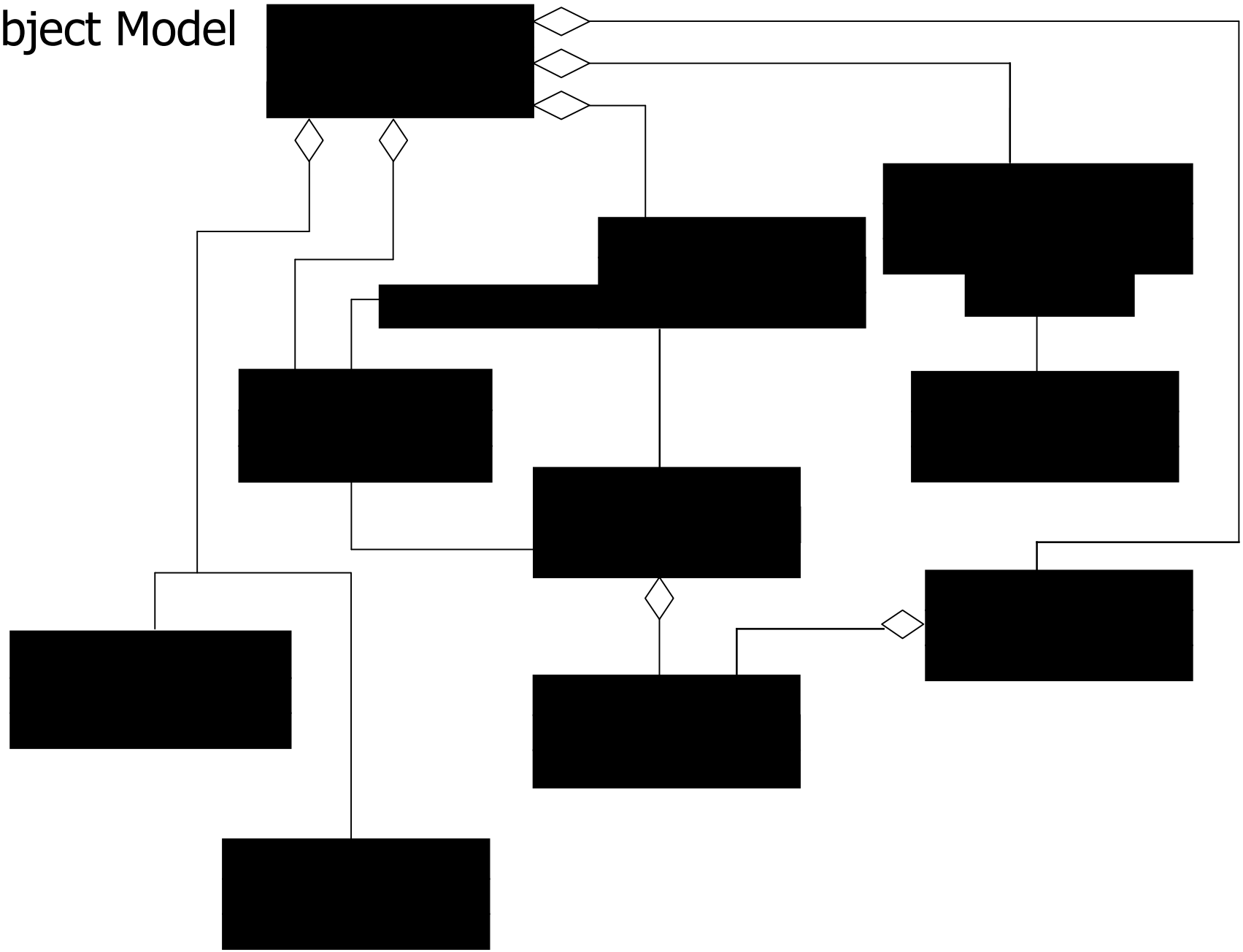
Start inspection: Initialize InspectionSession



InspectionSession initialization (cont.)



Object Model



Time analysis

- Expected time (designing/source code doc./coding/testing): 300 hours
  - designing: 100 hours
  - coding: 110 hours
  - testing: 90 hours

Time analysis continued

- Actual time (designing/source code doc./coding/testing): 510 hours
  - designing: 40
  - coding: 300
  - testing: 170

Breakdown of actual time

- base application:
  - designing: 40 hours
  - coding: 170 hours
  - testing: 90 hours
- user interface modifications and additions
  - 210 additional hours

additional time not included in analysis

- documentation and presentation
  - 150 hours
- total actual time: 660 hours

Experience

- We learned about
  - challenges in estimating development time
  - challenges of developing robust and extensible file formats
  - challenges of integrating multiple APIs to develop a single application
  - gained in-depth knowledge of the Win32 and OpenInventor APIs
  - advantages of using an OO design methodology for developing large projects

Project Demonstration