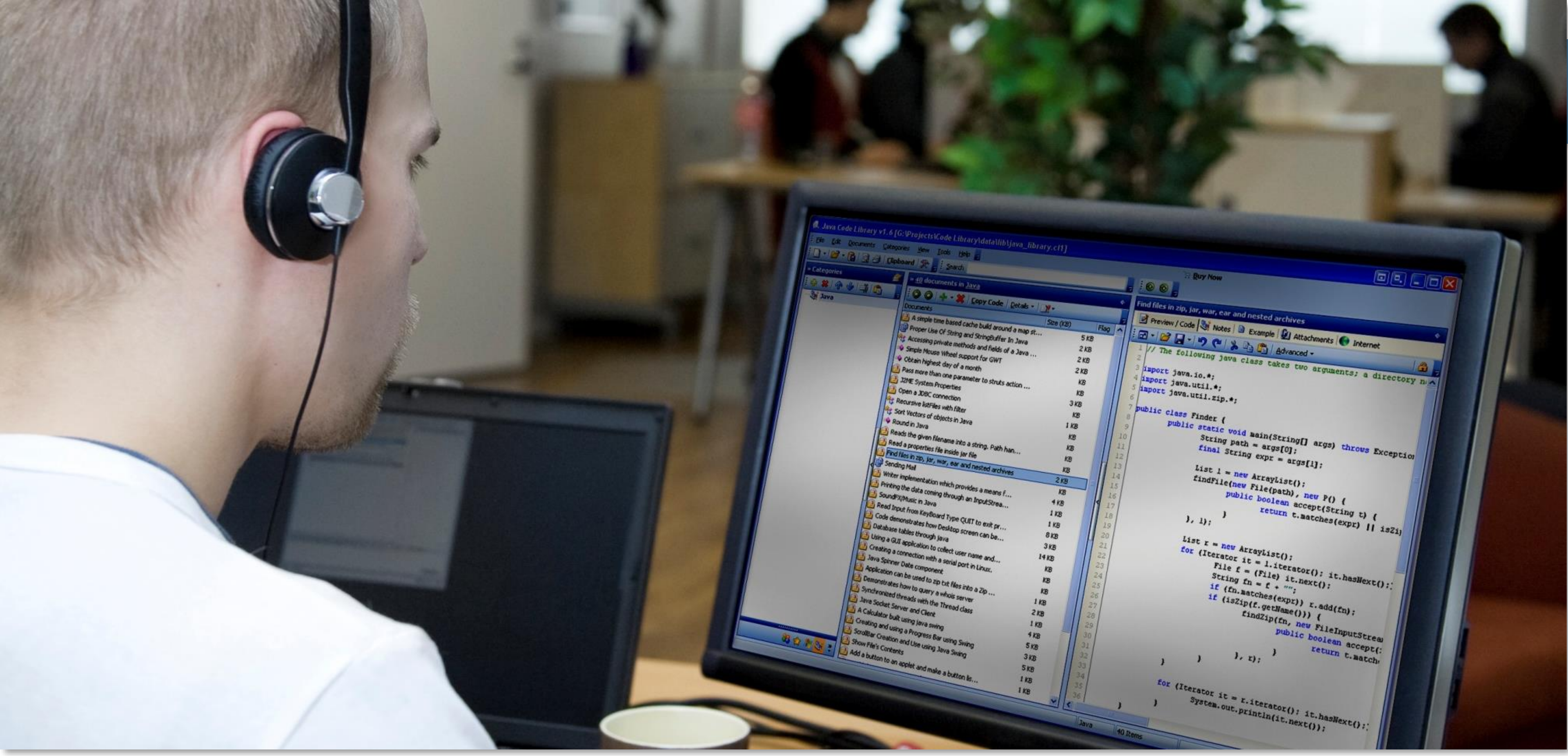


Domain Models

Software development tools

Topics and Agenda

- Lecture/Class Discussion
  - What is the Domain Model?
  - Why Architects Need to Understand Domain
  - How to Model the Domain
- Exercise 13: Identifying View Types
- Reflection

Course Progress

- Project 01: Risk Assessment
- ➢ Project 02: Initial Design
  - ~~Class 10: Abstraction~~
  - ~~Class 11: Creating Software Abstractions~~
  - ~~Class 12: Using Software Abstractions~~
  - ~~Assessment 04~~
  - ➢ Class 13: Domain Models
  - Class 14: Information Model
  - Class 15: Actors Vs. Individuals
  - Assessment 05
  - Class 16: The Design Model – Boundary Models and Internals Models
  - Class 17: Context Diagrams, Modeling Components
  - Class 18: Design Decisions
  - Assessment 06
- Project 03: Creating Models
- Project 04: Patterns and Styles
- Project 05: Using the Architecture

What Is A Domain Model?

- Domain Modeling helps refine the requirements by describing the problem space
  - Entities – the things involved in the problem
  - Relations – the ways in which the entities are associated
  - Behaviors – how things are done
- Consider an on-line music store. What are some :
  - Entities?
  - Relations?
  - Behaviors?
- Clarifying the domain is critical to building the right system.

Definition

- A **domain model** expresses enduring truths about a domain. Domain models are also called *concept models, conceptual models, or abstract models*, but the idea is the same: to express the details of the domain that are not related to the system's implementation.

- A Domain Model illustrates meaningful concepts in a problem domain.
- It is a representation of real-world things, not software components.
- It is a set of static structure diagrams; no operations are defined.
- It may show:
  - Concepts / entities [Modeled as classes and objects]
  - associations between concepts
  - attributes of concepts

Do I Need to Model the Domain?

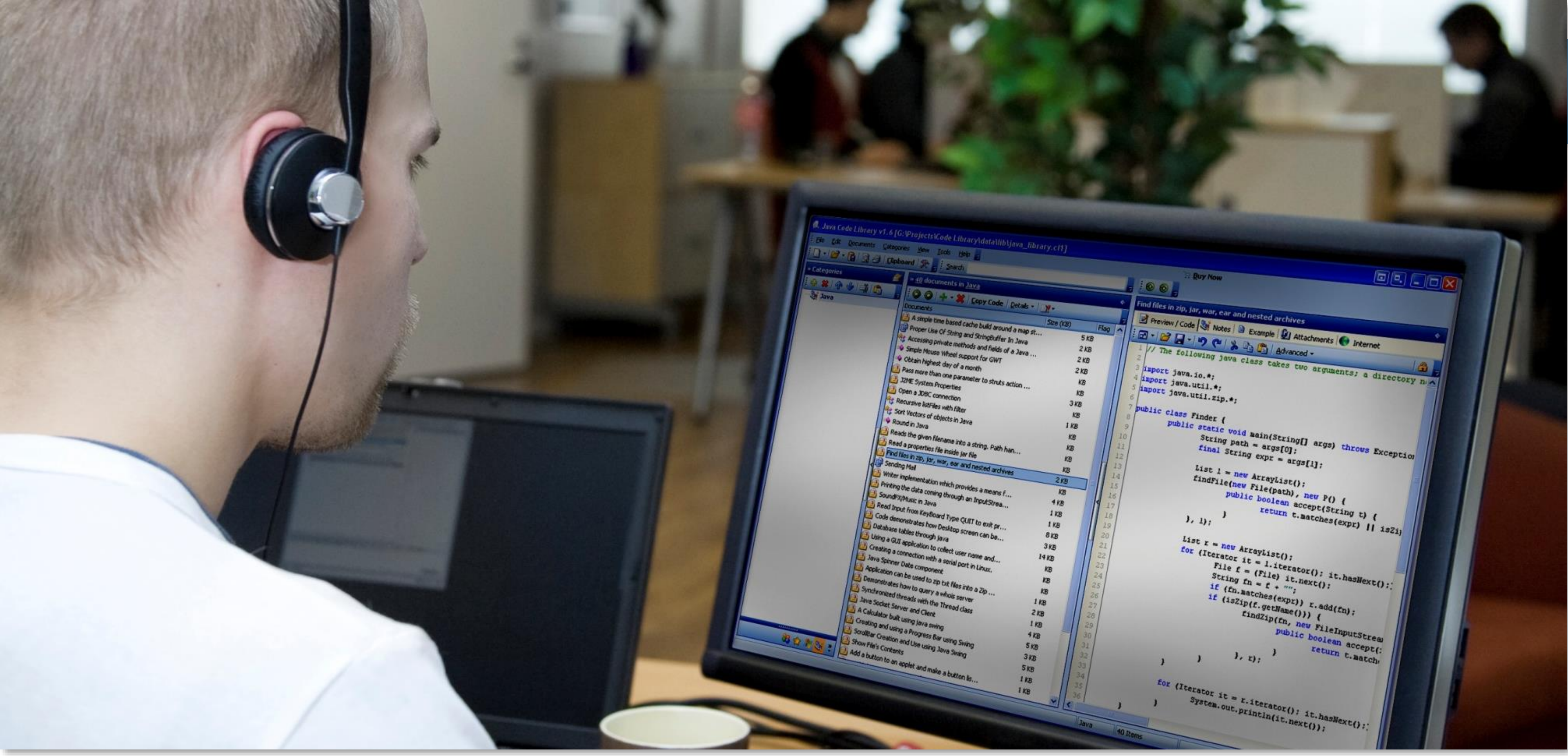
- The quick answer is almost always yes!
  - Maybe not if you have just finished another project in the same domain with the same customer
  - Otherwise, do it!
- Benefits
  - The process adds clarity
  - Consider the the clarification we got from exploring the on-line store in the previous slide

But, When to Stop?

- The quick answer – that depends!
- Domain models could be infinitely deep – you don’t have time for that!
- When you ask yourself a question, consider whether knowing the answer will impact that design.
  - If it’s not related to one of your architectural risks, it’s probably not worth including in the domain model
  - As Fairbanks says, consider whether it is an “enduring truth” about the domain.

Modeling the Domain

- Textual Information Models
  - Simple natural language description of the entities in the domain
  - Descriptions should include information about the relationships among the entities.
- Graphical Information Models
  - Box and line diagrams that focus on information about the relationships
- Snapshots
  - Shows relationships among entities in the domain for a specific case, or instance
- Functionality Scenarios
  - We looked at these when we discussed use cases



Information Model

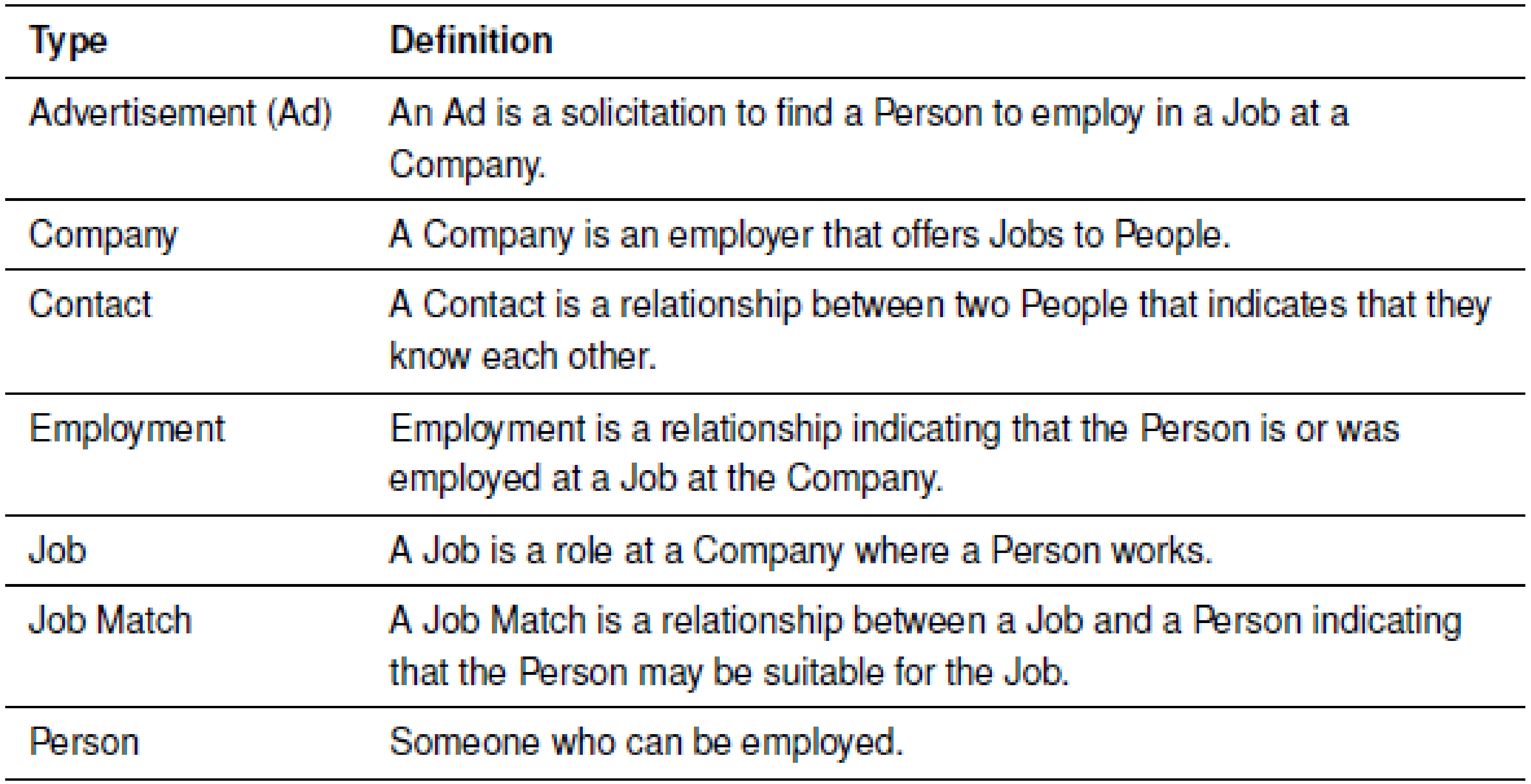
Software development tools

Information Model

Describes in detail the things in a domain—types of people, accounts, books, pieces of equipment, etc. —and the relationships among them.

- What are some of the things you think belong in the ATM domain?
- Identifying and defining these things helps you refine your understanding of the problem.
- Information models can help confirm your understanding with the client.
- Words of Wisdom: When you find yourself making assumptions about the things in the information model, stop and hammer out the details.

Information Model



Type	Definition
Advertisement (Ad)	An Ad is a solicitation to find a Person to employ in a Job at a Company.
Company	A Company is an employer that offers Jobs to People.
Contact	A Contact is a relationship between two People that indicates that they know each other.
Employment	Employment is a relationship indicating that the Person is or was employed at a Job at the Company.
Job	A Job is a role at a Company where a Person works.
Job Match	A Job Match is a relationship between a Job and a Person indicating that the Person may be suitable for the Job.
Person	Someone who can be employed.

Figure 8.1: A textual information model for the job ad and business networking domain. The Yinzer system's design and implementation will need to be consistent with this model of the domain.

Graphical Information Models

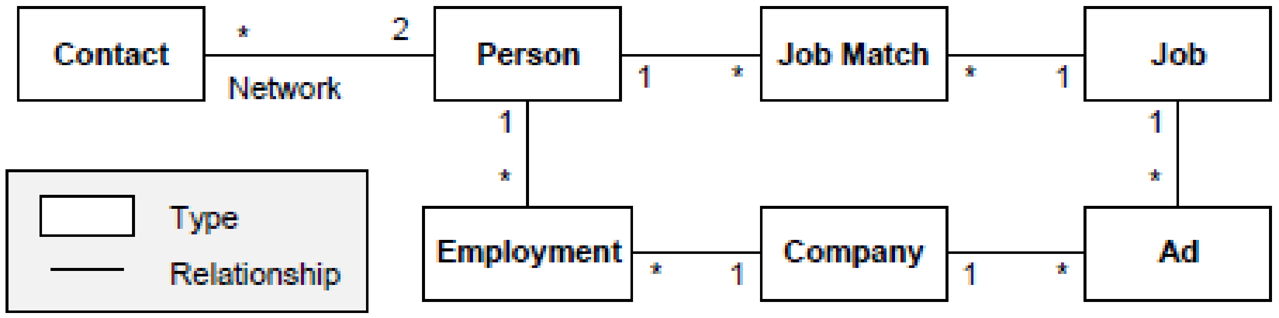


Figure 8.2: An information model for the job advertisement and business networking domain, shown graphically using a UML class diagram.

Exercise Reflection

- If the architect is unfamiliar with the domain, how should she or he approach creating the information domain model?

Snapshots

- It is easy to make mistakes when thinking about information models because they refer to general types, not concrete instances. For example the information model talks about a Person, not Bradley, and a Company, not Widgetron. You can draw a *snapshot, or instance diagram, which shows instances, not types, as seen in Figure*

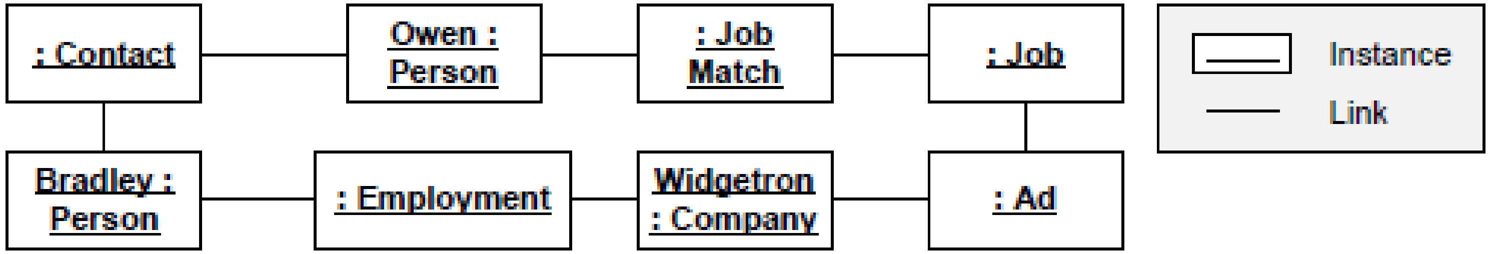


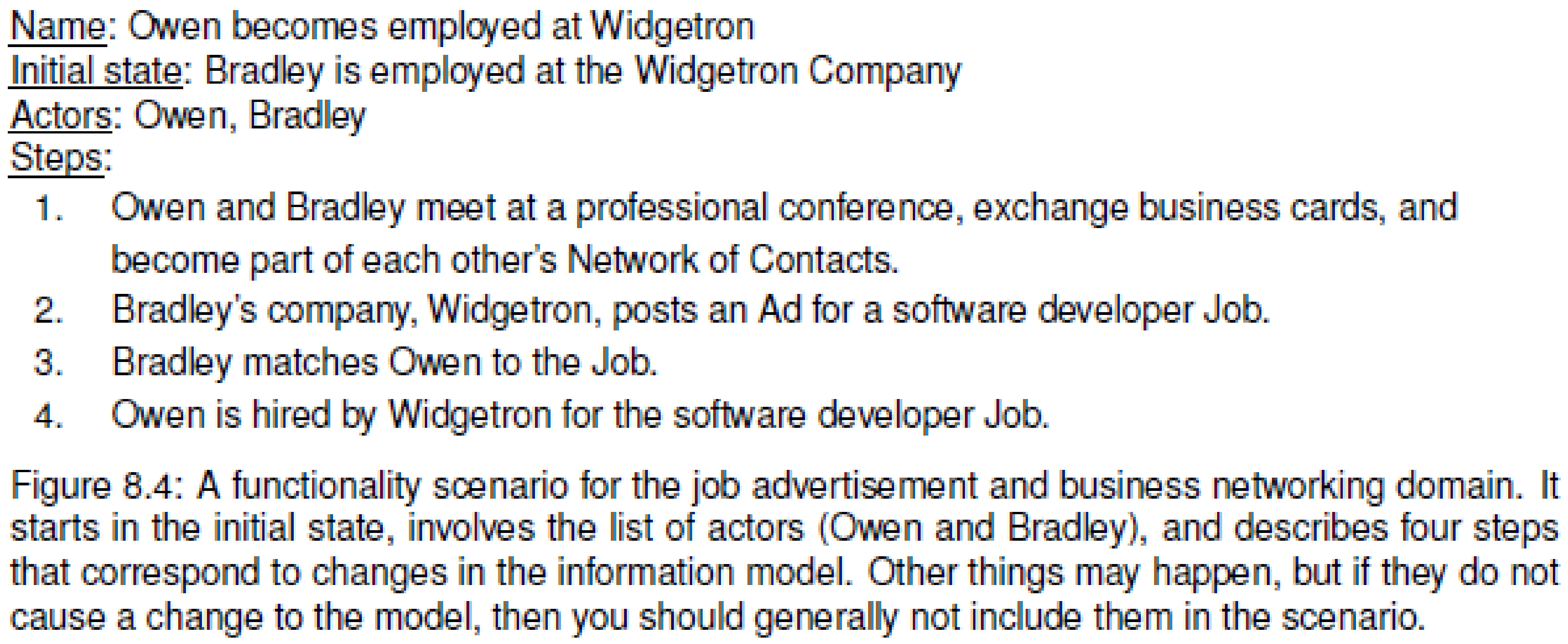
Figure 8.3: A snapshot (i.e., instance diagram) of the information model for the job advertisement and business networking domain. Notice that some instances are named (Bradley and Widgetron) and the rest are anonymous.

Functionality scenarios

- A snapshot expresses how the instances in the model might be linked at an instant in time and the information model expresses all the possible snapshots. What you cannot yet express is how the domain might change from one snapshot to another.
- *Functionality scenarios, also called simply scenarios express* a series of events that cause changes to the information model.
- A functionality scenario uses the vocabulary defined in the information model, like Ads and Contacts. Each scenario starts in an initial state that is written out textually in the scenario. That initial state could equivalently be drawn out as a snapshot. Each step in the scenario changes the model. If you were to draw out a snapshot of the state of the model, it would be different after each step

Functionality scenarios

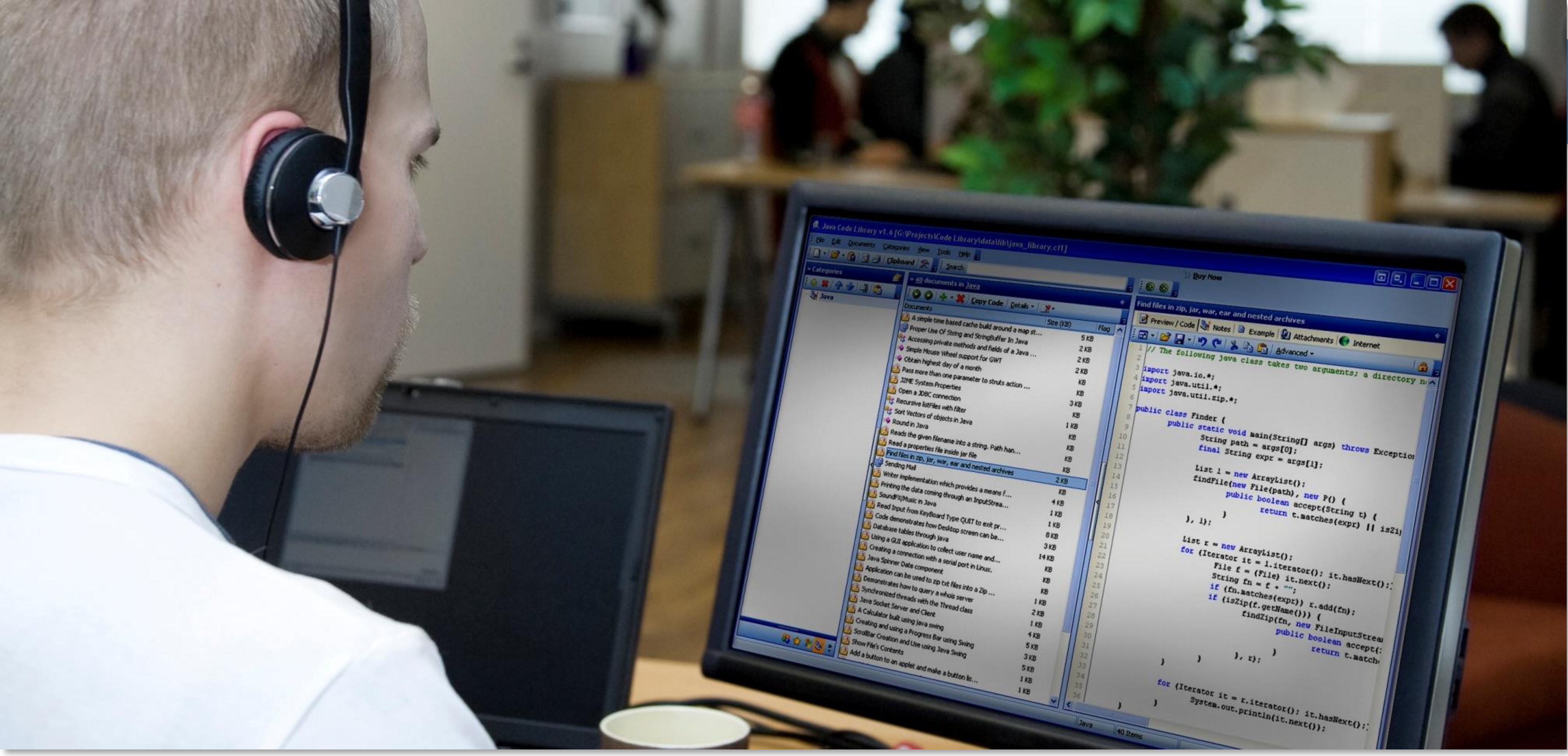
- For example, after the first step in the scenario from Figure 8.4, the snapshot will contain a new Contact instance linking Owen and Bradley.



Name: Owen becomes employed at Widgetron
Initial state: Bradley is employed at the Widgetron Company
Actors: Owen, Bradley
Steps:

1. Owen and Bradley meet at a professional conference, exchange business cards, and become part of each other's Network of Contacts.
2. Bradley's company, Widgetron, posts an Ad for a software developer Job.
3. Bradley matches Owen to the Job.
4. Owen is hired by Widgetron for the software developer Job.

Figure 8.4: A functionality scenario for the job advertisement and business networking domain. It starts in the initial state, involves the list of actors (Owen and Bradley), and describes four steps that correspond to changes in the information model. Other things may happen, but if they do not cause a change to the model, then you should generally not include them in the scenario.



Actors Vs. Individuals

Software development tools

Actors Vs. Individuals

- Can more than one person be the manager of a soccer team at one time?
- Can more than one person be a teacher of a class?
- Does it matter if you think about these questions when you are creating a software system to support what these people do?
- YES, it does!
  - In the case of the teacher, yes, some courses have two teachers!
    - Will the LMS system need to allow both equal access?
    - If so, will it will need to be designed to avoid conflicts if they are working with the system at the same time.
    - Will the grading system need to allow both to submit grades?

Whose Decision Is It?

- Working with the information model is intended to bring questions like this to the surface.
- It is NOT the software architect’s job to decide what the correct answer is!
- These kinds of decisions are business decisions that can have huge impacts, even legal impacts, on how the business functions
- ASK for clarification!

Exercise Reflection

- Do you see any possible issues with the following scenario?

  Sandra is a clerk in charge of taking payments for water bills, and she must log in to the system to the system using a special ID associated with the "clerk" actor for the water bill accounting system.

  The system logs all transactions made including ID of person interacting with the system, what was done, and amount if the transaction involved payment of posting a charge.

  One day Sandra is sick, so Justin uses the "clerk" ID to log in to the system to enter a payment for a customer.