

DOMAIN ENGINEERING OF BUSINESS DOMAIN

Sailaja

Definition of Domain

A domain can be defined by a set of problems or functions that applications in

that domain can solve . A domain can also be characterized by a common

jargon for describing the concepts and problems in that domain .

The term "domain" can be used in several associations :

- business area
- collection of problems (problem domain)
- collection of applications (solution domain)
- area of knowledge with common terminology

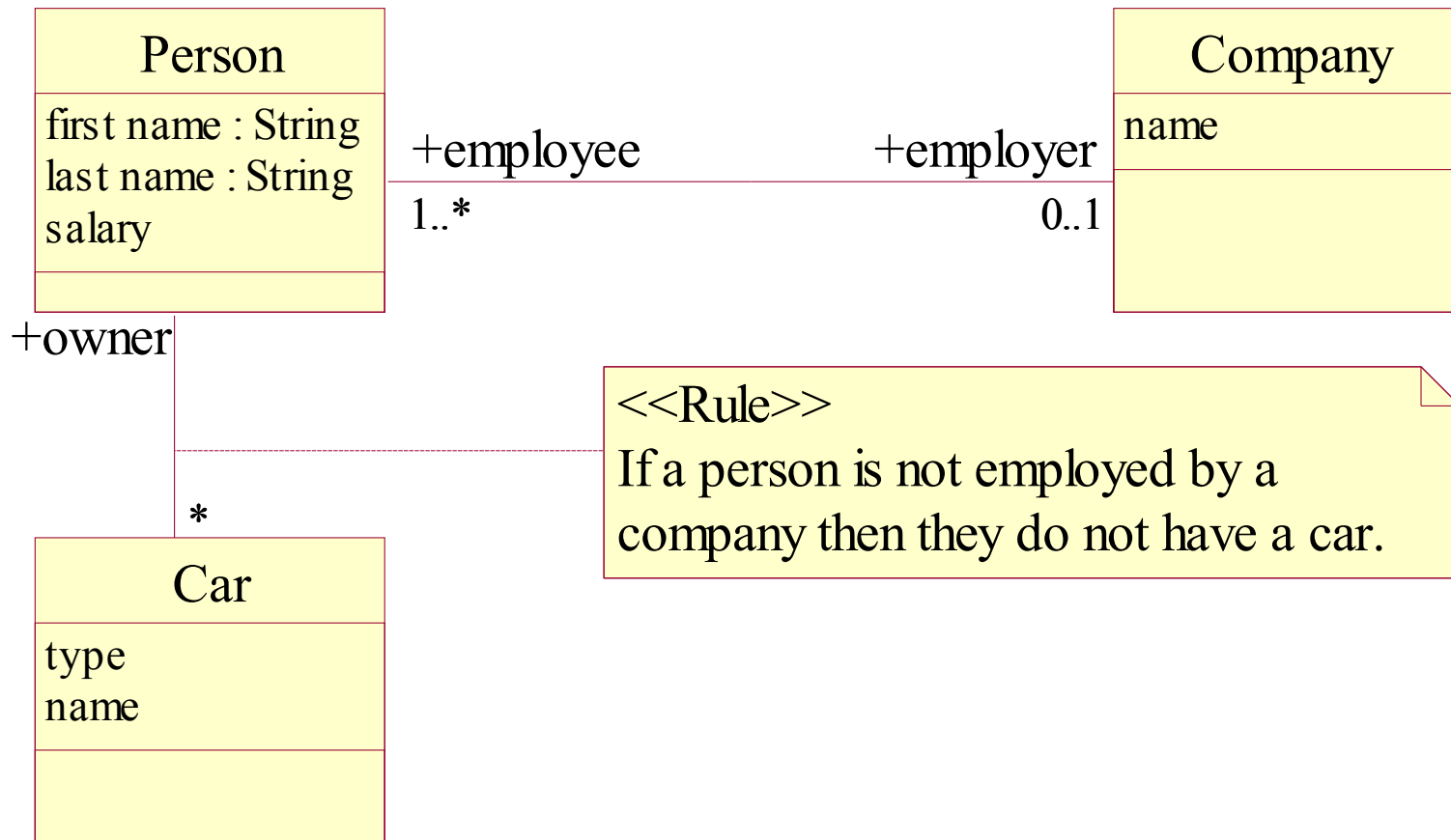
What is a domain model?

- A domain model captures the most important types of objects in the context of the business. The domain model represents the ‘things’ that exist or events that transpire in the business environment.” – I. Jacobsen

Why do a domain model?

- Gives a conceptual framework of the things in the problem space
- Helps you think – focus on semantics
- Provides a glossary of terms – noun based
- It is a static view - meaning it allows us convey time invariant business rules
- Foundation for use case/workflow modelling
- Based on the defined structure, we can describe the state of the problem domain at any time

Simple domain model



Domain classes?

- Each domain class denotes a type of object. It is a descriptor for a set of things that share common features. Classes can be:-
 - *Business objects* - represent things that are manipulated in the business e.g. *Order*.
 - *Real world objects* – things that the business keeps track of e.g. *Contact, Site*.
 - *Events that transpire* - e.g. *sale and payment*.
- A domain class has attributes and associations with other classes (discussed below). It is important that a domain class is given a good description

Identifying domain classes?

- An obvious way to identify domain classes is to identify nouns and phrases in textual descriptions of a domain.
- Consider a use case description as follows:-
 - 1. **Customer** arrives at a **checkout** with **goods** and/or **services** to purchase.
 - 2. **Cashier** starts a new **sale**.
 - 3. **Cashier** enters **item identifier**.
 - 4. System records the **sale line item** and presents the **item description, price** and running **total**.

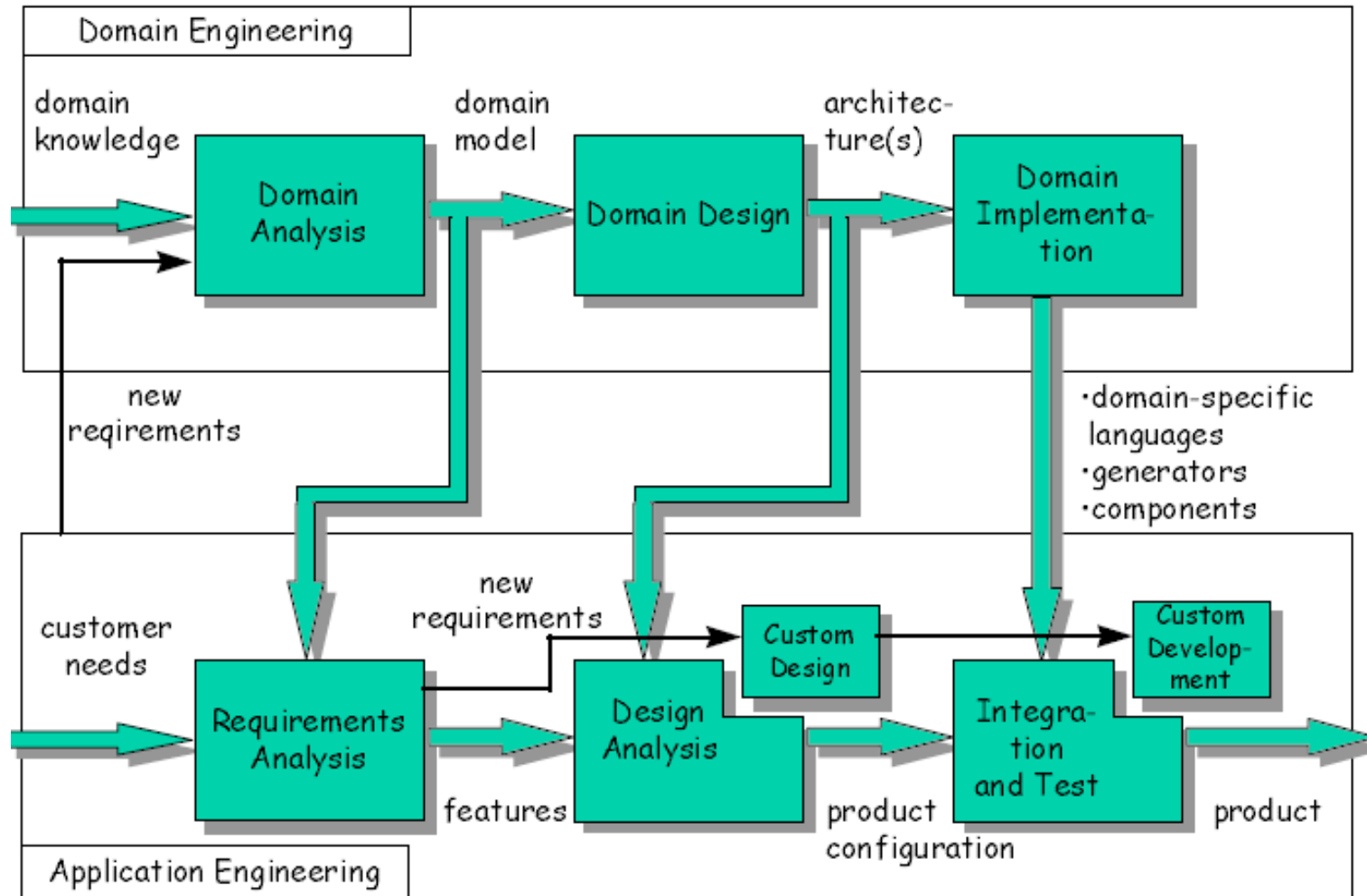
Identifying attributes ?

- A domain class sounds like an attribute if: -
- It relies on an associated class for it's identity – e.g. 'order number' class associated to an 'order' class. The 'order number' sounds suspiciously like an attribute of 'order'.
- It is a simple data type – e.g. 'order number' is a simple integer. Now it really sounds like an attribute!

What is Domain Engineering?

Domain Engineering (DE) is the activity of collecting, organizing, and storing past experience in building systems or parts of systems in a particular domain in the form of reusable assets, as well as providing an adequate means for reusing these assets when building (i.e., retrieval, qualification, dissemination, adaptation, assembly, and so on) new systems.

Software Development based on DE

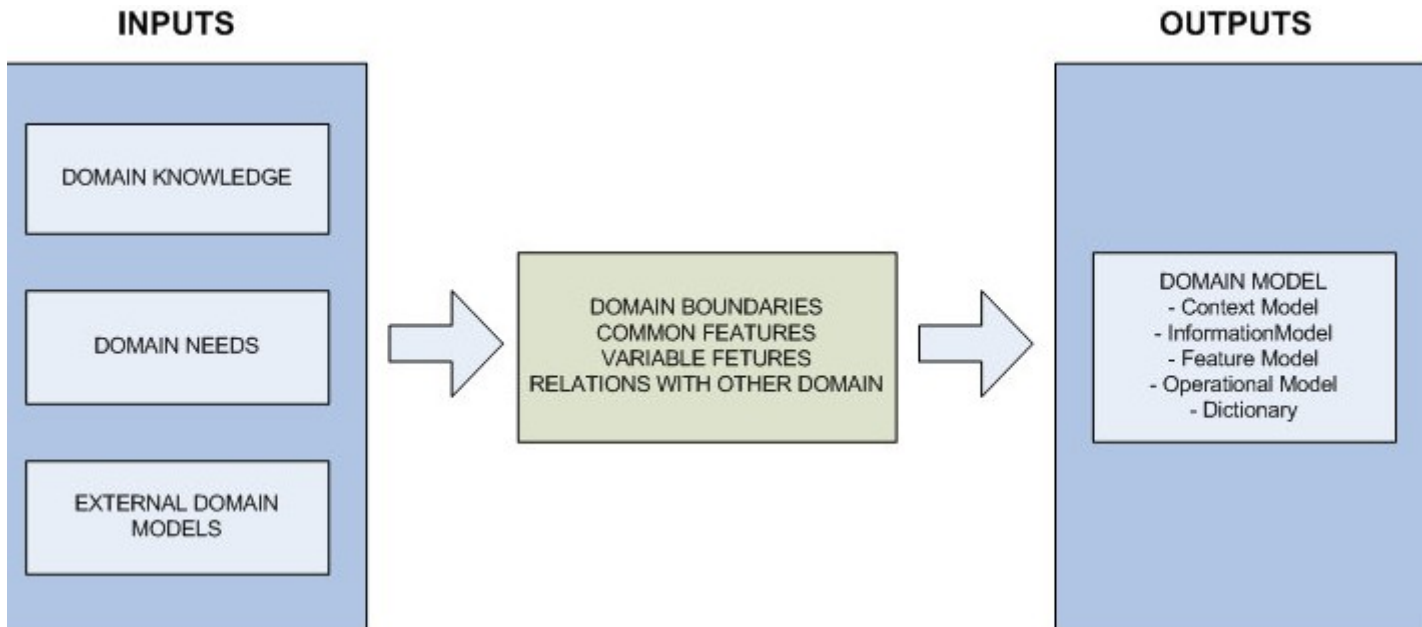


Domain Engineering Phases

Domain Engineering Phases	Main Purpose
Domain Analysis	Domain scoping and defining a set of reusable, configurable requirements for the systems in the domain.
Domain Design	Developing a common architecture for the system in the domain and devising a production plan.
Domain Implementation	Implementing the reusable assets, for example, reusable components, domain-specific languages, generators, a reuse infrastructure, and a production plan.

Domain Analysis

Domain Analysis phase defines the scope of the domain providing a set of models which represent formally all the knowledge obtained about the domain.



Domain Analysis can be divided into three different parts to get the final domain model:

- Requirement Analysis

 - Inputs : List of domain requirements & need**

 - Outputs: Requirements model**

 - Notation: SysML/UML**

 - Tools:Topcased**

- Context analysis

 - Inputs :standards, application framework**

 - Outputs: A context or data-flow diagram.**

 - Notation: SysML-Block Diagram/UML-Class Diagram**

 - Tools:Topcased**

- Domain modeling

 - Information model

 - Feature model

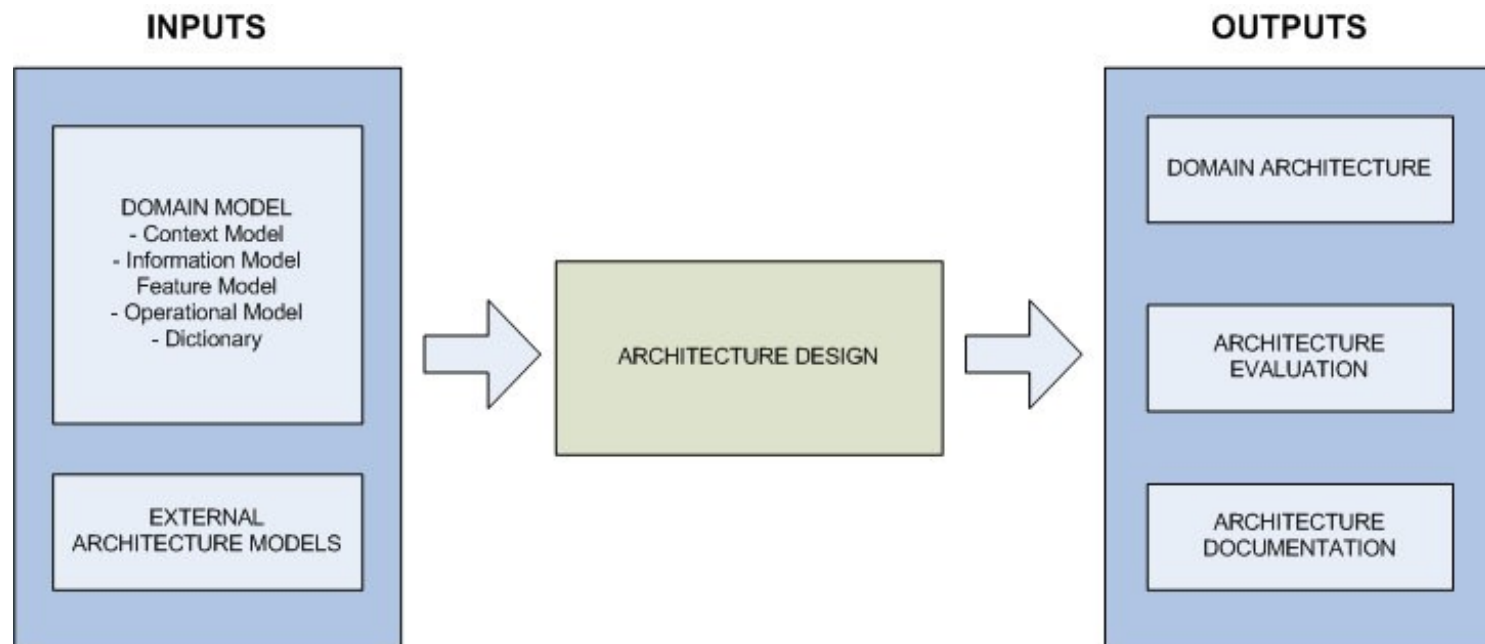
 - Operational model

 - Dictionary

Domain Design

Domain Design is used to specify the application architecture within a domain.

• The domain architecture is a generic based on the domain model. This architecture has to be documented and evaluated.



Inputs : Domain model (Context model, information model, feature model, operational model and dictionary) and other external architecture models.

Outputs :

Functional view: Class structure diagram, which represents the hierarchy of classes in a static view, state chart diagrams shows the internal behavior.

Non-Functional view: Timing, interaction overview and activity diagrams.

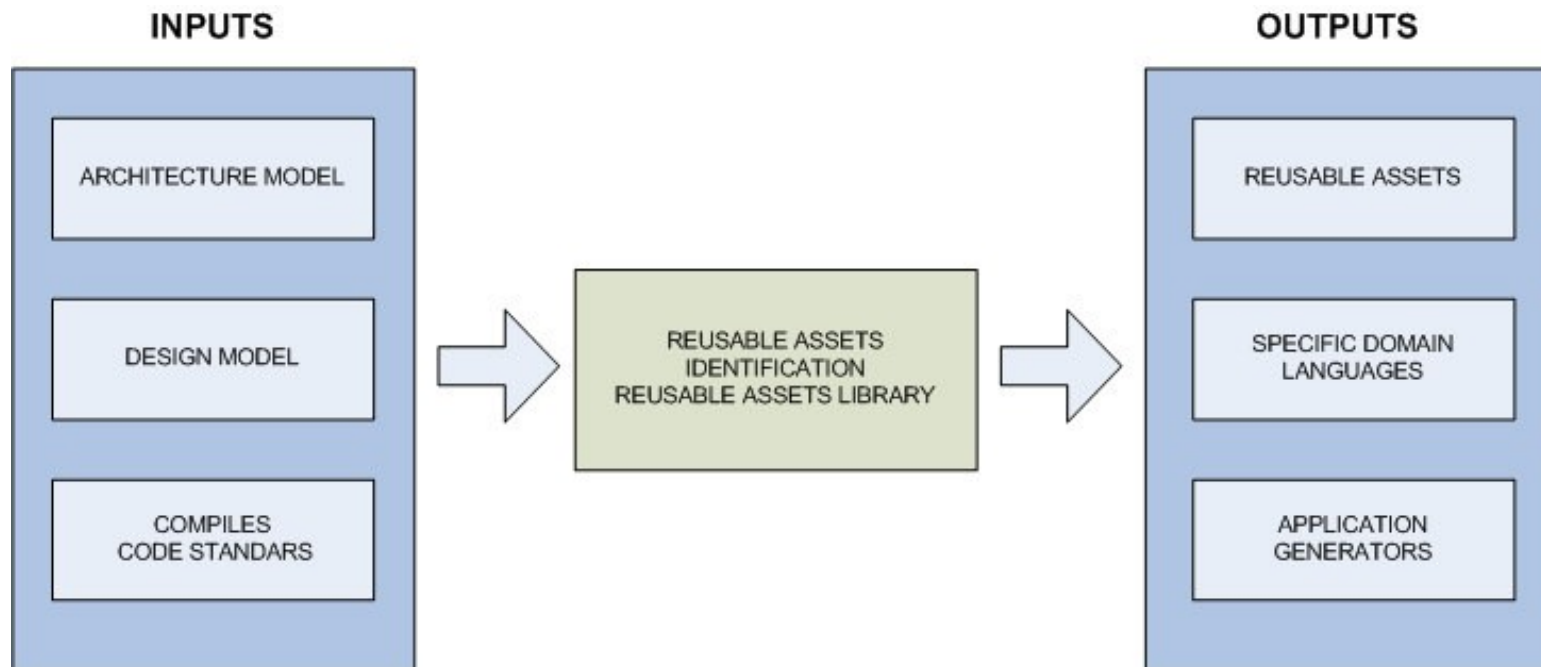
optionally sequence may be used

Notations: UML2

Tools: Topcased

Domain Implementation

Domain Implementation starts with the identification of reusable assets. This identification is made based on the domain model and general architecture obtained in the previous phases. Using compilers and code standards the assets are implemented.



- Code generation approach should be as generic as possible, so that each component could be transformed into a specific language and platform. This transformation process can be done manually, automatically or semi-automatically.
- The tool selected to perform this task is together, which is a product line from Borland. It provides UML and an integrated development environment.

Features of a domain model

- The following features enable us to express time invariant static business rules for a domain:-
- **Domain classes** – each domain class denotes a type of object.
- **Attributes** – an attribute is the description of a named slot of a specified type in a domain class; each instance of the class separately holds a value.
- **Associations** – an association is a relationship between two (or more) domain classes that describes links between their object instances. Associations can have roles, describing the multiplicity and participation of a class in the relationship.
- **Additional rules** – complex rules that cannot be shown with symbology can be shown with attached notes.

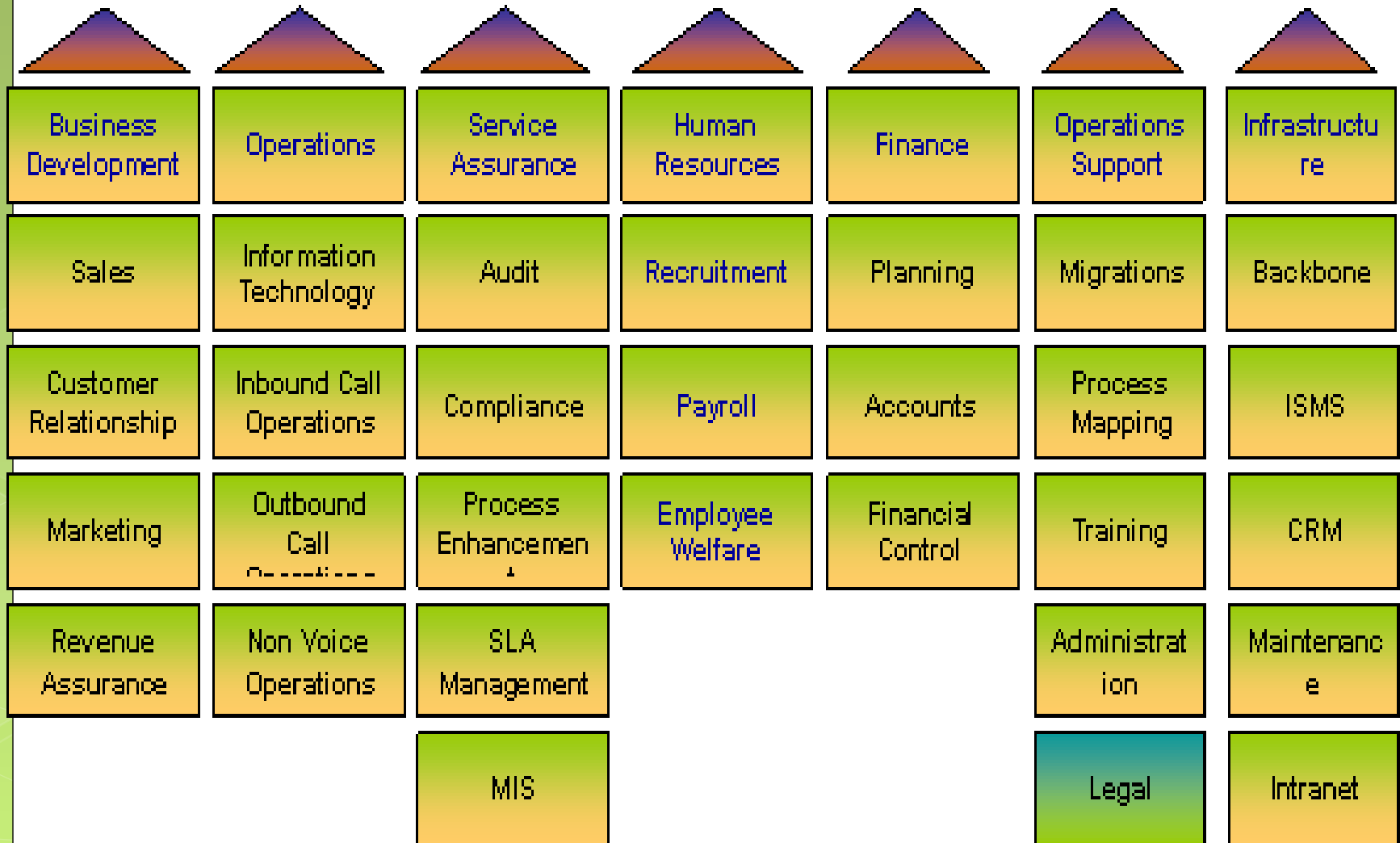
Definition of Business

A business (also known as enterprise or firm) is an organization engaged in the trade of goods, services, or both to consumers. Businesses are predominant in capitalist economies, where most of them are privately owned and administered to earn profit to increase the wealth of their owners. Businesses may also be not-for-profit or state-owned.

Management:

- The efficient and effective operation of a business, and study of this subject, is called management. The major branches of management are financial management, marketing management, human resource management, strategic management, production management, operations management, service management and information technology management.

CEO/COO



Human Resources



Recruitment

Sourcing

New Employee Orientation



Payroll

Attendance Management

Leave Records

Statutory & Regulatory

Compensation



Employee Welfare

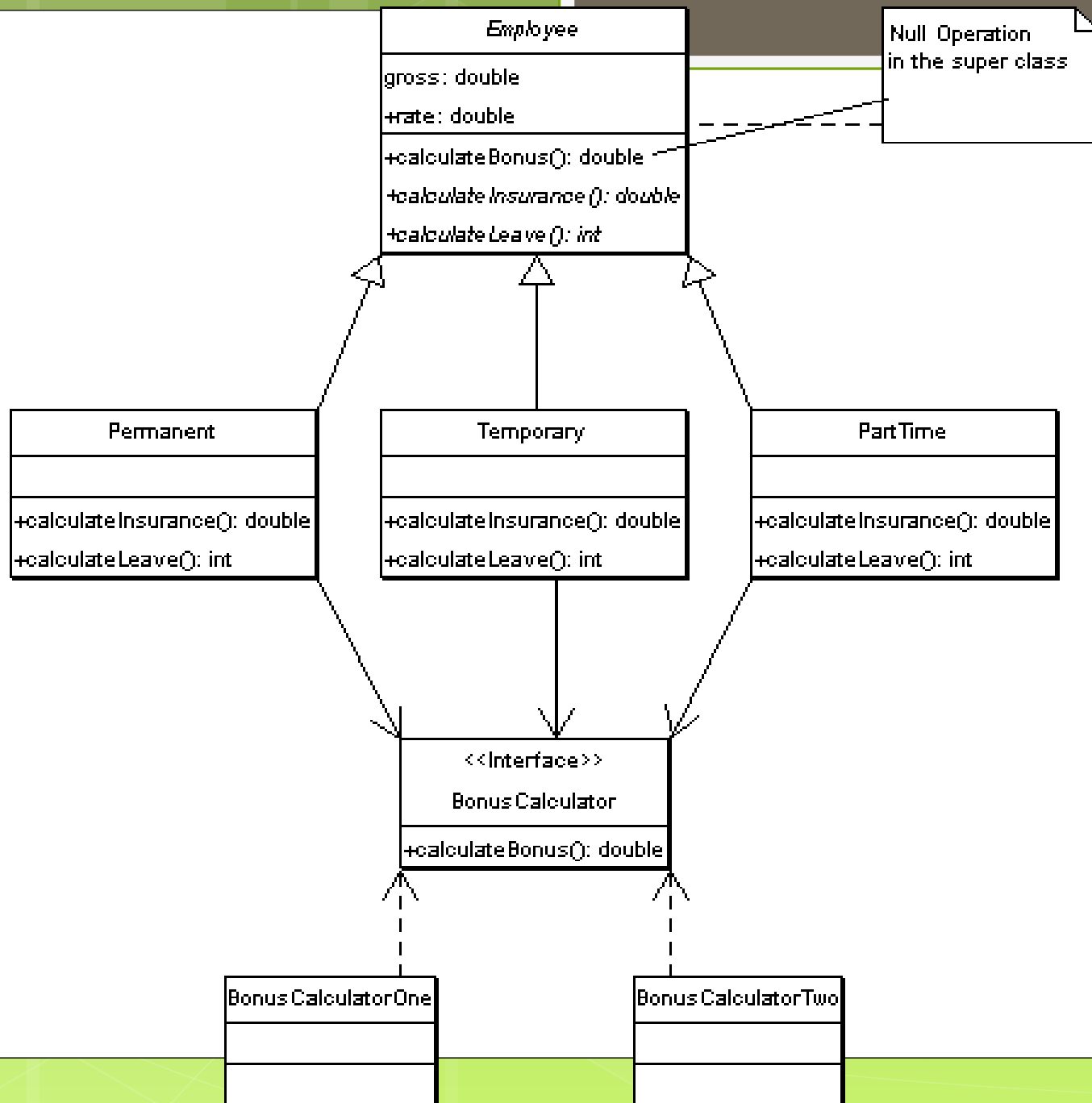
HR Policies

Code of Conduct

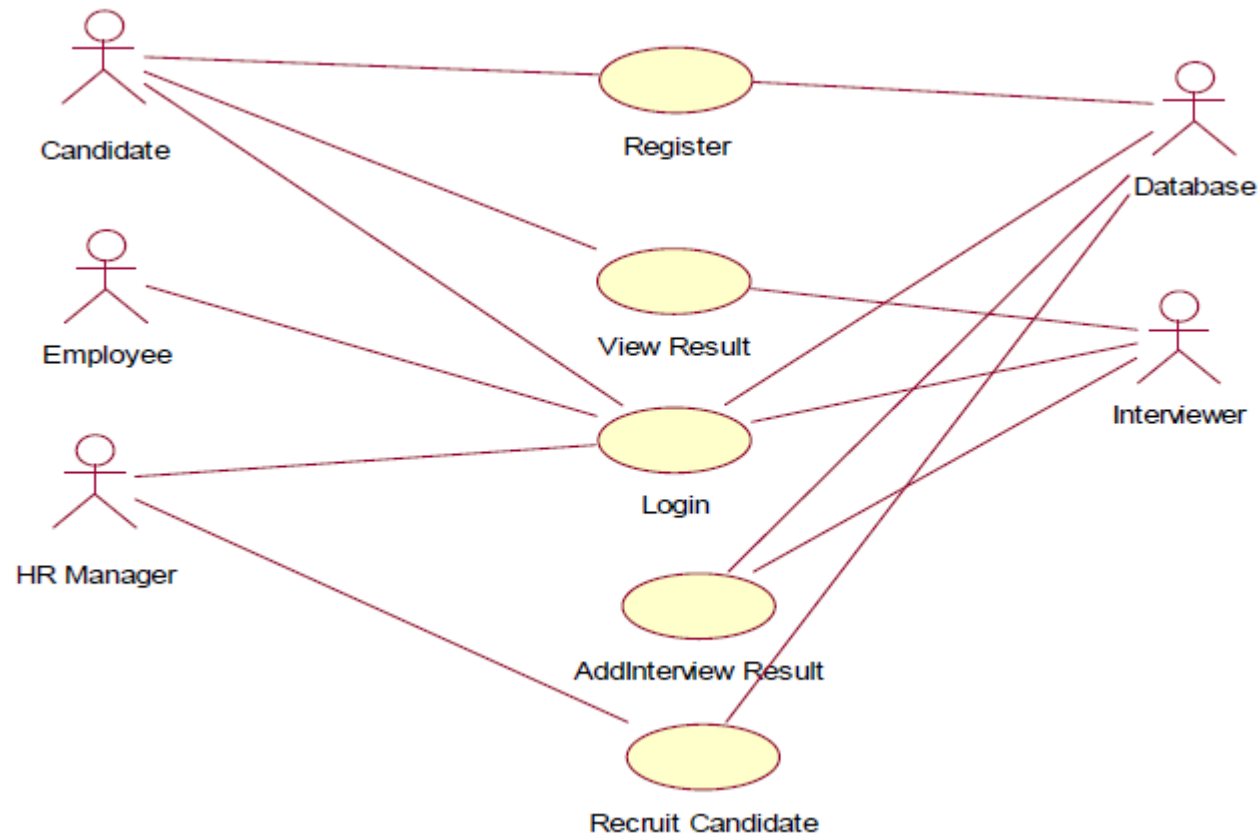
Employee Growth

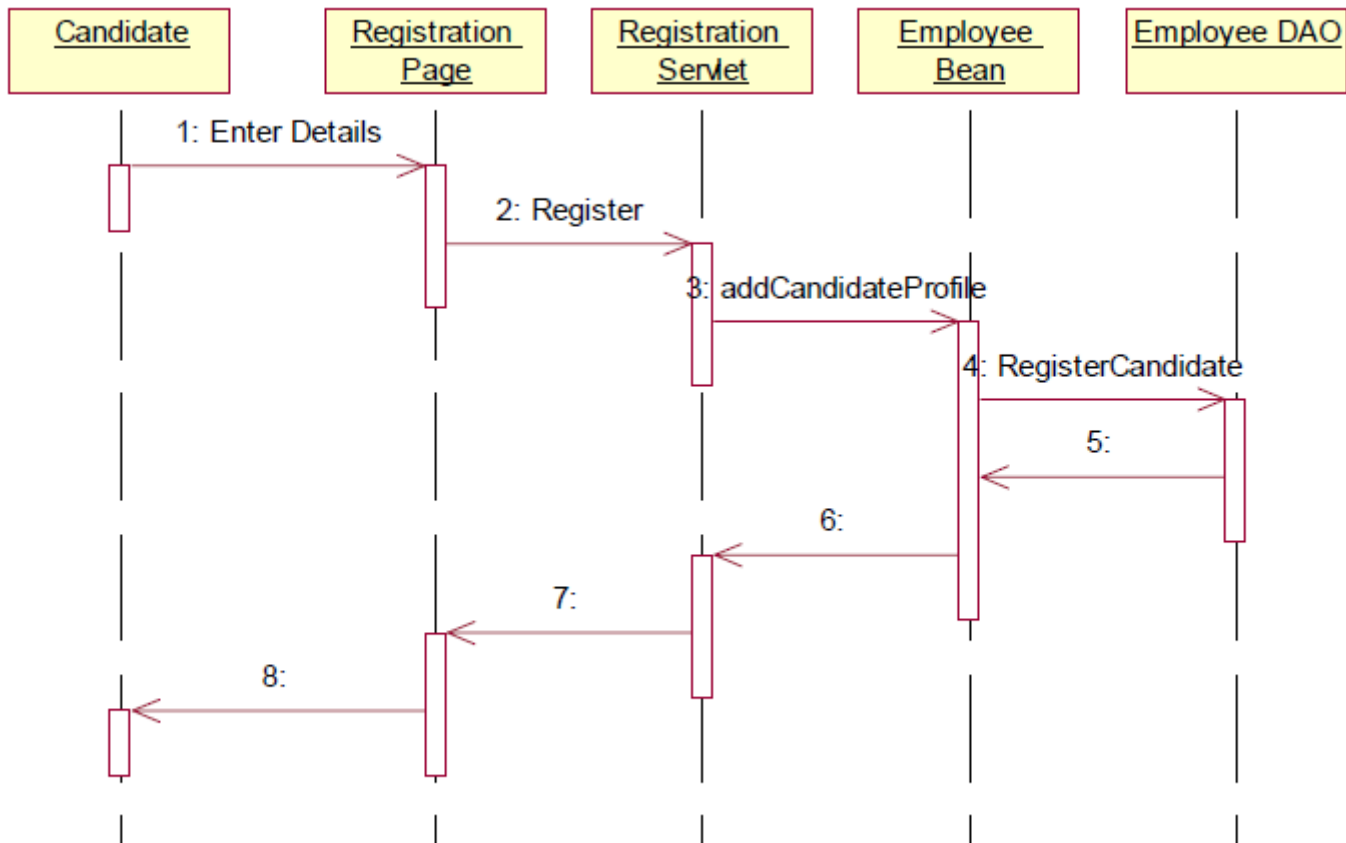
ESAT

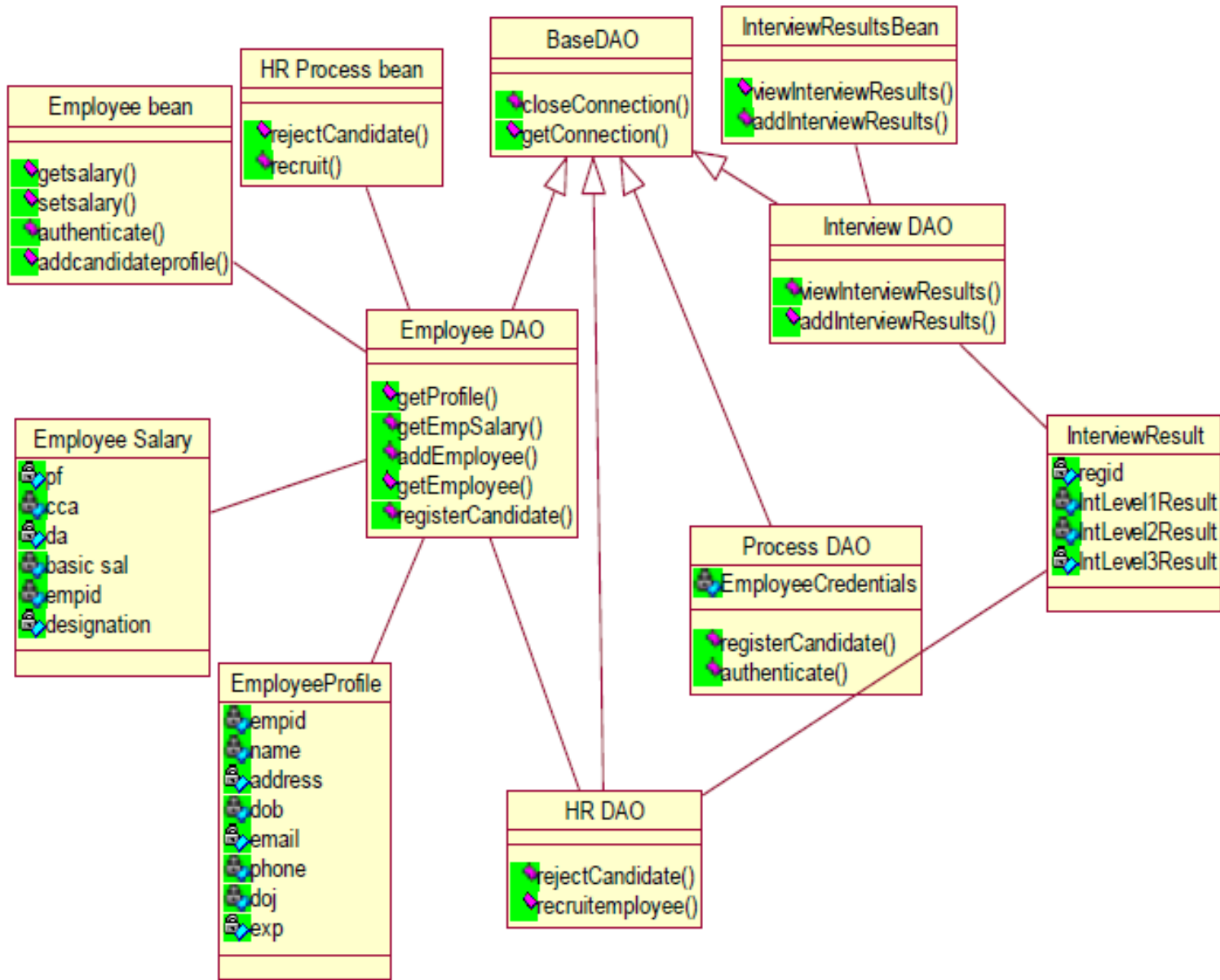
Retention



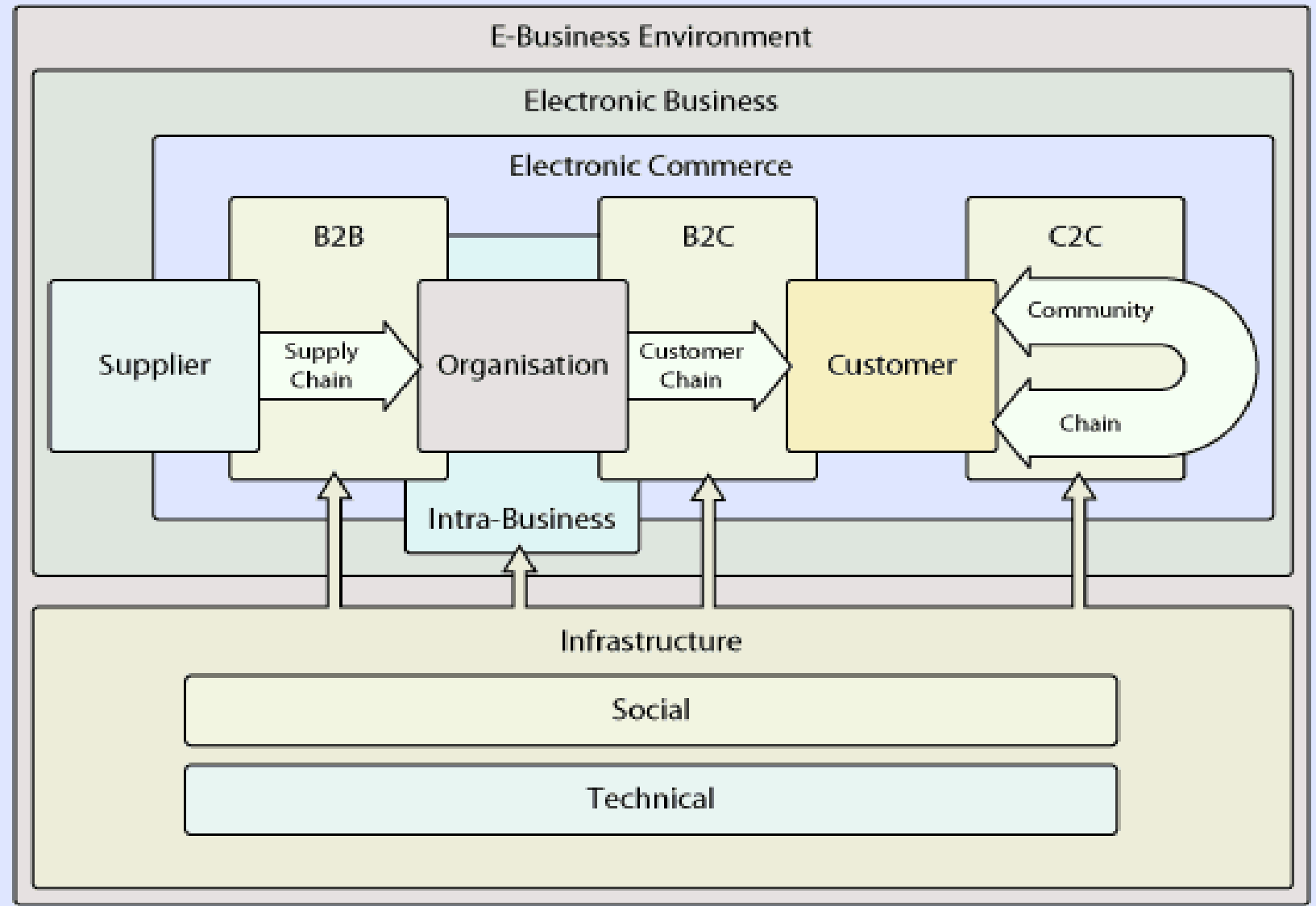
HR Portal Domain Components Extraction











CONCLUSION

Component systems for varied domains help in effective reuse of the product or system help in developing systems within estimated time and budget.

To know the advantages of the component extraction, Net beans profiler is used to study and analyses the performance of component based systems.

Study on component interactions for other domains such as e-governance, e-Banking etc., can be done by making a repository of specific components which might be reused efficiently.

One of the research issues is the mechanism to secure the identified components at the middleware level.

Design and development of new component extraction tools is needed to enhance the power of reusability. Another research issue is to reduce the effort of product development.

END