Chapter 3

Characterization of EM, PD and SD

An insight into how EM, PD and SD differ in character is achieved by comparing them. However, it is important to find an appropriate framework for comparison because the three activities are very different in nature. Consideration of artefacts forms a suitable basis for such a framework. The various aspects of EM, PD and software development are then compared with respect to their use of artefacts.

This chapter compares EM, PD and SD. The comparison focuses on the artefacts: the LSD specification, visualization and animation in EM, the sketch in PD, and the structure, behaviour and process models in SD. In addition to artefacts, the actions, subjects, constraints, environments and knowledge of the activities are compared: knowledge informs actions on an artefact within an environment to represent a subject under certain constraints.

3.1 Background

In the previous chapter EM, PD and SD were described as processes. Many activities are described in this way as processes comprising phases performed in sequence, with each phase characterized by the construction of an artefact. However, such a description is essentially a reconstruction of an activity and not a true characterization of how the activity happens [Goo90, Kap64]. A retrospective view of EM, PD and SD tends to give a misleading account of a planned sequence of actions, when, in fact, the actions were determined by the situation at the time [Sim81].

The lift project case-study provided a particular context in which to consider EM (LSD specification, visualization and animation), PD (sketch) and SD (structure model, behaviour model and process model). It was found that the character of EM, PD and SD was determined by the nature of the artefacts involved. In effect, the nature of the activities was shaped by the particular kind of artefacts used by modellers, designers and software developers in representing the subject. The nature of the artefacts was discovered to influence many aspects of the activity:

- the subject of the activity;
- the actions of the activity;
- the constraints that limit the activity;
- the environment in which the activity happens;
- the knowledge to perform the activity.

Each of these aspects of EM, PD and SD is discussed in the remainder of this chapter.

3.2 Artefacts

It was discovered that particular views of the subject were clarified during the construction of the EM, PD and SD artefacts in the lift project. The modellers found that the view of the system as a collection of observables and agents was made clearer during the construction of the LSD specification. Construction of the other EM, PD and SD artefacts was found to clarify the view of the lift system as a structure and the view of the lift system as a structure with a purpose. The artefacts were found to represent the structure and function of the subject in fundamentally different ways. See Appendix C for all the artefacts constructed during the lift project, including associated DoNaLD and ADM scripts and statements of requirements.

The LSD specification, visualization and animation were the artefacts constructed by modellers during EM in the lift project. The LSD specification, shown in Example 3.1, was introduced in the previous chapter. The term *visualization* is used in this thesis to mean the artefact resulting from **tkeden** interpreting a DoNaLD script. The visualization is essentially a line drawing that the modeller interacts with by changing the values of DoNaLD variables, as shown in Example 3.2. The visualization has no automatic behaviour - such behaviour is characteristic of the animation. The term *animation* is used in this thesis to mean the artefact resulting from **tkeden** interpreting a DoNaLD and ADM script. The ADM entities animate the visualization by changing the values of DoNaLD variables, as shown in Example 3.3.

The sketch was the artefact constructed by designers during PD in the lift project, as shown in Example 3.4. It seems to be widely accepted that the *sketch* is the principal artefact in the conceptual design phase of PD described in the previous chapter [Pug91, Pug96, Fer92]. This thesis concentrates on how the sketch was used by designers in the lift project as a means of turning their ideas for products into product designs with a view to subsequent detail design and manufacture [Pug91, Pug96].

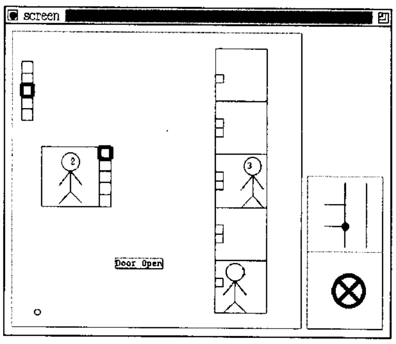
Software developers constructed structure, behaviour and process models during SD in the lift project, as shown in Example 3.5. These artefacts seem typical of those produced during object-oriented analysis using the mainstream methods outlined in the previous chapter. The *structure model* represents the organization of the system into Object classes, the *behaviour model* represents the Object class lifecycles, and the *process model* represents the dataflows between Object classes. This thesis concentrates on how the models were used by software developers in the lift project as a means of capturing the structural and functional aspects of the subject with a view to subsequently designing and coding software.

Example 3.1. LSD specifications in EM. The LSD specification of the Hydrolift pump agent

```
agent pump() {
state
  change target
oracle
  brake pressure chan1
handle
  brake pressure chan2
derivate
  k = 100,
  change is (pressure < target) ? k :
            (pressure > target) ? -k : 0
protocol
  target == pressure + change && brake == OFF
                                -> brake = ON,
  change == 0 -> target = chan1*k,
  pressure == target -> chan2 = target/k,
  brake == OFF -> pressure = pressure + change,
  brake == ON && change != 0 -> brake = OFF
}
```

shows how agents in an LSD specification are defined in terms of observables (states, oracles and handles), derivates and protocols.

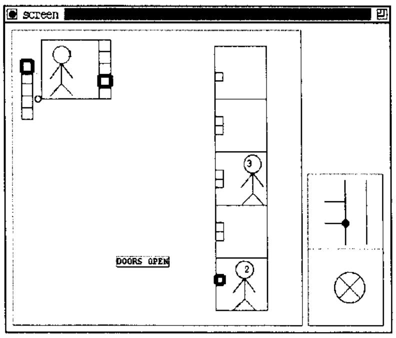
Example 3.2. Visualizations in EM. The screen-shot shows the state of the visualization of the Hydrolift after redefinitions (a) to (h). The sequence of redefinitions changes the state of the visualization to mimic a user on floor 2 travelling to floor 5.

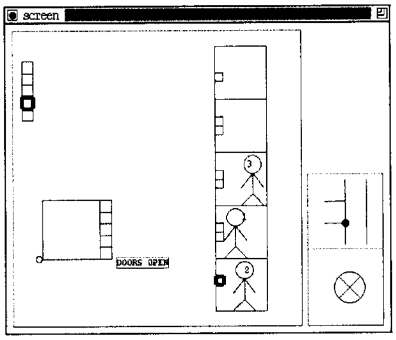



```
liftfloor = 2                  # a car at floor 2
dooropen = true                # b door open
inliftB = true                 # c userB enters car
floorB = liftfloor             # d link user-car
car/button5/light = true       # e select floor 5
dooropen = false               # f door closed
pumpshape/on = true            # g pump on
liftfloor = 3                  # h floor 3
liftfloor = 4                  # i floor 4
liftfloor = 5                  # j floor 5
car/button5/light = false      # k car arrived
pumpshape/on = false           # l pump off
dooropen = true                # m door open
inliftB = false                # n userN leaves car
floorB = 5                     # o unlink user-car
```

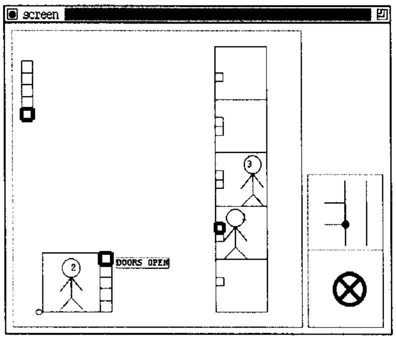
Notice that in (d) variables are synchronized in change and in (e) and (k) the hierarchical structure of the car and button shapes is used.

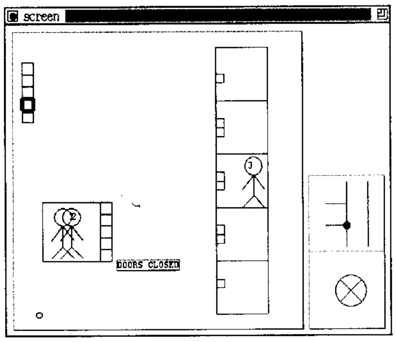
Example 3.3. Animations in EM. The screen-shots show the visualization shown in Example 3.2 with the addition of ADM entities for each user and the lift mechanism providing an automatic behaviour.





(a) User 1 enters and requests floor 2. (b) Arrives at floor 2 and user 1 exits.



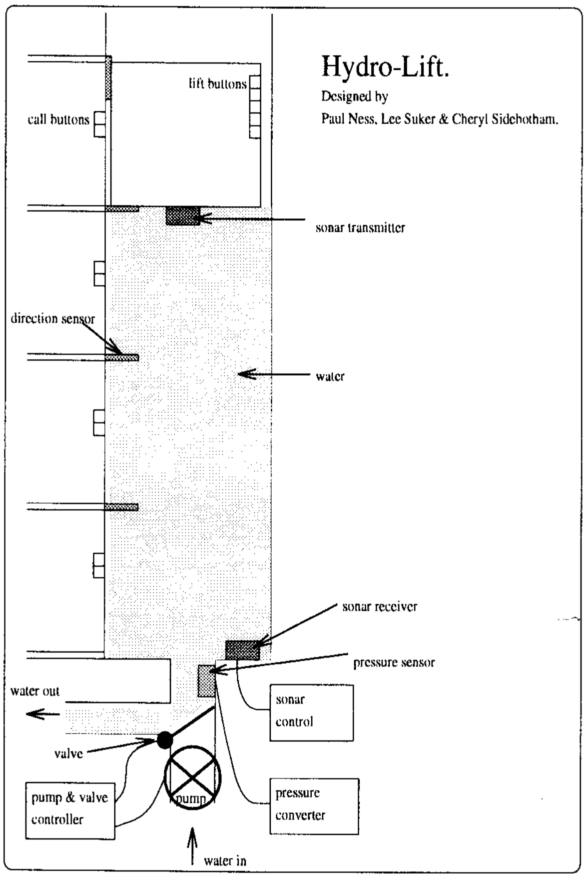


(c) User 2 enters and requests floor 5 . (d) Collects user 1.

The entities generated the redefinitions instead of the modeller.

The LSD specification was found to represent the subject in an intuitive way without any detailed representation of structure or function. The LSD specification was introduced in Chapter 2 as a statement consisting of agent definitions in which are specified the observables, derivates and protocols associated with the agent. It was argued in the previous chapter that these elements and their arrangement in the LSD specification reflect the observables and agents perceived by the modeller within the subject rather than the structure and function of the subject. The arrangement of elements in the LSD specifications of the lift project was found to correspond more to perceived agents and observables than to the structure of the subject or statements detailing the structure and function of the subject.

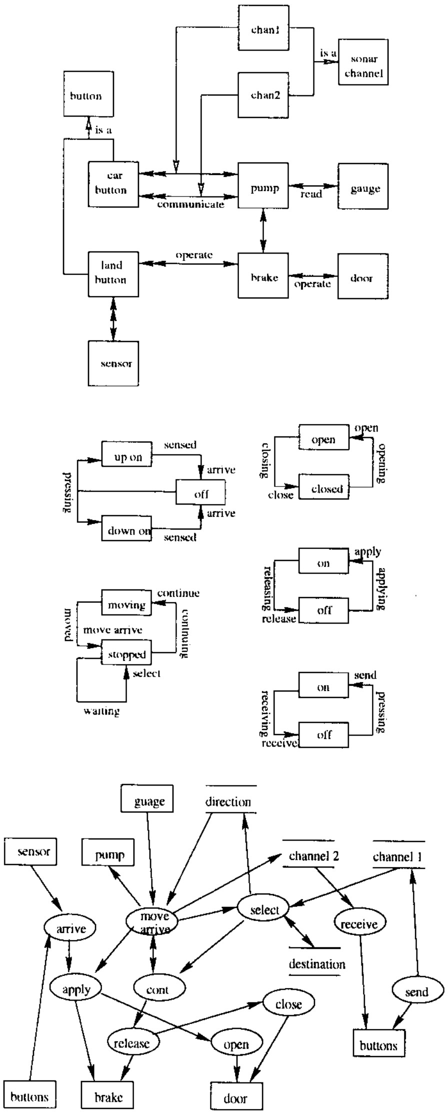
Example 3.4. Sketches in PD. The Hydrolift was first sketched out by designers in the lift project on paper and then using a general purpose line drawing application called xfig. The original computer-sketch



shows a realistic representation of the Hydrolift, albeit in caricature. The labels were not used in later sketches. Designers had to know the conventions for representing components, such as the crossed-circle representing the Hydrolift pump.

The visualization, sketch and structure model represented the structure of the subject in the lift project. Each element in the visualization and sketch was a caricature of the corresponding element in the subject. The modellers and designers used conventions for representing observables and components rather like a "graphical language" in the sense of Ferguson [Fer92] (Section 4.4.2). The structure model in SD serves as a graphical language to represent the structure of the subject, but this representation is more abstract than that in the EM visualization and PD sketch, where the actual arrangement of elements is reflected pictorially.

Example 3.5. Structure, behaviour and process models in SD. The structure, behaviour and process models and associated action definitions, shown here as pseudo-code



`open`	`open door`
`close`	`close door`
`apply`	`apply brake` `generate opening`
`release`	`generate closing` `release brake`
`receive`	`reset button`
`send`	`queue destination`
`select`	`change destination and direction` `generate waiting or continuing`
`arrive (landing)`	`reset button` `generate applying`
`arrive (pump)`	`generate receiving` `generate applying`
`continue`	`generate releasing` `generate moved`

`move`	`generate arriving`
	`move car`
	`generate waiting or continuing`

were the principal artefacts used by software developers in the lift project. The statement of requirements, another artefact used by software developers, is discussed later in this chapter.

The visualization and behaviour models in the lift project were found to represent the states of the subject. The visualization represents the states of the subject as values of DoNaLD variables corresponding to observables in the LSD specification. The visualization was found to improve on the single-state sketch by allowing the modeller to explore the states of the subject by changing the values of DoNaLD variables. The behaviour model represents the states of the subject as symbols in the behaviour model. It was found that the correspondence between the state symbols and the states in the subject was less significant in SD than in EM. The role of state symbols in the behaviour model was found to be as place-holders for abstract state transitions, whereas the visualization and animation represent the states of the subject directly using a visual metaphor.

It was discovered in the lift project that the animation and process model represented the function of the subject. The animation represents the function of the subject as ADM entities corresponding to instances of agents in the LSD specification. The entities redefine DoNaLD variables thus animating the visualization. The behaviour of ADM entities is prescribed by the ADM script and the implementation of the **tkeden** interpreter. In principle, the behaviour of the ADM entities could be defined in a behaviour and process model. The actions in the ADM script are conditional state changes corresponding to the dataflows of the process model that map onto the state transitions in the behaviour model.

3.3 Subjects

A link between the suitability of artefacts for representing the subject and the nature of the subjects was discovered in the lift project. The term *subject* is used in this thesis to mean the system being modeled, designed or analyzed. Whether the subject was novel or familiar to the modeller, designer or software developer was found to significantly influence their activities. For a given novel or familiar subject, some artefacts were found easier to construct, and thus considered more suitable for representing the subject, than others.

The subject of the lift project was lift systems. The systems modelled, designed and analyzed were the SUL, MUL and Hydrolift:

- the SUL is a conventional lift system, from the viewpoint of an individual using the lift (the artefacts based on this personal viewpoint are termed in SUL artefacts);
- the MUL is a conventional lift system, from the combined viewpoints of many

individuals using a lift (the artefacts based on this multiple viewpoint are termed MUL artefacts);

- the Hydrolift is an innovative concept for a lift system, from the imagined viewpoint of an engineering designer (the artefacts based on this engineering viewpoint are termed Hydrolift artefacts).

Essentially, the SUL and MUL were familiar whereas the Hydrolift was, at least to begin with, novel to the modellers, designers and software developers in the lift project. For a detailed description of the lift project subjects see the SUL, MUL and Hydrolift statements of requirements in Appendix C.

It was discovered in the lift project that the suitability of artefacts for representing the subject depended on whether the subject was novel or familiar. The artefacts of EM, PD and SD can be ordered based on this dependency, starting with the artefact found most suitable for representing novel subjects and ending with the artefact found most suitable for representing familiar subjects:

1. LSD specification
2. visualization and sketch
3. animation
4. structure model, behaviour and process model

The LSD specification was found to be most suitable for representing novel subjects, such as the Hydrolift at the start of modelling. The SD artefacts were found to be most suitable for representing familiar subjects, such as the SUL, MUL.

This link between the suitability of artefacts for representing the subject and the novelty or familiarity of the subject provides insight into the use of artefacts in the mental process of conceptualization. It was argued in the previous chapter that EM reflects the process of conceptualization [Bey97]:

1. interaction with artefacts: identification of persistent features and contexts;
2. practical knowledge: correlation between artefacts, acquisition of skills;

3. identification of dependencies and postulation of independent agency;

4. identification of generic patterns of interaction and stimulus-response mechanisms;

5. non-verbal communication through interaction with similar environment;

6. situated use of language;

7. identification of common experience and objective knowledge;

8. symbolic representation and formal languages: public conventions for interpretation.

By mapping the stages of conceptualization onto the earlier sequence of artefacts we see that the PD and SD artefacts address different stages of conceptualization: the sketch is most suited to representation mid-way through conceptualization and the SD models are most suited to representation towards the end of conceptualization. Perhaps most significant is that the LSD specification and visualization appear better suited than the other artefacts to representing the subject during the earliest stages of conceptualization.

3.4 Actions

It was discovered in the lift project that each kind of artefact had a standard repertoire of actions associated with it. The activities of EM, PD and SD can be thought of as sequences of situated actions performed by modellers, designers and software developers as they strive to fulfill their goal of representing the subject. Their actions are the means of attaining their goal given the constraints posed by their environment [Sim81]. An association between artefacts and actions emerged by observing different modellers, designers and software developers constructing artefacts in different environments in the lift project.

The actions of constructing the LSD specification in the lift project were found to reinforce the modellers' beliefs about the subject. The modellers added agent definitions and refined existing oracle, derivate and protocol definitions in the LSD specification, as shown in Example 3.6. The nature of the refinements were found to be, not so much corrections to what was represented, as shifts in the level of detail at which particular elements of the subject were represented, reflecting a better understanding of the subject by the modeller. Conviction about the validity of the model seemed to stem from the direct correspondence between the subject as perceived by the modeller and the representation of the subject in terms of observables and agents in the LSD specification.

The visualization, animation and sketch were typically constructed in the lift project by modellers and designers performing actions in an exploratory fashion. This was typically a two phase operation with modellers and designers first generating the artefacts and then evaluating the artefact against the subject, as shown in Examples 3.6 and 3.7. The generative phase of EM and PD in the lift project typically involved actions informed by the previous evaluation of the artefact:

- actions to change the number, position and appearance of shapes in the visualization by redefining the values of DoNaLD variables on-the-fly;
- actions to change the number and behaviour of ADM entities in the animation by redefining the values of ADM variables on-the-fly;
- actions to change the number, position and appearance of shapes representing components in the sketch by redrawing parts.

It was discovered that the actions of the modellers and designers were typically directed at refining the details of the artefacts whilst keeping the main structural and functional framework intact. This lent integrity to the artefact throughout the changes by maintaining the correspondence between artefact and subject.

Example 3.6. Generative and exploratory actions in EM. The Hydrolift was constructed by modellers in the lift project based on the MUL artefacts. The MUL oracles and handles of the car and shaft agents

```
floor direction brake destination
```

are associated with the mechanics of a lift system whereas the Hydrolift oracles and handles of the pump, sonar and sensor agents

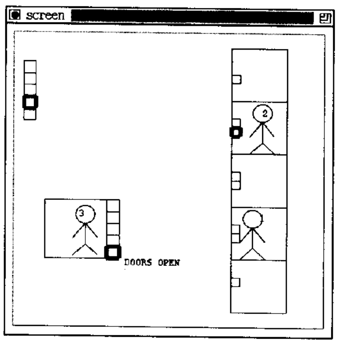
```
pressure chan1 chan2 direction sensed
```

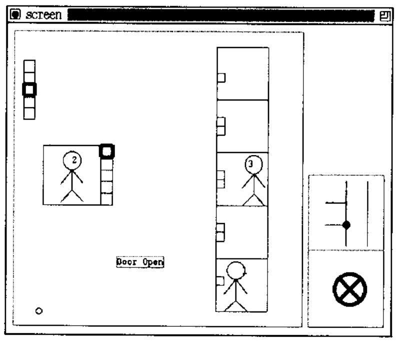
are associated with hydraulics and communication through fluids. The need to link the car, shaft, pump, sonar and sensor in the Hydrolift resulted in the creative exploration of alternative interpretations of the MUL observables:

- the floor could be interpreted as the pressure of the column of liquid at the base of the shaft;
- the direction could be interpreted as the signal from a direction sensor;
- the destination could be interpreted as a target pressure.

From this process of generating and exploring interpretations emerged the LSD specification for the Hydrolift.

The Hydrolift visualization (shown right) was constructed in the lift project by modellers adding DoNaLD definitions, representing shapes corresponding to a pump and valve, to the MUL visualization (shown left).





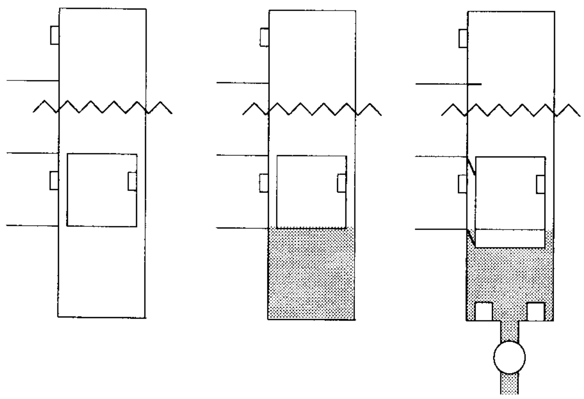
The Hydrolift animation was constructed by modellers adding ADM entities, corresponding to the pump and valve, to the Hydrolift visualization.

Having generated the new artefact the modellers and designers would evaluate it as a representation of the subject. Both modellers and designers explored the correspondence between artefact and subject through imaginary interactions with the subject. In parallel with this the designer performed thought-experiments [Kap64] based on the sketch. The modeller, on the other hand, was able to perform physical experiments using the visualization and animation:

- actions performed in parallel on the visualization/animation and subject to test correspondence;
- actions to setup similar scenarios in the visualization/animation and the subject with a view to testing the correspondence between artefact and subject for exceptional behaviours;
- actions to search for inconsistencies between the visualization/animation and the subject.

Such exploration typically suggested further changes to the artefacts to improve the representation of the subject. Inadequacies in the artefacts were found to diminish as modelling and design progressed with more time being spent by modellers and designers on refining details of the representation than on exploration.

Example 3.7. Generative and exploratory actions in PD. The following three sketches



show the three steps in designing the Hydrolift:

1. the designer retrieved their sketch for the MUL;
2. the designer added a representation of water filling the shaft;
3. the designer explored the composition and added more detail to the sketch in the form of appropriate components, such as a pump and sonar.

This approach to sketching the Hydrolift was found to keep the basic structure of the MUL.

It was found that software developers in the lift project constructed the structure, behaviour and process models by performing actions that were transformational in nature, as shown in Example 3.8. Exploratory type actions were generally found more likely to lead to ambiguities and inconsistencies in the models, so tended to be avoided, if at all possible, by the software developers in the lift project. The construction of models involved actions that transformed symbols:

1. actions to transform nouns into Object classes;
2. actions to transform noun phrases into structural relations between Object classes;
3. actions to transform verbs into actions;
4. actions to transform verb phrases into functional relations between Object classes;
5. actions to transform structure model Object classes into behaviour model state machines;
6. actions to transform structure model actions into behaviour model action labels and process definitions;
7. actions to transform structure model functional relations into state transitions;
8. actions to transform behaviour model state transitions into process model dataflows;
9. actions to transform behaviour model actions into process model actions;
10. actions to transform structure model Object classes and attributes into process model message sources and sinks and data stores.

It can be seen from the above list of transformations performed by software developers that the relative position of nouns and verbs within noun and verb phrases was important. These were typically written within a statement of requirements for the subject.

Example 3.8. Transformational actions in SD. It was discovered in the lift project that software developers generally constructed structure, behaviour and process models by transforming parts of other models or a statement describing the subject. For example, the requirements for the lift

> ... The shaft mechanism moves the car towards a destination landing stopping whenever the brake is applied. The brake is applied whenever the car arrives at a landing requested by a user ... The shaft mechanism releases the brake and starts the car moving again. For safety the door is opened and closed by the brake ensuring that the door is only open whilst the brake is on.

was used by software developers as the basis for transformations to construct the structure, behaviour and process model of the brake:

- The nouns "brake" and "shaft" were transformed into object Class representations in the structure model.
- The verb phrase "the door is opened and closed by the brake" was transformed into a functional association between the brake and door Object classes in the structure model.
- The verb phrase "The shaft mechanism moves the car towards a destination landing stopping whenever the brake is applied" was transformed into a functional association between the brake and shaft Object classes in the structure model.
- The phrase "The brake is applied whenever the car arrives at a landing requested by a user" was transformed into a functional association between the brake and lift button Object classes in the structure model.
- The verbs "applied" and "released" were transformed into the actions and transitions of the brake Object class represented in the behaviour model.

It was discovered that verb phases, functional associations, state transitions and dataflows had essentially the same meaning in SD. In effect, the software developer was expressing the meaning of model elements in different languages by transforming them.

3.5 Constraints

It was discovered that the timing and choice of actions during the process of EM, PD and EM was largely determined by the constraints imposed by elements within the environment. The constraints are the conditions for goal attainment [Sim81]. Pugh identifies a number of boundaries in total design that constrain the activities of designers [Pug91, Pug96]:

- the *business design boundary* represents the constraints placed upon the supply of technology and techniques to the design process by the elements of the business structure which include management, planning, organization and control;
- the *personal design boundary* (or interpersonal design boundary) represents the constraints placed upon the design process by the personal characteristics and skills of the designers which include their abilities to question existing practices, involve themselves in new disciplines and to communicate;
- the *product design boundary* is represented by the product design specification (PDS) that defines in detail the wide variety of constraints shown in Table 3.1 to be placed upon the design of a product.

Similar technical and non-technical constraints were identified in the lift project. The activities of modellers, designers and software were largely determined by what information and techniques they had available to them as well as those involved and the specifications of the subject.

Computer technology was found to be more of a constraint on the actions of modellers during the lift project than on the actions of designers and software developers. The designers and software developers used a simple line drawing application to construct their artefacts in the lift project. The **tkeden** interpreter, underlying the visualization and animation, is a computer-based tool that has the potential to support far more sophisticated modes of interaction. But visualizations and animations have to be simple given the current computer and **tkeden** interpreter technology. Although alternative technologies are being considered it seems that this limitation will always exist if the desired flexibility of the EM tools is to be maintained.

environment	ergonomics	scientific disciplines
company constraints	development costs	timescale
quality and reliability	maintenance	competition
product life span	life in service	materials
quantity	aesthetics	performance
standards and specification	profitability	research and development
customer	packaging	shipping
size	weight	market constraints
manufacture	product cost	safety

Table 3.1: Elements of the product design specification

The constraints on techniques was found to be most severe upon the actions of the software developer in the lift project. Pugh refers to the techniques available to a designer as their "tool-kit", that includes techniques of analysis, synthesis, decision making, costing and modelling. The tool-kit of the software developer in the lift project was essentially limited to the standard object-oriented method of analysis:

1. transform the statement of requirements into a structure model;
2. transform the structure model into a behaviour model;
3. transform the behaviour model into a process model.

This method orders the transformational actions of the software developer: constructing the structure model uses actions 1 to 4; constructing the behaviour model uses actions 5 to 7; constructing the process model uses actions 9 to 10. The method presumes the existence of an unambiguous and consistent statement of requirements describing the subject. When such a statement exists the method of analysis can be a very powerful tool to the software developer.

The actions of the modellers and software developers were found to be constrained by specifications of the subject in the lift project, as shown in Examples 3.10 and 3.9. Pugh describes the PDS as a having a number of essential qualities:

- the PDS is a comprehensive and unambiguous specification of the product to be designed;

- the PDS constrains the actions of developers both subliminally during the generation of a sketch and consciously during the evaluation of a sketch;
- the PDS is an evolutionary document which, upon completion of the design activity, has itself evolved to match the characteristics of the final product;
- the PDS is written as "short, sharp definitive statements" rather than in "essay" form.

The LSD specification is the same as the PDS in all these respects except that the LSD specification is used by modellers in order to constrain the construction of visualizations and animations. The statement of requirements is similar to the PDS only it represents a more severe constraint on the actions of the software developer: the statement of requirements should be complete (complete is stronger than comprehensive) and not change during the construction of models.

Example 3.9. Statement of requirements as constraint in SD. When used, the statement of requirements was found to constrain the software developer. The pattern of nouns (bold) and verbs (italic) and associated nouns and verb phrases were found to determine the transformations performed by the software developer in the lift project.

> On each **landing** there is an up and a down **button**. In the **car** there is a **button** for each **floor**. **Users** *make* **requests** for the **car** to *come* to their **landing** or *go* to another **landing** by *pressing* these **buttons**. The shaft **mechanism** *moves* the **car** towards a destination **landing** *stopping* whenever the **brake** is *applied.* The **brake** is *applied* whenever the **car** *arrives* at a **landing** *requested* by a **user** (for **requests** from **landings** the **direction** matters). On *arriving* at the destination **landing** the shaft **mechanism** *selects* the next **destination**. The shaft **mechanism** *releases* the **brake** and *starts* the **car** *moving* again. For safety the **door** is *opened* and *closed* by the **brake** ensuring that the **door** is only **open** whilst the **brake** is **on**.

The statement of requirements constrained the software developer to which transformations were performed. The sequence of these actions was constrained by the software development method.

Example 3.10. LSD specification as constraint in EM. It was found in the lift project that the LSD agent definitions, such as the definition of the shaft

```
agent shaft() {
state
  floor destination direction
oracle
  brake
handle
  brake
derivate
  direction is (floor < destination) ?  UP :
               (floor > destination) ?  DOWN : NIL
protocol
  brake == OFF -> floor = floor + direction,
  brake == ON && direction != NIL -> brake = OFF
}
```

seemed to be an absolute and unambiguous representation of the subject with few alternative representations emerging during the project. There seemed to be a direct and obvious correspondence between the agents and observables in the LSD definitions and those perceived within the subject:

- the shaft agent maps onto the notion of the mechanism responsible for raising and lowering the car;
- the observables map onto what the modeller thinks the shaft must be sensitive to or be able to act upon in order to move the car.

This authoritative definition of the subject was found to limit the activities of the modellers constructing the visualization and animation to represent the observables and agents within the LSD specification.

3.6 Environments

The constraints on the actions were found to be determined by the elements within the environment of the modellers, designers and software developers in the lift project. Constraints result from the presence of physical entities:

- the scientific and engineering knowledge recorded in papers, books, reports and such like determine the technological constraints;
- the people and tools available determine the constraints on techniques;
- the people in a group determine the interpersonal constraints;

- the specification document determines the constraint on the development of the system.

It was found in the lift project that EM, PD and SD were each characterized by the objects and people present within their respective environments. In particular, the environments were found to consist of a large number artefacts either archived or in the process of construction.

The emphasis of the environment of the modeller was found to be on the subject of lift systems. The environment of the modeller in the lift project consisted mainly of material linked with the current subject as, for example, the Hydrolift:

- books, papers, articles and project reports on lift systems;
- design sketches of the Hydrolift;
- computer running visualization or animation of the Hydrolift;
- LSD specification of the Hydrolift;
- computer archives of visualizations and animations of the SUL and MUL;
- archives of LSD specifications of the SUL and MUL;
- discussions between modellers about lift systems.

In addition, archives of information about EM was available in case the modeller had problems with one of the notations, tools or methods of representation. However, the environment of the modeller during the lift project consisted mainly of elements linked with the subject of lift systems.

As in EM, the emphasis of the designer's environment was found to be on the subject of lift systems. Ferguson describes a context for design in 1899, as shown in Example 3.11, that is similar in character to the design environment in the lift project:

- material for immediate reference including papers, sheaves and rolls of blueprints, and printed material;
- original full-size drawings stored in draws for immediate retrieval;

- data books and sketch books;
- consultations between designers;
- photographs and actual finished designs.

A modern design environment is typically computer-based. This has a number of advantages that are associated with the computer being able to store and manipulate components representations. This facility was not used in the lift project because of the simplicity of the designs and the novelty of the Hydrolift. An account of the negative impact of computers on innovation in product design, discussed at length by Ferguson and Pugh in [Pug91, Pug96, Fer92, Fer77], is outside the scope of this thesis.

The emphasis of the software developer's environment was found to be on the subject of SD. The environment of the software developer in the lift project consisted predominantly of material linked with SD. Similarities between the environment of the IBM WSDL (Appendix B) and the context for SD in the lift project were observed:

- books describing the object-oriented analysis and design method;
- printouts and screen displays of structure, behaviour and process models;
- discussions about improving the effectiveness of the method;
- computer tools for supporting the method;
- archives of software components for use in the design phase;
- detailed knowledge about the subject in the form of a statement of requirements or software for a system with the similar behaviour to the subject.

In the lift project, essentially all the information needed about the subject was in the statement of requirements. Similar characteristics were observed in the IBM WSDL project working environment.

Example 3.11. Traditional environments in design. Ferguson gives the Baltimore and Ohio Railroad drafting room in 1899 as an example of an archetypal product design environment before the computer workstation took over:

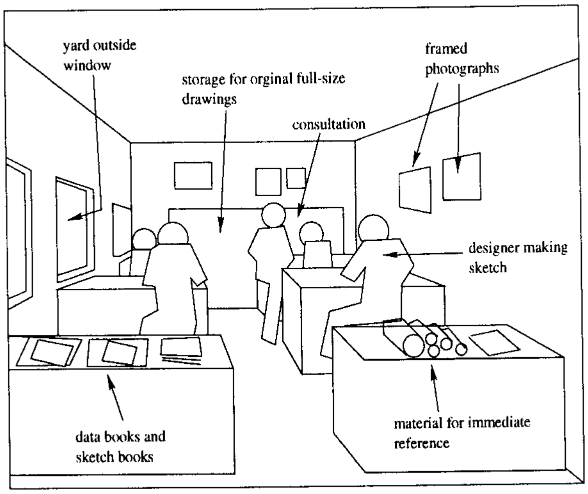


Figure 3.1: Baltimore and Ohio Railroad drafting room 1899.

"The photograph [copied in Figure 3.1] clearly shows where and how locomotives, rolling stock, and railroad structures were designed. The designer in the right foreground appears to be making a sketch that will be converted by a draftsman into a working drawing. He has for immediate reference papers, sheaves and rolls of blueprints, and printed materials. At the far end of the room, original full-size drawings are stored flat in drawers for immediate retrieval. Data books and sketch books are in the foreground. A consultation between two colleagues is taking place at the centre of the picture. The results of the designers' and the draftsmen's work are displayed in framed photographs on the walls, and one can walk into the yards to see the real thing. The designers are thus intimately in touch with the world they have designed, and they are engaged intellectually and physically at a detailed level in planning the future of their railroad" [Fer92].

3.7 Knowledge

It seemed that the modellers, designers and software developers used different kinds of knowledge in the construction of artefacts to represent the subject. The kind of knowledge appeared to reflect the nature of the artefacts:

- the modeller used knowledge about observables and agents to construct the LSD specification;

- the modeller and designer used knowledge about the structure, disposition and relationship between agents to construct sketches and visualizations;
- the modeller and software developer used knowledge about behaviour and function to construct the animation.

The knowledge used by modeller, designers and developers came from long-term memory and the immediate sensations of interacting with the artefacts and subject. It is this knowledge that informed the actions of the modellers, designer and developers in the lift project.

An appropriate way to view knowledge in the lift project is in terms of mental models. It is generally accepted among cognitive scientists that knowledge consists of mental models that mimic the characteristics of external entities [JL83, JL88, GS83]. Johnson-Laird provides a taxonomy of mental models [JL83]:

1. a simple relational model consisting of tokens and relations;
2. a spatial model is a relational model consisting of spatial relations;
3. a temporal model is a sequence of spatial models related temporally;
4. a kinematic model is a temporal model without temporal discontinuities;
5. a dynamic model is a kinematic model in which there are causal relations between certain spatial models contained within it;
6. a conceptual model is a dynamic models in which there are semantic and recursive relations.

The first five are termed physical models "in that, with the possible exception of causality, they correspond directly to the physical world. They can represent perceptible situations" [JL83]. The final mental model encompasses all conceptual models and is characterized by recursive relations and semantic relations that support the use of language.

There is a link between the characteristics of artefacts and mental models. It is possible to associate the artefacts of EM, PD and SD with the mental models described in the Johnson-Laird taxonomy:

- LSD protocols and agents correspond to dynamic and simple conceptual models respectively;
- visualizations correspond to dynamic models;
- animations correspond to kinetic models;
- components in a sketch typically correspond to dynamic models;
- structure, behaviour and process models correspond to conceptual models about structure and function.

It is expected that an agent corresponds to a dynamic model in which there are relations grouping causal relations. The SD artefacts would be expected to correspond to conceptual models that are complex systems of relations representing structure and function.

The notions underlying LSD protocols and agents were found to be familiar to everybody in the lift project even though the notions were difficult to define precisely. Perhaps this familiarity was because the general concepts underlying protocols and agents are reinforced whenever we perceive the consequence of an action or interact with another person. Causality and personification are widely recognized as being perhaps the most familiar of all concepts [HT95]:

- causality is the intuitive understanding that some regularities in the world are based on the relationship between causes and effects;
- personification means to treat something that is not a person as if it were one.

The concepts of causality and personification straddle the boundary between physical and mental models in Johnson-Laird's taxonomy. They are both difficult to visualize and difficult to put into words. However, they are recognized as playing an important role in the mental process of analogical transfer [FWS92, HT95].

The LSD specification has an important role to play in the conceptualization of novel subjects by encouraging the "mental leap" [HT95] from a physical to a conceptual model. Although the author can only surmize about the activity of the brain based on the evidence, it is plausible that the modeller has three kinds of mental models in his head when constructing and using an LSD specification:

- a physical model corresponding to the novel subject;
- a physical/conceptual model corresponding to protocols and agent definitions in an LSD specification;
- conceptual models in long-term memory.

This suggests that the LSD specification acts as a "conductor" for conceptual knowledge to flow to the novel subject so that it can start to be understood in terms of familiar concepts. This mental process is known as analogical transfer in which a relationship or set of relationships in one context is transferred to another, resulting in mental models that are analogous to those already familiar [FWS92]. In effect, the subject is created in the mind of the modeller through analogical transfer. Analogical transfer is an important part of common sense thinking: "[common sense] reflects an enormous amount of information that one has gained about the world and provides a large number of practical rules - many of them quite logical - for dealing with day-to-day life. It is so much part of everyday life that one seldom thinks about it" [Wol92]. Analogical transfer is also widely regarded as one of the most important processes in creativity [FWS92, Pug91, HT95].

3.8 Summary and conclusion

In this chapter, the disciplines of EM, PD and SD were characterized in terms of of the artefacts, subjects, actions, constraints, environments and knowledge of each discipline. It was found that EM and the conceptual design stage of PD were similar in the lift project:

- the artefacts corresponded to the subject;
- the subjects were novel or familiar;
- the actions were generative and exploratory in nature;
- the constraints acted to limit rather than prescribe the activity;
- the environment emphasized the subject;

- the knowledge used was essentially "physical" mental models [JL83].

In the lift project, SD was found to be different from both EM and PD:

- the artefacts corresponded to abstract conceptual models of the subject;
- the subjects were typically familiar;
- the actions were transformational in nature;
- the constraints acted to limit and prescribe the activity;
- the environment emphasized the process of developing software;
- the knowledge used was essentially abstract conceptual mental models [JL83].

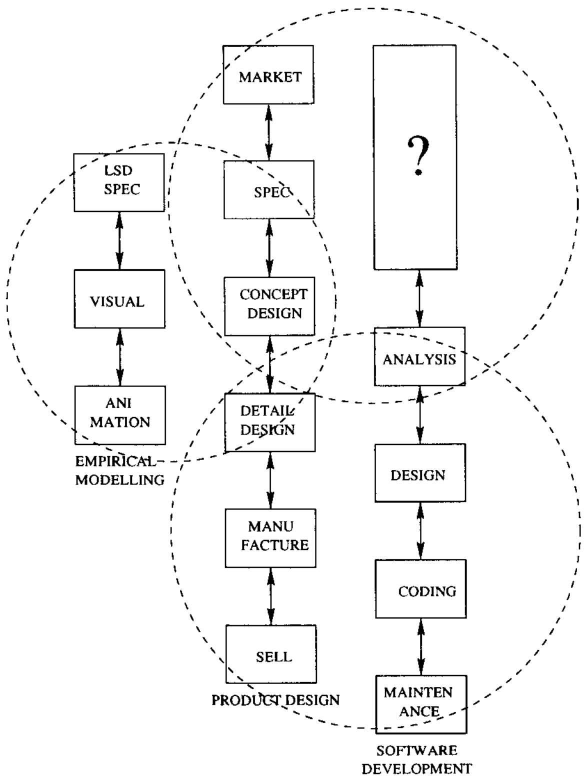


Figure 3.2: Comparison of EM, PD and SD.

These results are represented in Figure 3.2. It shows the similarity between EM and conceptual design identified here and in [ABCY94c, Car94, Bey89]. In addition, the figure shows the similarity between SD and the subsequent phases of PD, from detail design through to selling the product in the marketplace. The construction of the models in SD parallels engineering in detail design, characterized by the use of analysis, prototyping and computer simulations [Pug91, Fer92].

Figure 3.2 shows an activity preceding SD that has the same characteristics as EM and conceptual design in PD. The results of this chapter suggest that artefacts that support creativity will be essential determinates of this question-mark activity. With this emphasis on creativity the new approach to software development may appropriately be termed "creative software development".