

### Stakeholders

#### Characterisation 9.150

By a domain stakeholder we shall understand

- a person, or a group of persons, “united” somehow in their common interest in, or dependency on the domain; or
- an institution, an enterprise, or a group of such, (again) characterised (and, again, loosely) by their common interest in, or dependency on the domain.

#### General Application Stakeholders

- By general application domain stakeholders we understand stakeholders whose primary interest is neither the projects which develop software (from domains, via requirements to software design), nor the products evolving from such projects.
  
  - Instead we mean stakeholders from typically non-IT business areas.

#### Example 9.94 Railway Train System Stakeholders

When modelling, i.e., describing, the domain of railways, one may be well advised in considering the following stakeholder groups listed in an order that may reflect the view of the first group:

- the public government, the service sector, etc.
- transportation, manufacturing, mining, financial industries,
- public government, the service sector, etc.

---

**The aims are**

- to introduce the concept of (domain) stakeholders,
- to distinguish between different categories of stakeholders, and
- to sketch a fairly advanced (also formalised) example of enterprise stakeholders.

**Introduction**

- At the very outset of any phase of development,
  
  - whether the universe of discourse be some domain model development,
  - requirements development or software design,

  - it is important to identify all possibly relevant stakeholders.

  Throughout the development phase it is then important to ensure that each stakeholder (group) is properly “taken care of”, i.e., their concerns are properly modelled.

---

- The objective is
  
  - to ensure that you carefully consider and include the concerns of all relevant stakeholders
  - when in future you are developing
    
    - domain descriptions,
    - requirements prescriptions and software designs.

- The treatment is from systematic to formal (sketches).

---

**Domain Stakeholders**

- The prerequisite for following this part of the lecture is that you have followed lectures over-viewing
  
  - software development in general and
  - domain engineering in particular.
owners (e.g., stockholders or a government),
management (consisting of
  executive management,
  mid-level management,
  operational (“floor”) management (i.e., “white collar workers”),
  etc.),
railway staff at large (“people on the floor” other than “floor”
management, i.e., “blue collar” workers — and possibly arranged
into several stakeholder groups including families),
customers (passengers and freightors (people etc., sending and
receiving freight)),
users (people coming to send or receive passengers, and people
coming to send or receive freight).

The above example is typical of the kind of rough sketch, or even
narrative, documentation text that the software developer has to
produce in the course of describing a domain.
But the above list is merely indicative, not final.
It is also given here to “augment” our characterisation of what is
meant by the stakeholder concept.
Thus you can take this listing as a cue to construct similar
stakeholder listings for other domains.

Stakeholders from this “extrovert” domain are (thus) typically:
  The client
    ◦ contract management,
    ◦ client users
    ◦ and customers,
affected by the contracted software; and
  the software house
    ◦ contract management,
    ◦ software engineers
    ◦ and supporting technicians.

Characterisation 9.154 COTS Stakeholder:
  By COTS stakeholders we (thus) typically mean:
  ◦ people from software houses:
    ◦ software house owners and management (at least two groups),
    ◦ marketing, sales and service departments (three groups),
    ◦ the programmers, i.e., the software engineers,
    ◦ distributors of the software, and
  ◦ other software houses which base tailor-made software development on
COTS,
  as well as people from the application domains for which the software house
makes these products:
    ◦ customers (clients) and
    ◦ users

Software Development Stakeholders
One can identify two extremes of software (SW) developments:
turnkey software and commercial off-the-shelf software (COTS).

Turnkey Software Development Stakeholders
Characterisation 9.152 By turnkey software we understand
software that is developed — usually from “almost scratch” — in
very specific response to a specific client/developer contract

Characterisation 9.153 By a turnkey software development
stakeholder we thus understand a stakeholder from the software
developer or from that client domain

Commercial Off-The-Shelf (COTS) SW Development Stakeholders
  By COTS we mean
  ◦ generic kinds of software, i.e.,
  ◦ software whose functionality is as much, or more, decided upon
by the makers of the software than by the customers and users of
the software;
  ◦ software which is expected to cover, or actually covers the needs
of many clients,
  ◦ and which the maker thus expects to sell in dozens, hundreds or
thousands of copies.

Purpose of Listing Stakeholders
  lest we forget it, let us remind ourselves why we wish to
systematically record all possibly relevant stakeholder groups:
  it is so that we can systematically and “near exhaustively”
consider all relevant stakeholder groups;
  when we now go on to ascertain their view of, their perspective
of, the universe of discourse — here the domain.
Characterisation 9.155

By a stakeholder perspective we understand

- the, or an, understanding of the domain shared by
- the specifically identified stakeholder group —
- a view that may differ from one stakeholder group to another
- one stakeholder group of the same domain

- And thus two or more such group perspectives can give rise to
  - inconsistent, and/or
  - conflicting
  - overall views on domain attributes and facets.
- We shall return to the above issues when we later treat the methodological concerns of domain acquisition and validation.

All of the above stakeholder groups have perspectives that primarily focus on their shared domain: the general application area.

This is in contrast to the perspectives of stakeholders of the software house, the developer with whom the client contracts.

Besides wishing to secure, in their perspectives, the professional integrity of their company, the software house developer perspectives include those of satisfying the client.

The management groups have the following kinds of functions.

- Strategic management has to do with upgrading or downsizing, i.e., converting an enterprise’s resources from one form to another — making sure that resources are available for tactical management.
- Tactical management has to do with temporal, typically medium- to long-term scheduling and spatially allocating these resources, in preparation for operations management.

For each stakeholder group we have to investigate (elicit, acquire, and analyse) its perspective with respect to

- each of the possible domain attributes, as covered next, and
- each of the possible domain facets, as covered thereafter.

With respect to stakeholder perspectives we may be prepared to observe

- that one and the same phenomenon may be considered by two different groups to possess
  - not quite commensurate attributes,
  - and not quite commensurate facets.

The stakeholder perspectives for general application domains are generally of several concerns:

- Client executive and other upper-level management expects computing systems to improve their company’s competitiveness, financial position, etc.
- Tactical and operational management usually have perspectives that pertain to management and organisational issues.
- Non-management staff usually have perspectives that pertain to their daily work and to its interface with customers.

We now present a rather lengthy example that illustrates the interface between a number of stakeholder perspectives.

The stakeholders are (simplifying):

- an enterprise’s top level, executive management (who plan, take and follow up on strategic decisions),
- its line management (who plan, take and follow up on tactical decisions),
- its operations management (who plan, take and follow up on operational decisions),
- and the enterprise “workers” (who carry out decisions through tasks).

After some analysis we arrive at the following:

- Let R, Rn, L, T, E and A stand for resources, resource names, spatial locations, times, enterprises (with their estimates, service and/or production plans, orders on hand, etc.), respectively tasks (actions).
- SR, TR and OR stand for strategic, tactical and operational resource views, respectively.
  - SR expresses (temporal) schedules: which sets of resources are either bound or free in which (pragmatically speaking: overall, i.e., “larger”) time intervals.
  - TR expresses temporal and spatial allocations of sets of resources, in certain (pragmatically speaking: model finer-grained, i.e., “smaller”) time intervals, to certain locations.
  - OR expresses that certain actions, A, are to be, or are being applied to (parameter-named) resources in certain time intervals.
We introduce a number of functions.

- These partial, including loosely specified, and in cases nondeterministic functions \( \text{srm} \), \( \text{trm} \) and \( \text{orm} \) stand for strategic, tactical, respectively operations resource management.
- Let \( p \) be a predicate which determines whether the enterprise can continue to operate (with its state and in its environment, \( e \)), or not.
- To keep our model small we have had to resort to a “trick”:
  - putting all the facts knowable and needed in order for management to function adequately into \( E \)
  - Besides the enterprise itself, \( E \) also models its environment:
  - that part of the world which affects the enterprise.

### Strategic Resource Management

- \( \text{srm} \) proceeds on the basis of the enterprise: as it is now (\( e \)), and as one would like it to become (\( e^{\text{mem}} \)), as well as its current resources (\( r_s \)).
- \( \text{srm} \) “ideally estimates” all possible strategic resource acquisitions (upgrading) and/or downsizing (divestments) (\( s_{rs} \)).
- And \( \text{srm} \) selects one desirable strategic resource schedule (\( s_{sr} \)).

### Tactical Resource Management

- \( \text{trm} \) proceeds on the basis of the enterprise: as it is now (\( e \)), and as one would like it to become (\( e^{\text{mem}} \)), as well as one chosen strategic resource view (\( s_{r} \)).
- \( \text{trm} \) “ideally calculates” all possible tactical resource possibilities (\( s_{ts} \)).
- And \( \text{trm} \) selects one desirable tactical resource schedule and allocation (\( s_{tr} \)).
- Again, \( \text{trm} \) cannot be fully algorithmised.
- But some combinations of partial answer computations and decision support can be provided.

### Actual Enterprise Operation

- \( \text{ope} \),
  - enables, but does not guarantee,
  - some “common” view of the enterprise.
- \( \text{ope} \) depends on the views of the enterprise, its context, state and environment, \( e \), as “passed down” by management.
- and \( \text{ope} \) applies, according to prescriptions kept in the enterprise context and state, actions, \( a \), to named (\( m:R_{n} \)) sets of resources.
- The above account is, obviously, rather idealised,
  - but, we hope it is indicative of what is going on.
9.3.1 Perspectives of General Applications

The enterprise reinvoication argument, n°,

- a result of operations,
- is intended to reflect the use of strategically, tactically and operationally acquired,
- spatially and task allocated and scheduled resources,
- including partial consumption, “wear and tear”, loss, replacements, etc.

The let \( e^{m} : E \rightarrow p(e^{m}, e^{m}) \) in \ldots shall model a changing environment.

The latter is the interesting one:

- Solution, by iteration towards some acceptable, not necessarily minimal fix-point, “minims” the way the three levels of management and the “floor” operations change that state and “pass it around, up and down” the management hierarchy.
- The operate function “unifies” the views that different management levels have of the enterprise, and influences their decision making.
- Dependence on \( E \) also models potential interaction between enterprise management and, conceivably, all other stakeholders.

The let \( e^{m} : E \rightarrow p(e^{m}, e^{m}) \) in \ldots, which “models” the changing environment, thus summarises the perspectives of “all other” stakeholders!

We are modelling a domain with all its imperfections:

- We are not specifying anything algorithmically;
- all functions are rather loosely; hence partially defined; in fact only their signature is given.
- This means that we model well managed as well as badly, sloppily or disastrously managed enterprises.

We remind the listener that — in the previous example — we are “only” modelling the domain!

That model is, obviously, sketchy.

But we believe it portrays important facets of domain modelling and stakeholder perspectives.

The stakeholders were, to repeat: strategy (“executive”) management \((smr, p)\), tactical (“line”) management \((trm)\), operations (“floor”) management \((orm)\) and the workers \((ope)\). The perspective being modelled focused on two aspects:

- their individual jobs, as “modelled” by the “functions” \((smr, p, trm, orm, ope)\), and
- their interactions, as “modelled” by the passing around of arguments \((e, e', e^{m}, e^{m})\).

We can, of course, define a great number of predicates on the enterprise state and its environment \((eE)\), and we can partially characterise intrinsics — facts that must always be true of an enterprise, no matter how.

If we “programme-specified” the enterprise then we would not be modelling the domain of enterprises, but a specifically “business process engineered” enterprise.

Or we would be into requirements engineering — we claim.


**Perspectives of Software Development**

- If the application domain is that of software development itself
  - either just domain engineering,
  - or just requirements engineering,
  - or just software design,
  - or the first two, the last two, or all three of the above,
  as is the subject domain of these lectures,

---

**Discussion: Stakeholders and Their Perspectives**

**General**

- This lecture has discussed the concept of stakeholders.
- In subsequent lectures we shall take up the thread and, occasionally, indicate where we differentiate, in our descriptions, etc., between perspectives of different stakeholders.
- In a subsequent lecture’s treatment of business processes and management and organisation we may occasionally refer to the need for special descriptions of stakeholder perspective.

---

**Principle 9.38 Domain Stakeholder Perspective:**

- At the very outset of a development project
- define, together with designated domain stakeholders,
- their roles, their “jurisdictions” and their “rights and duties”.
- Be prepared, throughout a project, to revise the roles of domain stakeholders.

---

**Techniques 28 Domain Stakeholder Liaison:**

- Typically such communications deal, as we shall see later, with:
  - role assignments,
  - acquisition and
  - validation

**Tools 9.12 Domain Stakeholder Liaison:**

- The tools mentioned under information documents (Slide 277) apply equally well here.