

## Tools-and-Materials Pattern Language

### Task 1: Basic Tools and Materials

For this task, it may help if you look at some additional literature on tools and materials. I suggest [1], but you can, of course, use other material from the library or from the web, too. For example, you can find lots of material at [www.jwam.org](http://www.jwam.org).

**1a) Task:**

Explain the terms Leitmotiv (*Leitbild*), usage model (*Benutzungsmodell*), and design metaphor (*Entwurfsmetapher*). What are the relevant instances in the context of the tools and materials approach?

**Solution:** The definitions can be found in [1]:

**Usage Model (*Benutzungsmodell*)** is a domain-oriented model of how the software can be used to perform activities in its usage context. It is sensible to use design metaphors based on a Leitmotiv to realise a usage model.

**Leitmotiv (*Leitbild*)** is (in software engineering) a common frame of orientation for the parties concerned in development and usage of the application. It supports design, usage, and evaluation of software and is based on values and objectives.

An example *Leitbild* is the notion of the “Scientist” as a person who, without prejudice or religion, analyses and explains the world from objective mathematical or empirical findings.

**Design Metaphor (*Entwurfsmetapher*)** is a figurative concept used to make a leitmotiv more concrete. A design metaphor is always connected to a constructive interpretation in the form of construction plans and design patterns.

The Tools-and-Materials approach uses the leitmotiv of the *Arbeitsplatz für eigenverantwortliche Expertentätigkeit* (Workplace for independent expert work). The primary objective is a supportive and flexible behaviour of the system. The user is envisioned as an expert in his/her domain who needs to be able to use tools flexibly as the situation may require. The two main design metaphors are material and tool: A material is anything that can be manipulated by the expert. A tool is something that enables such a manipulation. Tools may be materials themselves—for example, they may be configurable. Further metaphors are: *work environment* (possibly separated into *rooms*), *automaton*, and *container*.

**1b) Task:**

How do tools use materials? What is the construction pattern (it’s called a ‘design pattern’ in [1]) behind the coupling of tools and materials?

**Solution:** Users want to be able to manipulate a material with different tools. Also, the same tool may be used for different kinds of materials. This is achieved by making tools (for example, a `List` tool) work on aspects (for example, something `Listable`), which are implemented by materials (for example, a `FinancialRecord`). This can, for example, be implemented by using `ADAPTER`, `DECORATOR`, or `EXTENSION OBJECT`.

**1c) Task:**

What parts do tools consist of? How is the communication between these parts organised? Why is this beneficial?

**Solution:** Tools consist of a function class and an interaction class. This separates the tools functionality from the precise means of accessing this functionality. The interaction class directly accesses the function class, however, the function class communicates with the interaction class only via an instance of the OBSERVER pattern.

**1d) Task:**

Explain the layers in an application built following the tools and materials approach. What constraints must be observed regarding communication and generalisation?

**Solution:** There are three layers: GUI (tool interaction classes), business functions (tool function classes), and data (materials). Direct communication (including the directed associations for purposes of navigation) always goes from higher layers to lower layers. Communication in the other direction uses (n)T-H, or TH hooks.

## Bibliography

1. Heinz Züllighoven et al. Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material-Ansatz. dpunkt.verlag, 1998

## Task 2: Financial Consultancy Tool Smithery

Financial consultants need to perform a variety of tasks. They plan investments, set up credit plans, sell retirement plans etc. They maintain a client folder for each client, which contains records of all transactions performed for this client. For example, there is a credit folder which contains data on all credits and loans handed out to this client, and informs about their status. There is also a sheet containing information about the current income and regular costs of the client; this defines the client's monthly budget. There is also material on all insurances the client has obtained.

When the financial consultant wants to plan a new credit for a client, he must understand the monthly income and obligations of the client. He may also need to analyse the client's investment plan, potentially adjusting it to make room for the credit back payment.

Using the tools and materials approach, design an application that will support the financial consultant's work.

**2a) Task:**

What materials can you determine from the description above?

**Solution:** Possible materials are `client folder`, `client folder archive`, `investment plan`, `credit plan`, `retirement plan`, `budget sheet`.

**2b) Task:**

What tools will the financial consultant require in the course of planning a new credit? What materials do they work on?

**Solution:**

- `Client folder extractor` to extract a `client folder` from the `client folder archive` and to open it on the desktop.
- `Credit creator` to create a new `credit` in the `credit plan`.
- `Budget sheet inspector` to analyse the monthly budget.
- `Investment plan inspector` to inspect the client's `investment plan`.

- Investment plan editor to modify the client's investment plan.

**2c) Task:**

Compare the solution above to a solution that would have come out of a more standard design process. What are the advantages?

**Solution:** Standard development processes use use cases and associated scenarios as a basis for software design. The danger is here that only the exact scenario is supported, but small deviations—which happen in everyday life—are prevented by the software. This is the typical “Sorry, the computer cannot do this” syndrome, where using the computer actually disallows things that used to be possible in pre-computer times. In the tools and materials approach, the standard scenario is also supported, but because it has been divided among several tools, these tools can also be recombined to allow for some deviation from the standard scenario. It is of course still the responsibility of the software developer to make the right decisions when designing the individual tools.