Understanding product innovation using Complex Systems Theory.
Frenken, K.

Citation for published version (APA):
Chapter 3

A generalised model of product innovation in complex systems

Analogous to random mutation as the source of variation in biological evolution, search in a design space has been modelled in the previous chapter as an algorithm that randomly mutates one or more elements of a complex system. The assumption of random mutation follows the majority of models in evolution economics (Nelson and Winter 1982; Andersen 1994; Auerswald et al. 2000; Kauffman et al. 2000). The argument holds that the outcome of investments in Research and Development are inherently uncertain, and can therefore best be modelled as a random move in a design space.

Although random mutation models in evolutionary economics acknowledge the inherent uncertainty of investments in R&D, the conception of innovation as an exogenous event still prevails. Assuming that innovation occurs randomly implies that innovation is conceptualised as exogenous to agents’ behaviour. To some extent, evolutionary models have endogenised innovation, for example, by assuming that the rate of mutations is a function of the level of R&D investment (Nelson and Winter 1982). However, the direction of innovation is still randomised as innovations are modelled as stochastic moves in whatever direction in capital-labour space.

Several students of technological development, however, have stressed the non-random nature of technological innovation. In their own appreciative work on technological development, Nelson and Winter (1977) themselves stressed that firms apply heuristics that, on the one hand, define a number of core technological concepts and, on the other hand, guide innovative activity in particular directions of improvement. Similarly, Dosi (1982) speaks of technological trajectories of incremental and cumulative innovations that are based on technological paradigms. These series of innovations are not “blind” but based on a particular knowledge base and a particular set of functional attributes of the technology. Heuristics in search thus include both means and ends, i.e. (i) guidelines concerning the elements in which innovations take place and do not take place, and (ii) guidelines concerning the functional attributes to be improved and to be sacrificed.

The possibility of directed instead of random mutation in human problem-solving constitutes an important difference between technological evolution and biological evolution. In biological evolution, genes have an equal probability to be mutated as mutations are considered “copying errors” in the reproduction process of genes from parent to offspring. The state of the selection environment bears no influence on the probability of a gene to become subject of mutation. By contrast, human problem-solving in technological systems is generally motivated by improving particular functional attributes of the system. More precisely, before designers engage in search they generally have a mental model of the future user group. The mental model guides search into particular directions rather than in other directions. In modelling terms, the mental model restricts mutations to take place only in particular elements of a system, which are considered instrumental for improvements of particular kinds.

With the introduction of the term “user-producer interaction”, Lundvall (1988) emphasised the importance of the coupling between design activity with particular user
groups. Different social groups can be distinguished that put different meanings to a technology and have different interests regarding the directions in which it is to be improved in the future (Pinch and Bijker 1984). Search in technological systems is therefore not characterised by randomness but by bias. Depending on the user group that designers have in mind as the future users of their technology, search is directed in particular directions rather than other. The evolutionary metaphor of "the struggle for survival" in biological evolution does no longer apply to the evolution of technologies. Rather, one can consider technological evolution as being dependent on the "social struggle" between different user groups that try to influence the future direction of technological development in different directions. Possibly, the existence of different user groups leads designers to follow a strategy of product differentiation rather than aiming at a design that suits all user groups.

This chapter contrasts random search to non-random search in complex technological systems. The objective of this chapter is to show a way in which the directed nature of technological innovation can be modelled as resulting from the interaction between designers' search activity and particular user groups. To model non-random and random search within the NK-model, the NK-model is first extended to include a more general representation of the selection environment. Building on work by Altenberg (1995, 1997), I generalise Kauffman's (1993) original NK-model of complex systems to include systems that are selected on the basis on multiple functional attributes (speed, size, comfort, safety, etc.) instead of a single functional attribute. Using this generalised model, trade-offs between different functional attributes can be formally represented. Different user groups can be distinguished by the way they rank the different functional attributes.

A search algorithm will then be specified that is different from random mutation. The alternative search strategy, called "function space search", sequentially optimises functional attributes starting with the most preferred and ending with the least preferred attribute. The most preferred attribute is optimised by directing mutations solely to those elements that affect the level of this attribute. The second most preferred attribute is optimised by directing mutations to only those elements that affect this attribute but that do not affect the most preferred attribute that already has been optimised. The third most preferred attribute is optimised by directing mutations to only those elements that affect this attribute but that do not affect the most preferred and the second most preferred attributes that have already been optimised, et cetera. By means of this algorithm, designers are able to direct mutations to sequentially optimise attributes for specific user groups according to the ranking that characterises each user group. The application of the algorithm implies that different design solutions can be found for different user groups resulting in product differentiation. What is more, function space search is also a way to reduce search time as function space search is limited to subsets of elements that affect a particular function.

Apart from the generalisation of the NK-model to include the possibility of different user groups, the NK-model is also generalised to include innovations that are of a different kind than mutations in an element. Following Henderson and Clark (1990), I distinguish between modular, architectural, incremental, and radical innovation. Modular innovation corresponds to a mutation in one or more elements as in the original NK-model. It is shown how the three other types of innovation can be modelled in the (generalised) NK-model. The introduction of the types of innovation within the complex systems framework can be considered as further generalisation of the biological model into a model of complex technological systems.
3.1 Altenberg's Generalised NK-model

The NK-model is based on the idea that each element in a system performs its "own" sub-function within the system with regard to the attainment of one overall function on which selection operates. Each element \( n \) is conceived to have a particular fitness value \( w_n \) that reflects its functional contribution to the system as a whole. The fitness of the system as a whole \( W \) is derived as the average of the fitness of each individual element \( w_n \).

The system's fitness can be thought of as an unidimensional measure of performance. For example, as in the neoclassical theory of the production function, it can be thought of as the cost efficiency of a process technology in transforming inputs into a given output. It is in this way that Auerswald et al. (2000) and Kauffman et al. (2000) used the NK-model to model process technologies, or, as they call it, "production recipes". Each recipe refers to a particular design of a production system that contains \( N \) "operations" that collectively produce a given output. For each operation \( n \) there exists \( A_n \) alleles that refer to a set of discrete choices for this operation. Each string then refers to a recipe i.e. a particular organisation of the production process.

In these models, fitness of a technology is measured by labour-efficiency. This specification corresponds to the learning-by-doing model of Arrow (1962a) in that the efficiency of technologies is expressed in labour-efficiency assuming the capital/output ratio to be constant. The novelty in the model concerns the assumption of epistatic relations between pairs of operations, which render the labour-efficiency of an individual operation dependent on the choice of other operations. Firms can therefore not simply choose the optimal process technology as it is assumed in the original neoclassical model of the production function.\(^{59}\) They have to engage in hill-climbing on the fitness landscape by experimenting with different combinations of operations through trial-and-error. Search in "recipe space" through hill-climbing leads a firm to a locally optimal fitness peak with fitness defined as labour-costs per unit output.\(^{60}\)

 Though the NK-model can be straightforwardly applied as a model of search for designs of process technologies, it is argued here that the application of the NK-model to an individual product technology as a car or an aircraft is problematic. The fitness value of an individual element \( w_n \) refers to the performance of an individual element with regard to its sub-function in relation to the attainment of one overall function of the system as a whole. This specification is based on the idea that each element \( n \) performs an "own" sub-function within the system, and the degree in which this function can be realised \( w_n \), depends on the choice of alleles of \( K \) elements with which it is epistatically related.

This description falls short as a description of a system that is selected on the basis of multiple selection criteria as it is generally the case for product technologies. For example, an epistatic effect of a vehicle's suspension on a vehicle's engine renders the car's fuel-efficiency dependent on both the choice of allele of the suspension element and the choice of

---

59 As pointed out by Auerswald et al. (2000) and Kauffman et al. (2000), firms in this NK-model look for combinations of operations that are complementary. This model can thus also be considered as an elaboration of transaction cost theory (Coase 1937; Williamson 1985). In this theory, firms are considered efficient institutional forms of production that internalise the benefits arising from complementary operations (called "intranalities", Kauffman et al. 2000: 145). In this context, benefits from centrally coordinating individual operations stem from dynamic efficiencies: only when the choice of allele is coordinated with respect to the choice of other alleles high local optima can be found (cf. David 1994b: 213-215).

60 As shown by Auerswald et al. (2000), the fitness of strings found by successive mutations reproduces the logistic learning curve, which is a stylised fact in the cost per unit output (see also Section 1.1 of this study).
allel of the engine element. And, an epistatic effect of a vehicle's suspension on a vehicle's tires renders the car's comfort dependent on both the choice of allele of the suspension element and the choice of allele of the tire element. In this example, the suspension element affects two selection criteria of a car, and cannot be considered as having an "own" sub-function. *A generalised model of complex systems described by N elements and F functions is needed to deal with systems that are selected on multiple selection criteria.* This model has been developed by Altenberg (1995, 1997) in the context of the evolution of complex biological systems, and will be used here as a model of complex technological systems.

### 3.1.1 Genotype-phenotype map

Altenberg (1997) developed a generalised model of complex biological organisms, which can be used to model a complex product technology. Altenberg's generalisation of the NK-model provides a model of complex systems containing \( N \) elements \((n=1,...,N)\) and \( F \) functions \((f=1,...,F)\). In biological systems, for which Altenberg's generalised NK-model was conceived, an organism's \( N \) genes are the system's elements and an organism's \( F \) traits are the selection criteria. The string of genes constitutes an organism's genotype and the set of traits constitutes an organism's phenotype. At the level of the genotype of an organism, mutations take place, which are transmitted in its offspring. At the level of the phenotype of an organism, natural selection operates in terms of its relative success to produce offspring.

A single gene can affect one or several traits in the phenotype and a single trait can be affected by one or several genes in the genotype. The number of traits that is affected by a particular gene in the genotype is referred to as a gene's "pleiotropy". The number of genes that affects one particular trait in the phenotype is referred to as a trait's "polygeny" (Altenberg 1997). The structure of relations between genes and traits can be represented in a "genotype-phenotype map". This map specifies which genes affect which traits. An example of a genotype-phenotype map of a system with three genes \((N=3)\) and two traits \((F=2)\) is given in Figure 3-1-1.

![Genotype-phenotype map](image)

Figure 3-1-1: Example of a genotype-phenotype map \((N=3, F=2)\)

---

61 Matthews (1984) introduced the concept of pleiotropy in evolutionary organisation theory.
Analogously, a technological system can be described in terms of its $N$ elements and the $F$ functions it performs. Typical - by no means universal - functional attributes of technologies are speed, size, weight, comfort, safety, et cetera. The string of alleles describes the "genotype" of a technological system, and the list of functional attributes describes the "phenotype" of this system. Different from neoclassical production function models in which fitness of a technology is expressed by a single selection criterion (cost efficiency), the fitness of product technology is to be expressed by some function of the levels of multiple functional attributes (product quality).

Functional attributes of product technologies are generally called characteristics since Lancaster (1966, 1971, 1979, 1990) and Ironmonger (1972) introduced this concept in economic theory.\textsuperscript{62} Lancaster argued that demand for a consumer technology should be understood as demand for the bundle of service characteristics that is embodied in a product. Here, I follow Lancaster (1979: 17), who stated that:

\begin{quote}
"(i)ndividuals are not interested in goods for their own sake but because of the characteristics they possess."
\end{quote}

Where Lancaster described product technology solely in terms of the quality attributes that are taken into account by users, Saviotti and Metcalfe (1984) proposed to describe technologies in terms of both technical and functional attributes. They speak of "technical characteristics" referring to the technical dimensions of an artefact, and of "service characteristics" referring to functional attributes of an artefact. Technical characteristics are for example the type of engine, tires and materials used in cars, which are chosen by designers. Service characteristics are for example speed, safety and comfort, which are the dimensions that users take into account when they compare different designs. Importantly, designers can achieve particular levels of service characteristics only through the right choice of technical characteristics. Saviotti (1996: 66) speaks of a "twin" characteristics approach including an "imaging" pattern between technical and service characteristics.

The terminology of technical and service characteristics will also be used here throughout this study. The technical characteristics of a product concern the alleles of elements in a system (e.g., a car with a gasoline engine, spring suspension, and block brakes). The alleles of elements in a system are the subject of manipulation by designers as they search in design space by experimenting with different combinations of alleles. Service characteristics are the "selection criteria" of a product technology that users take into account in their purchasing decision.\textsuperscript{63} Following this conceptualisation, an artefact can be considered as the "interface" between the designer who constructs the internal system and the user who makes use of the services it provides in particular user contexts (Simon 1969 [1996]: 6; Saviotti 1986, 1996; Andersen 1991, 1994).

Analogous to the biological model, technical characteristics make up the "genotype" of a product technology and service characteristics make up the "phenotype" of a technology. Following Altenberg (1997), the product architecture of epistatic relations between elements regarding particular functions can be represented as a genotype-phenotype map. The example given in Figure 3-1-1 represents a technological system that contains three elements and two

\textsuperscript{62} See also Thiss and Norman (1994).
\textsuperscript{63} In the empirical framework proposed by Saviotti and Metcalfe (1984), technical characteristics and service characteristics can refer to discrete variable as well as continuous variables, while in Altenberg's systems framework technical characteristics refer to elements with discrete choices for alleles, and service characteristics refer to continuous characteristics expressing the degree of fitness with regard to a function.
functions. The epistatic relations in the system are such that the first and second elements epistatically affect the first function, and the second and third elements epistatically affect the second function. Each product architecture can be represented by a matrix that describes which elements affect which functions in a technology (cf. Saviotti and Metcalfe 1984; Saviotti 1996: 66).

The matrix describing the architecture of epistatic relations contains $F$ functions and $N$ elements yielding a matrix of size $F \times N$. I call the matrix the element-function matrix $M$, for which holds:

$$M = [m_{fn}], \quad f = 1, \ldots, F, \quad n = 1, \ldots, N$$

The cells in the matrix $m_{fn}$ indicate the presence or absence of an epistatic relation between function $f$ and element $n$. As in the original NK-model, "x" represents that function $f$ is affected by element $n$, and "-" represents that function $f$ is not affected by the element $n$.

Each element is assumed to affect at least one function. If an element does not serve a function it is redundant in the (functional) description of the system. And, each function is assumed to be affected by at least one element. When a function is not affected by any element it is zero by definition (Altenberg 1995). As an example, the matrix of the system's architecture in Figure 3-1-1 is given in Figure 3-1-2.

![Element-Function Matrix](image)

Figure 3-1-2: Element-function matrix of genotype-phenotype map in Figure 3-1-1

Each column in the matrix $M$ represents an element's pleiotropy vector. The number of x-values in a column $n$ equals the pleiotropy value of element $n$, which is the number of functions affected by element $n$. Each row in $M$ represents a function's polygeny vector. The number of x-values in a row $f$ gives the polygeny value of function $f$, which is the number of elements affecting a function $f$. Division of the sum of all pleiotropy values by the number of elements $N$ equals the system's average pleiotropy. Division of the sum of all polygeny values by the number of functions $F$ gives the system's average polygeny. In the example, the pleiotropy of the first and third element equals one and the pleiotropy of the second element equals two. The polygeny value of both functions equals two. In total, there exist four relations between elements and functions. Average pleiotropy thus equals $4/3$ and the average polygeny equals $4/2$.\(^{64}\)

\(^{64}\) The model also allows to group functions (cf. Alexander 1964). The polygeny of this aggregated function is derived from the two disaggregated functions. In the example of Figure 3-1-2, aggregating $f=1$ and $f=2$ in one function would yield one function with a polygeny value three. Mutatis mutandis, the same applies for disaggregation of one function in several more functions.

90
As explained by Altenberg (1997), an NK-system is a special case of the generalised element-function matrices. For NK-systems, it holds that the $F$ equals $N$. It also holds that the diagonal in NK element-function matrices is always characterised by presence ("x") of a relation between element and function, since each element affects its "own" function. Furthermore, the K-value in the NK-model implies that each function is affected by the same number of elements. Thus, in the NK-model the polygeny of each function is equal to $K+1$, while the pleiotropy of each element is on average equal to $K+1$. The NK-model thus covers a subset of all possible systems that can be represented by element-function matrices.

### 3.1.2 Fitness landscapes

The way in which fitness landscape are constructed for generalised element-function matrices follows the same logic as the original NK-model discussed in Chapter 2 (Altenberg 1997). For each element that is mutated, all functions that are affected by this element are assigned a new, randomly drawn value from the uniform distribution [0,1]. Total fitness is again derived as the mean of the fitness values of all functions:

$$W(s) = \frac{1}{F} \sum_{f=1}^{F} w_f(s)$$

(3.2)

A simulation of the fitness landscape of the example of the element-function matrix in Figure 3-1-2 is given in Figure 3-1-3 for all possible combinations between alleles in design space. In this simulation, a mutation in the first allele generates a random change in $w_1$, a mutation in the second allele a random change in both $w_1$ and $w_2$, and a mutation in the third allele a random change in $w_2$. Since only the second element affects both functions, the existence of local optima can only be related to different alleles of the first element (here 000 and 110). For each allele 0 or 1 of the second element, there exist an optimal set of other alleles.

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>000:</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>001:</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>010:</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>011:</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td>100:</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td>101:</td>
<td>0.2</td>
<td>0.6</td>
</tr>
<tr>
<td>110:</td>
<td>0.9</td>
<td>0.3</td>
</tr>
<tr>
<td>111:</td>
<td>0.9</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Figure 3-1-3: Simulation of fitness landscape of element-function matrix in Figure 3-1-2

^65 Other specifications of the total fitness as a function of fitness values of functions in which functions are not weighted equally are discussed in the next chapter.
The simulation in Figure 3-1-3 shows a fitness landscape in which the trade-off between optimising the first function in the local optimum 110 \( (w_1 = 0.9, w_2 = 0.3) \) and optimising the second function in the global optimum 000 \( (w_1 = 0.8, w_2 = 0.9) \). Hill-climbing on this fitness landscape by means of a one-element mutation strategy can end up in either optimum, depending on starting string and the sequence of mutations that follow.

3.1.3 Decomposability

The principle of decomposition as explained in Chapter 2 applies correspondingly to fitness landscapes of generalised element-function matrices. For architectures in which a function is affected by a subset of elements that do not affect any other function, decomposition of the system is possible. Search can then be decentralised at the level of non-overlapping subsets of elements, each of which affects a particular function. The relevant design space to search then concerns the set of design spaces of the subsystems.

The concept of cover size also applies correspondingly to fitness landscapes of generalised element-function matrices. The cover size of a system is again given by the size of the largest design space of subsystems. The example of the architecture in Figure 3-1-2 is non-decomposable since the subsets of elements that affect a function are overlapping. The second function affects both functions. When \( m_{12} \) would no be present ("1") but absent ("0"), the system would be decomposable (Figure 3-2). In the same way, a change in \( m_{22} \) from "1" to "0" also yields a decomposable system. In these examples, the cover size of the system would equal two as the largest of the two subsystems contains two elements.

<table>
<thead>
<tr>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>( x )</td>
<td>-</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>-</td>
<td>( x )</td>
</tr>
</tbody>
</table>

Figure 3-2: Example of decomposable element-function matrix

Optimisation of a decomposable system is achieved by optimising each single subsystem. In the example of Figure 3-2, to optimise the first subsystem containing only the first element, only two possible states have to be evaluated. To optimise the second subsystem containing the second and third element, four possible combinations have to be evaluated. The total number of trials thus equals six. The total amount of time required when optimisation is done in parallel equals four, since time is bounded by the size of the subspace of the largest subsystem.
Summarising this section, Altenberg's (1995, 1997) generalised model of complex systems provides one with a model of complex systems of all possible sizes and with all possible degrees of multi-functionality. From generalised element-function matrices, pleiotropy and polygeny values can be derived for each element and each function, respectively. Kauffman's (1993) original NK-model is shown to cover a subset of all possible complex systems, i.e. systems in which the number of elements equals the number of functions and in which each function has the same polygeny-value \((K+1)\). The fitness landscapes of generalised element-function matrices can be simulated according to the same logic as proposed by Kauffman (1993). Moreover, the principle of decomposability and the determination of cover size as discussed in Chapter 2 applies correspondingly to systems represented by the generalised NK-model.

3.2 FUNCTION SPACE SEARCH ON FITNESS LANDSCAPES

So far, I discussed the generalisation of the NK-model to simulate complex systems with any number of elements, any number of functions, and any architecture of epistatic relations. I now turn to a generalisation of the selection environment. So far, the selection environment has been represented by a fitness function that calculated system's fitness as the average of the fitness values of functions (Altenberg 1997):

\[
W(s) = \frac{1}{F} \sum_{f=1}^{F} w_f(s)
\]

However, as an empirical specification of quality of a product technology, this formula obviously does not account for the general case in which users may apply different weights to different service characteristics (Lancaster 1966). Put another way, the fitness function that calculated system's fitness as the average of the fitness values of functions, refers to the special case, in which users apply the same weights for all functions.

Generally, users do not assign the same weight to each characteristic but value some characteristics higher than others. In that case, fitness of a product technology can no longer be derived as the average of the fitness levels of individual functions, but needs to be derived by some weighted sum over the fitness levels of individual functions. The difference in valuations of functions as expressed in weights carry information for designers that they can use in deciding how to search. I will formulate below how this information can be translated into a search heuristic for designers.

3.2.1 Homogenous selection environment

Allowing for different values of weights for each service characteristic one gets for a linear summation of fitness values:

\[
W(s) = \sum_{f=1}^{F} \beta_f \cdot w_f(s)
\]
A homogeneous selection environment can then be defined by some set of weights \( \{\beta_1, \beta_2, \ldots, \beta_F\} \) that is applied by all users of the technology. The case in which there exist heterogeneous user groups that apply different sets of weights will be discussed in Section 3.2.2.

The general specification of the fitness as the weighted sum of the levels of service characteristics of a design \( \{\beta_1, \beta_2, \ldots, \beta_F\} \) does not alter the concept of a fitness landscape as discussed in Chapter 2. It is clear that different weights for different functions can eliminate local optima. Different from the case in which all characteristics are weighted equally, local optima are now more likely to be characterised by high values of the characteristics with higher weights. The number of local optima can thus be considerably less than in the case in which all functions are weighted equally. In the extreme case that users take only one characteristic into account, one weight \( \beta_f \) equals one and all other weights equal zero. As a consequence, only the designs with the highest value for the service characteristic \( f \) that is weighted by one, are (global) optima in the fitness landscape.

The fitness landscapes that are constructed by the weighted sum of the values of service characteristics, can be searched in exactly the same trial-and-error manner as discussed in the previous chapter. Importantly, in this generalised specification of the selection environment, an alternative search strategy is also possible. When service characteristics have different weights, the values of these weights can be ranked. The ranking of weights for service characteristics provides designers with information that can be used for a heuristic search strategy. In this context, a heuristic means a way to search the design space other than through random mutation (Simon 1969). Designers can now decide to start with optimising the most important function, then the second most important function, et cetera. This kind of search can be labelled “function space search” (Bradshaw 1992). Function space search can be contrasted to hill-climbing in which mutations in elements occur randomly. Hill-climbing has been termed “design space search” as opposed to function space search, as the latter search strategy starts from the ranking of functions before deciding what element to mutate, while the former search strategy does not take information on the selection environment into account (Bradshaw 1992).

A formalised model of function space, which is presented below, goes beyond the biological metaphor of hill-climbing in models of technological evolution. Hill-climbing in design space is based on random mutations that occur independently of the state of the selection environment analogous to random mutations in genes in biological evolution. However, as argued by many students of technology studies in different ways (Rosenberg 1969; Von Hippel 1976, 1988, 2000; Teubal 1979; Pinch and Bijker 1984; Clark 1985; Van

\[
\sum_{f=1}^{F} \beta_f = 1, \quad \beta_f > 0
\]

(3.4.2)

This is a relatively simple function sometimes applied in multi-criteria analysis of project selection in planning. This function implies that each function can infinitely be substituted by an increase in other functions. Various alternative functions exist to derive a fitness value or “utility” from a collection of characteristics (Lancaster 1966, 1971, 1979, 1990; Nijkamp et al. 1990). Among possible alternative specifications is the Cobb-Douglas function that describes fitness as the product of fitness levels with exponents adding up to one reflecting decreasing marginal utility. Furthermore, one may add minimum requirements of service levels for each function to reflect that technologies require minimum levels of service characteristics to be of use at all. In this way, one obtains a specification of finite elasticity of functions.
den Belt and Rip 1987; Henderson 1995; Jørgensen and Karmøe 1995), variation and selection in technological evolution cannot be considered as operating independently. 67

Designers that come up with new variations can be expected to take information on users’ preferences into account when deciding on their search strategy. 68 Using this information, designers can direct innovative activity to specific technical characteristics (alleles of elements) that are thought to raise the level of particular service characteristics (fitness levels of functions). 69

Moreover, changes in the preferences of users can induce product innovation by designers. In particular, at the start of a product life-cycle, users are not familiar with the uses of a technology and can therefore not be expected to have stable preferences. Instead, preferences are formed during the process of use of early product models (e.g., Pinch and Bijker 1984; Garud and Rappa 1994). When preferences change and the ranking in service characteristics also changes, designers can be expected to direct innovation to elements that affect the service characteristic that has become more important at the cost of less important characteristics.

In this view, the ranking of weights of functions carries information that designers can use to focus search in particular elements instead of mutating elements randomly. Preferences act as “focusing devices” for designers as they can concentrate search efforts on those elements that are known to affect the more important functions (Rosenberg 1969). Note that this perspective differs fundamentally from the biological mechanism of random mutation and natural selection, since the variety-creating mechanism in technological evolution is no longer independent from the selection criteria.

A search algorithm that takes into account information on users’ preferences, requires information on the matrix of the system’s architecture (which elements affect which functions), and on the selection environment (how do users weight the different functions of a technology). 70 Only when the matrix of epistatic relations is known, or at least approximately known, designers can focus mutations in elements because these elements affect a particular function.

The function space search algorithm involves a sequential optimisation of the various functions by means of trial-and-error within the subset of elements that affect each function. Search starts in the subset of elements that affect the most important function, then search continues in the subset of elements that affect the second most important function, et cetera. Once the most important function is optimised, the second most important function is optimised given the fixed alleles of elements affecting the first most important function.

67 Nelson and Winter (1977) choose to model innovation as occurring independently from the selection environment when they stated that the selection environment “takes the flow of innovations as given”, but they realised that “there are important feedbacks” (Nelson and Winter 1977: 49). Modelling innovation independently from the selection environment was methodologically motivated at the time: “(w)e are attempting to build conformable sub-theories of the process that lead up to a new technology” (Nelson and Winter 1977: 49).

68 This information can be collected, at least by approximation, by various means of communication with users (surveys, visits, prototype tests, etc.).

69 Note that when the designer is also the user, the communication of preferences to the designer is close to perfect. When the designer is not the user, the communication of preferences can be imperfect and comes at a cost.

70 To identify the matrix of relations between elements and functions that make up the system’s architecture, designers first need to experiment with a number of different designs to find out the precise relations between elements and functions as contained in a product’s architecture. In the model, this can be formally expressed by mutating each element one by one and evaluating for each mutation in one element which functions are affected. This procedure would require only N mutations.
Then, the third most important function is optimised given the fixed alleles of elements affecting the first and second most important functions. This procedure is repeated until all elements have been mutated. Note that only in the case that the most important function is affected by all $N$ elements (highest possible pleiotropy), function space search is equivalent to exhaustive search.

In the example of a fitness landscape of a system containing three elements and two functions in Figure 3-1-3, two sequences in function space are possible:

- optimising the first function and hereafter optimising the second function given the alleles of the first and second elements that are found optimal with regard to the first function.

- optimising the second function and hereafter optimising the first function given the alleles of the second and third elements that are found optimal with regard to the second function.

In the case that the first function is weighted higher than the second function ($\beta_1 > \beta_2$), function space search starts with mutations in the first and second elements since only these elements affect the value $w_1$. Since there are four possible combinations between alleles of the first and second element, four trials suffice to find the combination 11# that optimises the value for $w_1$ with $w_1(11#)=0.9$. Then, given alleles 11# for the first and second element, two trials suffice to optimise the third element with respect to $w_2$. The optimal allele for the third element turns out to be allele 0, so search halts at 110 with $w_2(110)=0.3$.

In the case that the second attribute is weighted higher than the first attribute ($\beta_1 < \beta_2$), search would start in the second and third element as only these elements affect $w_2$. Since there are four possible combinations between alleles of the second and third element, four trials suffice to find the combination #00 that optimises the value for $w_2$ with $w_2(#00)=0.9$. Then, given alleles #00 for the second and third element, two trials are needed to optimise the first element with respect to $w_1$. The optimal allele for the first element is allele 0, so search halts at 000 with $w_1(000)=0.8$. Thus, different rankings of weights of function can lead designers to different design solutions.\(^{71}\)

Function space search does not necessarily find the global optimum, since only the fitness of the most important function is globally maximised. The alleles chosen to maximise the most important function constrain further search activity aiming at improving the other functions. The important advantage of function space search over random search, however, is that search time is considerably reduced. Since elements that have been optimised earlier with reference to one function are no longer candidate for mutation with reference to another function, the number of possible mutations is considerably reduced.

For example, the search time required for function space search in the example above is six trials against eight trials required for global optimisation through random search. This reduces only time by 25 percent, but time reductions increase for larger systems. Consider an $N=5$ system with $F=2$ and polygeny equalling three for both functions. Logically, this implies that all elements affect one function (pleiotropy one) except for one element affecting both

\(^{71}\) Note that function space search is not equivalent to hill-climbing in decomposable subsystems as discussed in the previous chapter. Decomposition requires that (subsets of) functions are affected by non-overlapping subsets of elements. By contrast, function space search or "decomposition in function space" can be applied to non-decomposable systems too, since this algorithm allows for overlapping subset of elements. For example, in Figure 3-1, the subset of elements affecting the first function (the first and second element) overlaps with the subset of elements affecting the second function (the second and third element).
functions (pleiotropy two). In that case, optimising one function first will require mutations in a subset of three elements leaving two elements to optimise the other function. Function space search then requires $2^3 + 2^2 = 12$ trials against $2^5 = 32$ trails for exhaustive search, a reduction of 62½ percent. And, for an $N=7$ system with $F=2$ and polygeny equalling two for both functions, function space search takes $2^4 + 2^3 = 24$ trials against $2^7 = 128$ trails for exhaustive search, a reduction of 81¼ percent.

One can thus conclude that function space search provides designers with an important heuristic to reduce the design space without loss in fitness with regard to the most important function, but possibly with a loss of fitness in the other functions. Function space search is an example of a search heuristic that economises on search time and search costs at the cost of fitness of the solution. Depending on the cost per mutation and discount rate of search time, function space search may well prove economically a more optimal strategy than exhaustive search. Note again that if the most important function is affected by all $N$ elements, function space search is equivalent to exhaustive search and is not advantageous.

Several empirical examples exist that reflect what I called “function space search”. A highly elaborated historical analysis of function space search can be found in Bradshaw and Lienert (1991) and Bradshaw (1992) who explained the success of the Wright Brothers in aircraft design by means of a model of function space. Their thesis holds that the Wright Brothers were successful in building the most successful aircraft at the time within only few years because they searched only in particular areas of the design space using a function space search algorithm. By contrast, other designers in the nineteenth century searched in ways that resemble design space search through hill-climbing using almost random mutations (Bradshaw and Lienert 1991: 607-609).

The strategy of the Wright Brothers was based on isolating the primary functions of an aircraft first, constructing solutions for each function and only hereafter constructing a whole, full-size aircraft. They considered three functions to be most important: lateral control, sufficient lift, and sufficient thrust. They started in 1899 with tackling the problem of lateral control, for which they found a solution via warping the wings using a small experimental kit. They turned to the problem of realising sufficient lift in 1901 experimenting with different wing designs in wind tunnels to test exclusively on the properties of wings related to lift. In late 1902, they turned to the problem of realising sufficient thrust and started to experiment with propeller designs in the wind tunnel, which led to the propeller concept as a moveable wing with lift forward. The three solutions were combined in the first successful aircraft in 1903 and in its successor in 1905. The latter covered an unprecedented distance of 24 miles where contemporaries had never flown further then half a mile.

Bradshaw and Lienert (1991) and Bradshaw (1992) explain the success of the Wright Brothers by their application of the strategy of function space search. In contrast, contemporaries of the Wright Brothers searched in design space by constructing complete aircraft designs based on local search hoping to find a design that would meet all functions at once. These variations included 1 to 80 number of wings, three types of wing configurations, three types of wing placements, three types of wing angles five types of cambers of wings, four types of wing shapes, and three types of tail placements. The number of design dimensions and alleles per dimension amounted to an enormous size of the design space offering little chance for success for hill-climbing strategies (Bradshaw and Lienert 1991).

This example is illustrative for the efficiency of function space search. Another well-documented example of function space search in which selected problems were dealt with sequentially is Carlson’s (2000: 156) description of Edison’s design work on telephones. Furthermore, Clark’s (1985) description of the early development of car technology can also
be considered an example of function space search, which I will discuss in more detail in Chapter 4. Murmann and Tushman (1998) also mention a number of artefacts that developed through a time-sequence of solutions to problems.

3.2.2 Heterogeneous selection environment

In the previous section, a generalised model has been developed, in which the fitness / quality of a product was no longer derived as the average of the fitness levels of service characteristics, but as a weighted sum over the fitness levels of service characteristics. The weights that users attach to each of the F service characteristics \( \{ \beta_1, \beta_2, \ldots, \beta_F \} \) reflect one homogeneous user group, since each user assigns the same weight to each function.

A second generalisation of the selection environment can be made by allowing different users to assign different weights to the same functions. In this case one speaks of heterogeneous users and heterogeneous demand. This conceptualisation of heterogeneous users follows Lancaster’s characteristics approach, the starting point of which can be summarised by the following statement (Lancaster 1979: 17):

"Differences in individual reactions to the same good are seen as expressing different preferences with respect to the collection of characteristics possessed by that good and not different perceptions as to the properties of the good."

This quote underlines the basic tenet of the characteristics approach that consumers buy a product for their bundle of service characteristics irrespective of the technical characteristics ("properties") that are incorporated in the artefact. This means that one and the same artefact can be valued very differently by different users.

Heterogeneous users can be represented in terms of user groups where each user group includes all users that assign the same weights for all functions. The size of a user group can thus range from one to all consumers. For a \( G \) number of user groups \( g (g=1,...,G) \) there is corresponding \( G \) number of sets of weights describing the preferences of each group regarding the \( F \) number of service characteristics:

\[
\{ \{ \beta_{11}, \ldots, \beta_{F1} \}, \{ \beta_{12}, \ldots, \beta_{F2} \}, \ldots, \{ \beta_{1G}, \ldots, \beta_{FG} \} \} \tag{3.5.1}
\]

For each user group \( g \), fitness \( W_g \) of a design can then be derived by the following formula:

\[
W_g(s) = \sum_{f=1}^{F} \beta_{fg} \cdot w_f(s) \tag{3.5.2}
\]

\[
\sum_{f=1}^{F} \beta_{fg} = 1 , \ \beta_{fg} \geq 0 \tag{3.5.3}
\]

\[72\] This notion is also central in the Social Construction Of Technology approach (Pinch and Bijker 1984).

\[73\] Note that contrary to homogeneous users case, in the heterogeneous user case some groups can weigh a function by zero when they infer no utility from the function. In the homogeneous case, the specification of a non-valued function would be abundant.
Obviously, this specification of the selection environment includes the specification given above for a homogeneous selection environment as the special case in which \( G=1 \).

A product innovation can now be defined as an innovation in the technology that improves the quality of the product for at least one user group as measured by the weighted sum of the fitness levels of service characteristics on which users select. For product development to be successful for at least one user group, effective communication regarding preferences with users is pivotal (Von Hippel 1976, 1988, 2000; Teubal 1979).74

Using the generalised heterogeneous selection environment as specified in formula (3.5.2), one can model the design process for different user groups. In function space search, different user groups, which are characterised by a different ranking of service characteristics, lead designers to apply a different sequence in optimisation of functions for different users. One can now model the case in which a selection environment is characterised by two user groups in stead of one. User group one \((g=1)\) is characterised by a preference for the first function over the second function \(\{\beta_{11}>\beta_{21}\}\) and user group two \((g=2)\) is characterised by a preference for the second function over the first function \(\{\beta_{12}<\beta_{22}\}\).

Following the simulation of the fitness landscape in Figure 3-1-3, function space with priority of the first function over the second function leads to design 110 with \(w_1=0.9\) and \(w_2=0.3\). Priority of the second function over the first function leads to design 000 with \(w_1=0.8\) and \(w_2=0.9\). Thus, function space search for user group 1 would lead designers to design 110 and function space for user group 2 would lead designers to design 000. The trade-off between optimising the first function at the expense of the second function or vice versa, leads designers to different designs adapted to the preferences of different user groups. Concluding, function space search thus leads designers to different solutions adapted to different user groups reflecting trade-offs between functions (Teubal 1979; Frenken 2001a).

The existence of different user groups is, however, not a sufficient condition for horizontal differentiation to persist in a market where different product designs compete. It can be the case that one design is superior for all user groups. Consider two user groups for which holds \(\{\beta_{11}=0.6, \beta_{21}=0.4\}\) and \(\{\beta_{12}=0.4, \beta_{22}=0.6\}\). In this case, design 000 turns out to be optimal for both user groups as \(W_1(000)>W_1(110)\) and \(W_2(000)>W_2(110)\), since:

\[
(0.6 \cdot 0.8) + (0.4 \cdot 0.9) > (0.6 \cdot 0.9) + (0.4 \cdot 0.3), \text{ and } \\
(0.4 \cdot 0.8) + (0.6 \cdot 0.9) > (0.4 \cdot 0.9) + (0.6 \cdot 0.3)
\]

In this case, both user groups would opt for the same design 000. When heterogeneity in preferences is more dispersed, it is less likely that one design is optimal for all user groups. In the example, when the two user groups are characterised by \(\{\beta_{11}=0.9, \beta_{21}=0.1\}\) and \(\{\beta_{12}=0.1, \beta_{22}=0.9\}\), one obtains:

\[
(0.9 \cdot 0.8) + (0.1 \cdot 0.9) < (0.9 \cdot 0.9) + (0.1 \cdot 0.3), \text{ and } \\
(0.1 \cdot 0.8) + (0.9 \cdot 0.9) > (0.1 \cdot 0.9) + (0.9 \cdot 0.3)
\]

In this example, horizontal product differentiation is expected to occur because function space search for different user groups leads designers to different solutions, each of which is indeed superior for each user group.

74 And when users are not able to express preferences with regard to a product design they have not yet used, this communication process should be an iterative process in which preferences become gradually known after a series of product trails.
The example above deals with one of the "simplest" of complex systems with three elements and two functions, and with only two users groups in the selection environment. More often, complex technologies such as transport vehicles, machinery, buildings, consumer durables, software and services are characterised by many elements, many functions, and various user groups. Whatever the number of elements $N$, the number of functions $F$, and the number of user groups $G$, all such complex systems can be modelled in the generalised NK-model.

A number of propositions can be derived from the function space algorithm in complex NK fitness landscapes regarding the occurrence of product differentiation when different user groups exist. First, the number of user groups bounds the ultimate degree of horizontal product differentiation. Since individuals within a user group share the weights they apply, they also share the fitness values they assign, and thus share the opinions what design is optimal.

Second, the scope for horizontal product differentiation depends inversely on the decomposability of a system. When a system is completely decomposable (when all elements have pleiotropy one) each element only affects one function as in the example of Figure 3-2. In this case, all functions can be optimised independently. This means that whatever sequence of optimisation of function is chosen, the same set of alleles will be found. Function space search will then always lead to the same, globally optimal design, irrespective of the sequence of optimisation that is chosen. Put another way, for all user groups, the same design is the optimal design. When a system can be decomposed in less than $F$ subsystems, different designs may be optimal for different user groups as there exist subsystems with more than one function.

With regard to vertical product differentiation that can occur when users differ in the budgets that they are willing to spend, differentiation is not bounded by the number of user groups and the decomposability of the system. Vertical differentiation occurs when users belonging to the same user group in terms of the weights they apply in valuing different designs, still opt for different designs. This is the case when a design with lower fitness costs is preferred over a design with higher fitness, because the latter is too expensive. This can well be the case when the more fit design also incorporates more expensive alleles for some of the elements. In that case, vertical differentiation can take place between high-quality product designs sold at a higher price and low-quality designs sold at a lower price. The issue of vertical product differentiation is further elaborated in Chapter 4.

The many local optima in large complex systems reflecting various trade-offs between functions open up many ways to satisfy specific preferences of user groups. However, the many possibilities provide in itself no warranty that designers will come up with such a variety of designs. Apart from the search time and costs involved in the innovation of a new product that adds to the product variety, there may be production constraints involved. For designs that are produced in small amounts for a small user group, scale economies in production may turn out so low, that prices cannot cover production costs (Lancaster 1979, 1990). In that case, this user group will purchase another design adapted to another user group. Although this design has lower fitness $W_g$ for this user group, it is offered at a lower price as scale economies are larger. In Chapter 4, the trade-off between product variety and cost efficiency related to scale economies is discussed in further detail, in which I will point to the effects of economies of scale on product variety and the effects of economies of scope on product variety.
3.3 A formalisation of Henderson and Clark’s classification of innovations

In this chapter, I continue with a discussion of a classification of innovation in complex technological systems as proposed by Henderson and Clark (1990). This classification includes types of innovations other than mutations in elements as discussed so far. This classification concerns modular innovation, architectural innovation, incremental innovation, and radical innovation. Modular innovation corresponds to mutation as in the original NK-model. The other types of innovations can also be modelled in the (generalised) NK-model. The inclusion in the model of other types of innovation provides one with yet a more generalised framework of product innovation in complex technological systems. Below, I show how these four types of innovations can be represented in the generalised NK-model.

3.3.1 Modular innovation

Modular innovation corresponds to mutation in one or more elements *i.e.* a particular subset ("module") of elements. Henderson and Clark (1990: 12) define a modular innovation as a change in an element of a technological system without a change in its architecture (the element-function matrix). This type of innovation is represented in the generalised model of complex systems above as a mutation in an element by substituting one allele by another allele. A modular innovation thus corresponds to a mutation in one or more elements, by which designers move along one or more axes on a fitness landscape.

So far, I restricted the discussion of innovation in complex technological systems only to this type of innovation. Both the design space algorithm of hill-climbing and the function space search algorithm are based on mutation *i.e.* modular innovation. These algorithms differ in the sequence of mutations. In design space search the sequence is randomly determined (analogous to blind variation in biology) while in function space search the sequence is determined by the importance of different functions.

Modular innovation is common to the evolution of technological systems as many alleles of elements are mutated into new alleles in the course of time. Henderson and Clark (1990: 12) give the example of the replacement of analogue dialling device by a digital dialling device in telephones. Saviotti (1996: 103) gives another example of mechanical elements being substituted for digital elements in watches. Other examples concern the mutations that have taken place in the evolution of the car, motorcycle, and aircraft: the type of tires, suspension, brake, starting device, lighting, and many other elements have all been mutated in the course of its evolution (Constant 1980; Clark 1985; Vincenti 1990).

Modular innovation can be represented schematically as in Figure 3-3-1. In this example of a modular innovation, the allele of the second element changes from 0 to 1 leading to a change in design from 000 to 010. According to the architecture as represented by the matrix in Figure 3-3-1, this modular innovation changes the fitness levels of the first and second function from $w_1$ and $w_2$ to $w_1'$ and $w_2'$.

75 The term module is not taken to refer exclusively to a decomposable subsystem of elements that do not have epistatic relations with other elements in other subsystems as the term "module" is sometimes used (Langlois and Robertson 1992). Modular innovation refers here to any mutation in any subset of elements irrespective of the epistatic relations that exist within this subset and with elements outside this subset.
3.3.2 Architectural innovation

Though modular innovations constitute an important part of the evolution of product technologies, innovative activity is not restricted to modular innovation. The alleles of elements, constituting the design of a complex system make up one set of dimensions of a complex system. The epistatic relations represented by the element-function matrix, which constitutes what I called the system’s “architecture”, make up a second set of dimensions of a complex system. The absence \( m_{fn} = -\) or presence \( m_{fn} = x \) of an epistatic relation in the element-function matrix can also be subject of manipulation by designers. For example, changing the architecture of an automobile with a particular engine type placed in the front into an architecture with the same engine type placed in the back, re-organises the way elements epistatically interact. As a result, the same set of alleles of elements organised in a different way yield different fitness values for functions (Altenberg 1995, 1997). This can be modelled in the generalised NK-model of complex systems by changing present epistatic relations between a function \( f \) and an element \( n \) \( m_{fn} = x \) into absent epistatic relations \( m_{fn} = -\) and vice versa yielding a new set of fitness values \( w_f \).

This conceptualisation of architectural innovation is in line with Henderson and Clark’s (1990) definition. These authors defined an architectural innovation as a change in the way “in which the components of a product are linked together, while leaving the core design concepts (and thus the basic knowledge underlying the components) untouched” (Henderson and Clark 1990: 10). By contrast, a modular innovation corresponds to mutation of an allele of an element without changing the product’s architecture. Architectural innovation can thus be considered the opposite of modular innovation since an architectural innovation only changes the way in which elements interact and not the alleles of elements, while a modular innovation only changes one or more alleles of elements without changing the way in which elements interact.

So far, the possibility of architectural innovation has not been taken up in the models of technological evolution. Rather, it has been explicitly assumed by Auerswald et al. (2000) and Kauffman et al. (2000) that firms only innovate on NK fitness landscapes by modular innovation through local search within a given architecture of epistatic relations. They assume that \( K \) is “given by nature” (Kauffman et al. 2000: 145). The argument behind this assumption holds that the laws of chemistry and physics determine the epistatic relations between elements analogous to the fixed nature of epistatic relations among organism’s genes. However, even when one accepts the assumption of a fixed law-governed set of epistatic relations in biological organisms, the assumption is not justified in the context of technological evolution. There is no law that determines people to use a particular architecture that organises elements in a technological system in a particular way. The same set of alleles can be organised in different ways. For what concerns the use of architectures, there are empirical regularities but no laws (Dennett 1995 [1996]: 226-228).

Following the car example, the decision to place a given engine allele in the front of a car is not governed by any law, since the engine can as well be placed in the back. However, it may be the case that, at this moment in time, cars with engines in the front are superior according to the currently used technical characteristics (the alleles) and the current weighting of service characteristics. The observed regularity in the placement of the engine is thus ultimately governed by available element technologies and the weights of selection criteria and not by physical constraints. The placement of the engine may well again become subject to change in the future.
Historical accounts on the evolution of technologies show that a large part of design activity is devoted to the construction of different architectures (e.g., Constant 1980; Clark 1983; Sahal 1985; Vincenti 1990). The construction of the first car with the engine placed in the back is just one example. Henderson and Clark (1990: 12) give the example of the innovation of the portable fan. Portable fans contain the same set of alleles as the older fans mounted in the ceiling, but alleles in portable fans are differently organised in a system i.e. portable fans have a different architecture.

Another example of an architectural innovation has been the introduction of retractable landing gears in aircraft. Retractable landing gears significantly reduced the negative effects of the fixed landing gear on the aerodynamics and fuel efficiency of an airplane (Vincenti 1990). When landing gears were made retractable, the choice of a particular landing gear allele no longer affected the fuel of efficiency of an airplane (or at least to a far lesser extent). Thus, the allele of the element did not change, but the architecture changed in such a way that service characteristics acquired new fitness levels.

Schematically, an architectural innovation is represented in Figure 3-3-2. In the example in the figure, elements are re-organised in such a way that the first element no longer affects the third function. Thus, this architectural innovation does not involve any change in the choice of alleles of elements as indicated by design 000 at the left and the right of the arrow. The architectural innovation only changes one or more epistatic relations between elements and functions. In the example, the architectural innovation removes the epistatic relation between the first element and the third function. In the matrix of epistatic relations, this holds that $m_{31}$ changes from a presence of an epistatic relation ($m_{31} = \times$) into an absence of an epistatic relation ($m_{31} = \cdot$). The value of the fitness of the third function changes since the set of elements that affect the fitness of the third function has changed (Altenberg 1995, 1997). This change in fitness of the third function is indicated in Figure 3-3-2 by the change from $w_3$ into $w_3'$. Thus, an architectural innovation transforms the fitness landscape since the same design 000 has a new fitness value.

A particular important issue in this context holds that architectural innovation is often motivated by a strategy to render the system more decomposable. Designers often undertake architectural innovations with the goal to render the architecture more decomposable or nearly decomposable (Baldwin and Clark 1997, 2000; Von Hippel 2000; Schilling 2000). The example in Figure 3-3-2 is such an example of an architectural innovation that makes a non-decomposable architecture a decomposable architecture. As explained in Section 2.2 and Section 3.1, decomposable architectures have big advantages in search. Modular innovations in decomposable architectures are more time-efficient as the number of trials to search the design space can be reduced by restricting modular innovations to take place on the level of subsystems.

Architectural innovations can therefore be considered as “meta-innovations” in that these types of innovations are often motivated to facilitate future modular innovations by simplifying the architecture. The simplification can imply the construction of a fully decomposable structure or a nearly-decomposable system with a minimum of less epistatic interactions between subsystems (Sahal 1985: 64; Simon 1969). Simplification thus means here the removal of epistatic relations between elements and functions in the system such that future modular innovations in one of these elements create less negative by-effects in the other elements.
### Modular Innovation

<table>
<thead>
<tr>
<th>Design</th>
<th>Function Levels: ((w_1, w_2, w_3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>((w_1', w_2', w_3'))</td>
</tr>
</tbody>
</table>

Figure 3-3-1: Modular innovation in \(n=2\) changing \(w_1\) and \(w_2\) into \(w_1'\) and \(w_2'\).

### Architectural Innovation

<table>
<thead>
<tr>
<th>Design</th>
<th>Function Levels: ((w_1, w_2, w_3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>((w_1, w_2, w_3'))</td>
</tr>
</tbody>
</table>

Figure 3-3-2: Architectural innovation in \(m_{31}\) changing \(w_3\) into \(w_3'\).
**Figure 3-3-3:** Incremental innovation within allele \( n=2 \) changing \( w_1 \) into \( w_1' \)

**Figure 3-3-4:** Radical innovation in \( n=2 \) and in \( m_{31} \) changing \( w_1, w_2, w_3 \) into \( w_1', w_2', w_3' \)
The benefits of simplified architectures do not only concern the facilitation of future modular innovations to improve the technology for a given user group. Benefits also arise from the increased flexibility to change the design in response to new user groups or in response to changes in preferences of existing user groups. The larger the degree of decomposability or "modularity" of a design, the easier it becomes to design new products adapted to the preferences of new user groups. The lower pleiotropy of elements in decomposable architectures makes it possible to direct modular innovation to increase particular functions that are considered most important by a new user group without negative effects of other functions. Firms that design systems with architectures with a higher degree of decomposability have an important competitive advantage vis-à-vis competitors as they can respond quicker to changes in consumer behaviour. However, this competitive advantage comes at the cost of architectural innovations. A trade-off exists between design flexibility and the cost of construction of a modular architecture, an insight that has been recognised recently in the technology management literature (Sanchez 1995; Sanchez and Mahoney 1996; Baldwin and Clark 1997, 2000; Schilling 2000).

3.3.3 Incremental innovation

Apart from architectural and modular innovation, Henderson and Clark (1990) distinguish between incremental innovations and radical innovations. Incremental innovations refine the functionality of an existing string of alleles and an existing architecture without changing them. An incremental innovation can be understood as an improvement within an allele, which increases the fitness value of one or more functions that the allele affects. For example, innovations within a given allele of the engine element (e.g., innovations within the gasoline engine) can increase the fuel efficiency of a car. At the level of the description of a vehicle system, such an improvement is neither a modular innovation, since the choice of engine allele remains the same, nor an architectural innovation, since the matrix of epistatic relations remains the same. The improvement is the result of an innovation within the engine system itself, which takes place at a lower system-level (cf. Metcalfe 1995: 36). An example of an incremental innovation within the allele "0" of the second element affecting the fitness level of the first function is given in Figure 3-3-3.

It should be noted that although incremental innovations do not involve any change at the system level in the choice of allele or the architecture of epistatic relations, they can make up a great deal of the improvements over time (Sahal 1985). For example, the explosive rise in the processing power of computers is largely the result of innovations within the allele of the processing element. Only occasionally, modular and architectural innovations at the system level occurred (from vacuum tubes to discrete semiconductor device, and from discrete semiconductor devices to integrated circuits). Similarly, many improvements in cars, motorcycles, and aircraft are due to improvements within alleles, in particular the engine allele (Constant 1980; Clark 1985; Vincenti 1990).

However, the potential of incremental innovation is in turn crucially dependent on modular and architectural innovations. For example, scaling of performance of vacuum tubes created non-linear rises in heat, which in turn reduced their reliability. Moreover, the detection of defective tubes became ever more difficult in larger systems. These constraints

76 At this lower system-level, the innovation that gave rise to an incremental innovation at the system-level can itself be of any kind.
induced the search for alternative alleles that could be made smaller through the use of electric pulses, which eventually led to the substitution of vacuum tubes by transistors in computers. Similarly, the loss in reliability in transistors due to increasing the number of interconnections in transistors put constraints on further increasing their performance. The innovation of integrated circuits on chips of silicon eventually replaced the transistor technology and led to the development of the microprocessor. In short, series of incremental innovations within an allele at a lower system level are punctuated by modular and architectural innovations at the system level that enable further incremental innovation at the lower system level (Dosi 1982; Sahal 1985; Mokyr 1990).

3.3.4 Radical innovation

A fourth type of innovation that is distinguished by Henderson and Clark (1990) concerns radical innovation as opposed to incremental innovation. They define a radical innovation as a change in both alleles of elements and the architecture of epistatic relations. Put another way, a radical innovation takes place when a system undergoes both an architectural innovation and a modular innovation at the same time. As a result, both the string of alleles and the architecture of epistatic relations undergo change.

In the example in Figure 3-3-4, the radical innovation concerns a modular innovation in the second element as the design changes from 000 to 010, and an architectural innovation in the epistatic relation between the first element and the third function. As a result, the fitness levels of all three functions change. The modular innovation in the second element affects the first and second function according the architecture of the system. And, the architectural innovation in the epistatic relation between the first element and the third function changes the level of fitness of the third function. The radical innovation in this example thus changes the fitness values $w_1$, $w_2$ and $w_3$ into fitness values $w_1'$, $w_2'$ and $w_3'$.

3.3.5 Matrix classification of types of innovations

Following Henderson and Clark (1990: 12), the four types of innovation can be summarised in matrix as in Figure 3-4. The classification of the four different types of innovations follows from the definitions given above.

Modular innovation and architectural innovation lie on the one diagonal. Modular innovation refers to mutation in elements without a change in the architecture of epistatic relations, while architectural innovation refers to a change in epistatic relations without mutation in elements. Incremental and radical innovations lie on the other diagonal. Incremental innovation refers to innovation at a lower system level without mutation in elements and without a change in epistatic relations at the system level. Radical innovation refers to both mutation in elements and a change in epistatic relations. Radical innovation can thus be considered as a co-occurrence of modular innovation and architectural innovation.

77 Cf. Sahal (1985: 63, emphasis by Sahal): “(t)he thesis is advanced here that one of the most important clues to understanding the process of innovation is to be found in the web of links between the functional performance of a technology and its size and structure. Thus, it is conceivable that the origin of innovations lies in learning to overcome the constraints that arise from the process of scaling the technology under consideration.”
As shown in this section, all four types of innovations can be formally represented in the generalised NK-model of product innovation in complex technological systems. Using this classification, insights in innovation processes can be enhanced as the definitions of the different types of innovations follow from a formal representation of complex technological systems. Furthermore, the classification opens up new ways for simulating product innovation in evolutionary models as will be discussed in Section 9.4.2.

### 3.4 Summary

In this chapter, I developed a generalised NK-model of product innovation of complex technological systems. This generalised model combines elements of Kauffman’s (1993) original NK-model, Altenberg’s (1995, 1997) extended NK-model, Lancaster’s (1966, 1979) work on product characteristics, and Henderson and Clark’s (1990) classification of different types of innovations in complex product technologies. The model provides a generalisation compared to the NK-model in three respects.

First, based on Altenberg (1995, 1997), the NK-model has been extended for systems with any number of functions on which selection takes place. Each element can affect any number of functions (pleiotropy), and each function can be affected by any number of elements (polygeny). The application of Altenberg’s generalised model in the context of technological systems is in line with the distinction proposed by Saviotti and Metcalfe (1984) between technical characteristics of a system (the alleles of elements) and service characteristics of a system (the functions on which selection operates).

The second generalisation of the NK-model concerned the specification of a general fitness function, in which the service characteristics of a system are not averaged but weighted. This general specification of the fitness function enables one to introduce an essential feature of Lancaster’s (1966) characteristics approach to demand: that users are interested in consuming a bundle of characteristics of a product rather than the product “itself”. One then can introduce multiple user groups each of which is characterised by a different ranking of weights on service characteristics. A product innovation then can be defined as an innovation in the technology that improves the quality of the product for at least one user group as measured by the weighted sum of the fitness levels of service characteristics on which users select.
The third and final generalisation that has been put forward in this chapter is a classification based on Henderson and Clark (1990) that includes, apart from modular innovation that corresponds to mutation in the NK-model, other types of innovations (architectural, incremental, radical). It is shown how all innovations can be conceptualised in the generalised NK-model. I also pointed to the interplay between different types of innovation.

The generalised model of product innovation in complex technological systems provides new ways to model product innovation as an evolutionary search process. The algorithm of function space developed in this chapter in an example of a new way to model product innovation within this model. Different from the trial-and-error based on random mutation, function space is directed to improve a product design for specific user groups in that it follows the ranking of service characteristics that a user group applies.

Other possible applications of the generalised NK-model are discussed in Section 9.4 in the form of a number of further research questions. These applications include modelling (i) product life-cycle in Nelson-and-Winter models, (ii) the interplay between incremental, modular, architectural, and radical innovation, (iii) search heuristics other than design space search and function space search, (iv) inter-firm collaboration, and (v) the role of technical standards in coordinating innovative activity. In the next Chapter 4, the product life-cycle model is reinterpreted in the light of the generalised NK-model.