The role of feature integration in action planning

Inaugural-Dissertation
zur Erlangung des Doktorgrades
der Philosophie
an der Ludwig-Maximilians-Universität
zu München

vorgelegt von
Gijsbert Stoet
geboren in Voorburg

München, 1998
1. Gutachter PD Dr. Bernhard Hommel
2. Gutachter Professor Dr. Wolfgang Prinz

Ich möchte weiterhin Heidi John für die Korrektur der englischen Sprache, Bianca Pösse für die Korrektur der deutschen Sprache und Peter Wühr für die Lesbarkeitsüberprüfung danken. Auch Karl-Heinz Honsberg und Fiorello Banci, die für die Konstruktion der Reaktionstasten gesorgt haben, gilt mein herzlicher Dank.

Weiterhin möchte ich allen Kollegen danken für die motivierende, nette Arbeitsatmosphäre in unserer Abteilung.
Chapter 1  Action Planning and the Binding Problem .......................... 1
  1.1 Action Planning .............................................. 1
  1.2 The Binding Problem in Neuroscience and Cognitive Psychology ...... 3
  1.3 Distributed Representation of Action and the Binding Problem ......... 7
  1.4 Object Files, Action Files and Event Files .......................... 9
  1.5 Action Plans and the Code-Occupation Hypothesis ................... 13

Chapter 2  Testing the Code-Occupation Hypothesis .......................... 16
  2.1 The Experimental Approach .................................... 16
  2.2 Experiment 1 .................................................... 19
    2.2.1 Method ................................................... 20
    2.2.2 Results ................................................... 21
    2.2.3 Discussion ............................................... 22
  2.3 Experiment 2 .................................................... 23
    2.3.1 Method ................................................... 25
    2.3.2 Results ................................................... 25
    2.3.3 Discussion ............................................... 26
  2.4 Experiment 3 .................................................... 27
    2.4.1 Method ................................................... 28
    2.4.2 Results ................................................... 30
    2.4.3 Discussion ............................................... 31
  2.5 Experiment 4 .................................................... 33
    2.5.1 Method ................................................... 34
    2.5.2 Results ................................................... 35
    2.5.3 Discussion ............................................... 36
  2.6 General Discussion ............................................. 37
    2.6.1 Abstract Coding of Action Plans .............................. 38
    2.6.2 A Two-Process Approach to Action Planning ................. 39
2.6.3 Feature Binding as a General Principle in Cognitive Representation .................................................. 40

Chapter 3 The Generalized Code-Occupation Hypothesis ................ 41

3.1 The Common-Coding Approach ........................................ 41
3.2 From Common Coding to the Generalized Code-Occupation Hypothesis 44
3.3 Evidence for the Generalized Code-Occupation Hypothesis ........... 45
3.4 Facilitatory and Interfering Effects in Compatibility Paradigms ........ 46

Chapter 4 Testing the Generalized Code-Occupation Hypothesis ......... 49

4.1 Looking for S-R Overlap Costs with the ABBA Paradigm .......... 49
4.2 Experiment 5 .......................................................... 51
  4.2.1 Method .......................................................... 51
  4.2.2 Results .......................................................... 53
  4.2.3 Discussion ....................................................... 54
4.3 Experiment 6 .......................................................... 55
  4.3.1 Method .......................................................... 56
  4.3.2 Results .......................................................... 57
  4.3.3 Discussion ....................................................... 58
4.4 Experiment 7 .......................................................... 59
  4.4.1 Method .......................................................... 60
  4.4.2 Results .......................................................... 62
  4.4.3 Discussion ....................................................... 63
4.5 General Discussion .................................................... 64
  4.5.1 Differences Between Action Plans and Object Files ............. 65
  4.5.2 Response Priming is Based on Abstract Response Codes ....... 66

Chapter 5 Conclusions ..................................................... 68

5.1 Basic Conclusions ..................................................... 68
5.2 Implications of the Results ........................................... 70
5.2.1 Consequences for Theories of Action Planning ............... 70
5.2.2 Overlap Costs and Reversed Response Priming ............... 72
5.3 A Feature Integration Model of Action Planning ................. 74
  5.3.1 Assumption 1: Action plans have a distributed representation . 74
  5.3.2 Assumption 2: Feature codes have two properties: activation and
temporal pattern .................................................. 75
  5.3.3 Assumption 3: Integration through synchronization .......... 76
  5.3.4 Assumption 4: Feature integration costs time ................. 77
  5.3.5 Assumption 5: Code sharing leads to higher activation ...... 78
  5.3.6 Assumption 6: Action plans fall apart after the action goal is
  reached ................................................................. 78
  5.3.7 Assumption 7: Actions are represented on the common-coding
  level ................................................................. 79
5.4 Final Conclusions .................................................. 79

Chapter 6  German Abstract / Deutsche Zusammenfassung ............... 80

References .............................................................. 91

Lebenslauf .............................................................. 97
Chapter 1

Action Planning and the Binding Problem

In this chapter it is argued that the mechanisms underlying action planning are faced with the binding problem. First of all, I will discuss how action planning is conceptualized in cognitive psychology. Thereafter, I will describe the binding problem and how it is dealt with in neuroscience and psychology. In the third section the relationship between action planning and the binding problem will be looked at. In section 1.4 the file analogy for the temporal binding of information is discussed. Finally, a new hypothesis based on the considerations about action planning and the binding problem is postulated: the Code-Occupation Hypothesis.

1.1 Action Planning

Action planning is the preparation of goal-directed behavior. Most of the time, human behavior is guided by preset goals and intentions, and it is not so much driven by incoming sensory information alone. Thus, in order to understand the underlying processes of human behavior, it is important to understand action planning.

A problem for the study of action planning is that it is a very broad concept. On the one hand, action planning includes the preparation of long-term high-level behavior, like preparing a holiday. On the other hand, it includes short-term low-level behavior like pressing a response key. It is an open question whether such different types of action planning are based on the same underlying processes. At any rate, there are different terms for the types of action plans. Most researchers in the domain of action planning distinguish between motor programs and action plans. Motor programs are considered to represent prepared motor behavior on a very small temporal scale, whereas action plans may also represent behavior on a large scale. Furthermore, motor programming is considered to have no conscious component and is thought to lead to motor activity directly (cf. Rosenbaum, 1985). Nevertheless, even very short and simple actions are mostly guided by goals and intentions. The role of intention in such simple
tasks has been given much attention through the task-switching paradigms (cf. Rogers & Monsell, 1995).

Action planning, as I understand it, is the construction of a representation with two almost independent main parts. First, there is a goal structure representing the state that has to be reached. Second, there is a rule-like representation which represents the event (or a certain point in time) triggering goal- or a subgoal-oriented movements. For example, if you plan a meeting in the afternoon, the representation of the meeting itself would be the goal and that of the afternoon the triggering event. This definition is in comparable with high-level theories of action planning, which postulate the separation of the content of an action plan and the triggering event (cf. Goschke & Kuhl, 1993). As Goschke and Kuhl point out, humans in daily life form intentions and mostly postpone the actions until the appropriate moment of execution has come. In order to be able to realize action plans, humans need to remember both what has to be done and when it has to be done. The memory structure responsible for such tasks is called prospective memory. The reason that my definition of action planning is similar to definitions of high-level action planning is that the separation between the representation of the content and the event under which the plan has to be performed makes sense for low-level action planning, too: Even very simple actions can easily be postponed until some later point in time, and in the meantime, other actions can be performed. This aspect of simple actions will further be demonstrated empirically in the present work. It is especially this ability that points to neglected aspects of action planning, namely how the cognitive system successfully separates different action plans held in (prospective) memory.

My definition of action planning differs from that of a motor program in two aspects. First, after a motor program has been constructed, it represents the first to-be-executed action. In other words, if another action has to be performed, the already prepared motor program has to be deleted, so that the other action can be prepared and executed. My definition, however, permits the preparation of actions independent of the time of execution. Already prepared actions can wait in memory while other actions can be prepared and performed in the meantime. Second, the content of motor programs is
directly related to action parameters, whereas, in my definition, action representations can be in terms of anticipated action effects. This latter point is in accordance with the older conceptions of action coding (cf. Lotze, 1852; James 1890), and has also been considered as a valid conception of action planning in modern cognitive psychology (cf. Prinz, 1997c; see also chapter 3).

My definition of action planning posits a new question, namely which mechanisms are responsible for the separation of multiple coactivated action plans. Thus the question is how the cognitive system knows which of all activated action features belong to which action plan. This question is the core of the binding problem in action planning and will be elaborated on in section 1.3. Of course, this issue has been neglected in theories that only regard the programming of one action at a time. Moreover, I will restrict myself to extremely simple action plans, that is, to simple keypressing actions, well aware that this restriction neglects many important facets of action planning. The present work is, however, about the role of binding in action planning, and it is reasonable to study this issue in the most simple actions first. If even simple actions are confronted with a binding problem, there will be no reason to reject a generalization of this finding with respect to more complex actions.

Due to this new question in the domain of action planning, the hypotheses presented cannot and must not be directly compared with the work already done in action planning. Research in action planning has focussed on different aspects, like the serial-order problem, sensorimotor integration, the degrees-of-freedom problem, and the learning of skills (cf. Rosenbaum, 1991). Therefore, my work is not appropriate to validate or disprove the study of different aspects of action planning, as long as these aspects are not related to the binding problem in action.

1.2 The Binding Problem in Neuroscience and Cognitive Psychology

The binding problem refers to the question of how a cognitive system represents which feature codes belong together. Both neuroscience and cognitive psychology are faced with this problem because of the distributed nature of representations. Although
the problem is stated on different levels of description, psychological theories are clearly inspired by the most popular neurophysiological solution, which will be described below.

In neuroscience the binding problem is about the integration of parallel and distributed processed streams of information in the brain. For example, information received by the retina is processed in parallel in different pathways, which lead to different areas in the brain: Different areas represent the different features of a single object, for example color, motion, or depth (cf. Bruce & Green, 1990). Thus, the question is how the brain “knows” that the activity in these different areas are due to the same object. There are two basic solutions: combined coding and temporal binding.

The principle of combined coding is a solution of the binding problem, which assumes that there are neurons that only react to a special configuration of features. Such hypothetical cells are also called grandmother neurons, pontifical or cardinal cells. In the “single neuron doctrine” of Barlow (Barlow, 1972) it is stated that each decision of an animal can be traced back to the firing of a single pontifical cell. According to that approach, even complex objects can be represented by the activation of very few or even individual neurons. However, this solution is elusive, because the number of grandmother neurons, necessary to represent all possible feature bindings exceeds the number of neurons in the human brain. This conclusion is based on the consideration that every new object eventually encountered will require the recruitment of a new cell in the visual cortex (Sejnowski, 1986).

Temporal binding as a solution of the binding problem was first proposed by Von der Malsburg (1981). It is assumed that the correlated firing of neurons means that they take part in representing the same object. Thus, the temporal characteristics of spike trains contain important information about the relationship between feature-representing neurons. One of the most important advantages of this solution is that it allows the concurrent representation of various objects. In Hebb’s (1949) model of object representation by distributed neuronal assemblies, the simultaneous coactivation of multiple assemblies leads to unresolvable ambiguities. This problem is illustrated by Rosenblatt’s (1961) example of a perceptron, that is a simple perceptual neural network
consisting of just four neurons (see Fig. 1). Neuron 1 responds to the presence of a triangle and neuron 2 to the presence of a square. Neuron 3 responds to all objects in the upper visual field and neuron 4 to all objects in the lower visual field. If this system has to detect a square in the upper visual field, an output neuron would have to detect the simultaneous activity in neurons 2 and 3 (Fig. 1, left). But now suppose that there is triangle in the upper and a square in the lower visual field: The output neuron would falsely respond (Fig. 1, right). In other words, the perceptron can only handle one object at a time.

Figure 1. Left: Correct detection of a square in the upper visual field. Right: Perceptron cannot handle the coactivation of 2 objects.

The advantage of temporal binding is that coactivated object representations remain separable according to their individual temporal activity. It is hypothesized that the temporal binding hypothesis can be generalized to other domains than visual perception alone (Engel et al., 1997).

In psychology the binding problem is about the question of how several memory traces can be temporarily connected. There are many psychological domains in which this question arises. In perception: When more than one object is perceived, how is it represented which of all features involved belong together (cf. Treisman, 1996)? But also in other domains, this problem has been addressed. In language and reasoning: Rules are abstract descriptions which can be applied on concrete or other abstract entities. Such rules are necessary for reasoning and for language understanding and
language representation. The question and problem is how the rule representation is
temporally connected to the entities to which the rule is applied (Shastri & Ajjanagadde,
1993). It is interesting to note that Shastri and Ajjanagadde presented a connectionist
model, inspired by the temporal binding hypothesis, to solve this problem.

Questions concerning the binding problem in psychology date back to the 1930s,
when the Gestalt psychologists began to look for the criteria for feature integration of
visual scenes (Köhler, 1930). For instance, proximity or similarity were assumed to be
factors governing perceptual integration. In cognitive psychology the binding problem
relates to information processing: Which coding principles are used for representing that
two pieces of information temporarily belong together?

An important contribution to the discussion of dynamical binding in cognitive
psychology is the Feature Integration Theory of Treisman and Gelade (1990). This
theory relies on two main assumptions. First, it is assumed that perceptual feature codes
are the basic building blocks of object representations, and second that the integration of
the feature codes into object representations is an attention-demanding process.

Moreover, the binding problem has dominated the connectionist debate in the
past decade. Connectionism is an approach toward cognitive representation, mainly
studied through computer simulations (for an introduction and overview, see
McClelland & Rumelhart, 1986). The connectionist approach assumes that
psychological functions are based on the activity of large neuronal networks, rather than
on small local cognitive modules subserving some cognitive function. Now,
connectionist models on almost any cognitive function have become available.
Nevertheless, they have been attacked on theoretical grounds, claiming, for example that
connectionist models cannot flexibly bind concepts, opposed to symbolic models of
cognitive functioning. Different solutions have been put forward to solve this problem
in connectionist systems. For example, Shastri and Ajjanagadde (1993) presented a
model of reasoning using a conceptual connectionist network, that is a network in which
the individual nodes have a conceptual meaning, such as “House” or “Car”. The
dynamical binding of concepts was reached by representing the nodes by their
individual temporal pattern of activity, or signature. The temporal relationships between
concepts were represented by the signatures of the nodes. Thus, the model used empirical findings from the neuronal level to describe a mechanism on the conceptual level.

In section 1.4 a special approach to binding in cognitive psychology will be discussed.

1.3 Distributed Representation of Action and the Binding Problem

Action plans have a complex structure, which is reflected in different aspects. They include multiple components. Even elementary actions require the coordination of several body parts and the appropriate sequencing of muscle movements. For example, in order to grasp an object, the fingers, wrist, and arm movements and the spatio-temporal coupling of these movements with the goal object have to be coordinated (e.g., Jeannerod, 1981; Marteniuk, Leavitt, MacKenzie, & Athenes, 1990).

Another aspect of the complexity of actions is that they have multiple features or parameters, which have to be specified in such a way that coherent movements result. Again, to grasp an object, several features have to be selected, like the hand to be used, gripping force, speed of movement, and so on.

The fact that different brain systems participate in action control and the parameters of an action are specified by the different brain systems also shows the complexity of actions. Although older models of action control suggested the successive activation of three, hierarchically organized levels of representation (i.e. action goal, motor program, and basic commands) based on anatomical evidence for unidirectional pathways between the associative, premotor, and primary cortex, more recent data seem to disturb this notion: First, neuronal structures have a functional heterogeneity, that is, neurons taking part in one behavioral function can be distributed over different areas, and functionally different populations of neurons can be found in the same area. Second, a single neuron can be involved in different behavioral functions. Thus, it would be wrong to think of several specified neurons responsible for a certain function in motor
control. Third, different neuronal populations cooperate in the representation of a movement function (Requin, 1992).

Altogether, action plans have a multiplicity of components, both functionally and anatomically. This points to a binding problem in action planning, as there is the need for binding the different components. In the context of the neuronal coding of complex motor acts, this problem has been discussed by Singer (1994). He gives two possible solutions: First, the command-neuron solution, which greatly resembles the pontifical-neurons solution discussed. This command neuron is connected to the different neuronal populations taking part in the motor act. Nevertheless, no command neurons for complex motor acts have been found experimentally, as Singer (1994) points out.

The second solution is the temporal binding hypothesis, which relies on the synchronous firing of cell assemblies representing a motor act. According to this conception of motor representations on the neuronal level, motor programs emerge from the synchronization of activities of a large number of cells. And furthermore, the same cell can participate in more functions at the same time.

It is interesting that the binding problem in perception and action is rather similar. However, the binding problem in action control has not been an issue in cognitive psychology. There might be two reasons for this neglect. First, the binding problem can only be uncovered in psychological experiments if more than one action plan is activated at the same time. Yet, in most experimental paradigms participants are asked to perform one action at a time. This is also true for most dual-task paradigms, where the two tasks are planned and performed sequentially. However, in a situation in which two action plans have to be active at the same time, a temporal binding mechanism is essential to separate both action plans. Without such a mechanism, the cognitive system would be faced with the same problem as Rosenblatt’s Perceptron (see text and Fig. 1). Second, in psychology perceptual processes have been studied more often than processes of action coding (Prinz, 1997b). In sum, the binding problem has been occluded by experimental simplicity and by a preference for studying phenomena in perception rather than action.
1.4 Object Files, Action Files and Event Files

In the previous section several arguments for a binding problem in action have been given. Further, there are some indications for a similar binding mechanism in perception and in action representation on the neuronal level. Now I shall take a look at the proposed binding mechanisms in psychology and at experiments suggesting that response features are integrated in a similar manner as are perceptual features. Based on these ideas, I will propose a file concept for the representation of action plans.

Allport, Tipper, and Chmiel (1985) explained their negative-priming results with a binding mechanism. Negative priming occurs in experiments with to-be-attended and to-be-distracted stimuli: When a target stimulus had a distractor feature in the preceding trial, performance is worse compared to the situation where the target stimulus had no distractor feature. For example, assume there are two succeeding stimulus displays with a red and a green letter each, and that the participants are instructed to attend only letters of one color, for example red. If in the first display (called the “prime”) a red A and a green B are presented and in the succeeding presentation (the “probe”) a red B and a green C, the reaction to the red B is slower compared to a prime with a red A and a green D. The authors explain this negative priming with a temporal association between the identity and color for both the target and the distractor stimulus. Thus, the reaction to the target stimulus in the probe display is worse when it had a distractor feature in the prime display because the old association (between identity and color) has to be undone and the new feature association has to be established. Hence, negative priming depends on the alternation of feature conjunctions.
Figure 2 shows the four possible types of feature repetition and conjunction changes. On the left side the features of the “prime” and on the right side the features of the “probe” are depicted. In case of a complete match, the same object is presented both in the prime and the probe display. Thus, the temporal association established in the prime trial is also necessary in the probe trial. In case of a partial match, only one of the features is repeated, hence the association of the prime trial has to be undone and the new constructed. In case of complete mismatch between prime and probe, no association must be undone. Altogether, the partial match is the most time-consuming for the perceptual system.
Kahneman, Treisman, and Gibbs (1992) reported a comparable mechanism of feature binding in perceptual processing. In a reviewing paradigm two successive displays were presented, the preview and the target field. The preview field contained two or more letters, the target field one letter, which had to be named as quickly as possible. When the target letter was presented in both displays, naming times were somewhat faster. Interesting for the theory of binding is that naming was faster when the absolute or relative position was the same as in the preview field compared to the situation in which the absolute or relative position changed. The authors’ account of this result is similar to the above described account of Allport et al. (1985): For each object in the preview screen an object file is constructed that contains the identity and location of the preview letters. If in the target field the same letter appeared in the same location, the old object file could be “reused”, whereas a radical change between letter and position would ask for the construction of a new object file. The concept of an object file can be compared with the circles of Figure 2. The object file is a set of temporally interconnected features. When only some of these features need to be used again, the connections have to be undone and new ones are to be established, that is, there would be a partial match between preview and target object. Altogether, integration is easier with complete than with partial match, because in the case of a complete match the whole object file can be reused.

The object file as a cognitive structure for the temporal binding of stimulus features is well supported by the available data (Gordon & Irwin, 1996; Henderson, 1994; Henderson & Anes, 1994; Hommel, in press a). Nevertheless, Kahneman et al. (1992, p. 209) did not rule out that response information could become part of object files. Indeed, Hommel (in press a) showed that such response information can be integrated with stimulus information. In his study, participants gave two reactions to two stimuli in each trial. The reaction to the first stimulus was an already prepared single left or right keypress. The reaction to the second stimulus was a binary-choice keypress reaction signaled by one of its features (form or color). In addition to indications of feature integration, there were two indications for the temporal binding of stimulus and response features.
First, the second reaction was faster if the first response and the task-relevant feature for the second reaction was repeated. However, if only the response was repeated, but not the task-relevant feature for the second reaction, then response alternations yielded the better performance. Thus, a complete match between first and second reaction led to best performance, that is, with the repetition of both the task-relevant feature and the response, the second reaction is best. A partial match (response repetition, but alternation of the task-relevant feature) led to impaired performance compared to response alternations (no match). Altogether, this finding suggests the same feature binding mechanisms as discussed above, with only one addition: Response and stimulus features can be temporally bound into one episodic representation.

Second, the response-repetition advantage also depended on the feature “stimulus location”. If both stimulus location and response were repeated (complete match) or if both were alternated (no match) reactions were faster than with partial match. This finding suggests the same binding mechanism. After the first reaction, a temporal binding has been constructed between stimulus and response features. If only one member of the pair is repeated, that is, if there is a partial match between the first and second S-R configuration, reactions are slower.

Two other studies indicate the temporal integration of S-R relationships. First, Treisman (1992) had participants perform a number of tasks with the same nonsense patterns. Performance improved with practice, but there was little transfer between the tasks. Possibly, this was because task information or response was integrated with the object information, so that there was no longer access to acquired object files if the task changed. Second, Danzinger and Robertson (1994) had their participants respond to letter identity, while the color and location of the letters were also varied. They found that the repetition of the irrelevant stimulus features led to better performance than the alternation of these features, but that this effect was more pronounced when the response was also repeated. Thus, in this study too, a complete match is better than a partial match.

Altogether, there are several indications suggesting that the same mechanisms governing the temporal binding of object features also govern the temporal relationships
between stimulus- and response features. Therefore, Hommel (in press a) proposed to replace the more restricted term “object file” by the general concept “event file”. An event file can best be characterized as “an episodic memory linking codes of features belonging to an action-relevant object with codes of features characterizing the corresponding action.” (Hommel, in press a). Event files represent events, thus all stimulus- and action-related information that belongs to a situation.

1.5 Action Plans and the Code-Occupation Hypothesis

Now that it has been shown that there is ample empirical evidence for the dynamical binding between stimulus features and between stimulus and response features, it is tempting to assume that dynamical bindings also take place between response codes. In the above section several theoretical arguments have been presented that support this assumption: Action plans have a distributed representation, thus there must be some binding mechanism. Action plans only relying on action code activation would not allow the concurrent coactivation of several plans.

This assumption is in accordance with the study of Murthy and Fetz (1992). They reported correlated firing between cells in the sensorimotor cortex of the monkey in situations requiring action planning of a novel and complicated task. Furthermore, it is assumed that the dynamical binding of response codes is only necessary when there are no hard-wired associations between the task cues and the responses. Automatized and overlearned tasks may rely on such fixed associations, whereas novel tasks rely on temporal binding. This fits with the Murthy and Fetz data, which show that the synchronous firing was not found with the retrieval of an overlearned plan.

The question now is how this assumption can be tested in an experimental situation. In other words, what are the measurable consequences of the dynamical binding of response codes? I assume that the formation of a new action plan requires the dynamical binding of response codes, which is realized through the synchronization of the activity pattern of the action codes belonging to the same plan. Further, I assume that the activation of several plans can be distinguished by the individual temporal pattern of
activity belonging to each individual plan. It is my assumption that the synchronization of codes takes some time because the signals between the spatially distributed codes have to exchange their temporal patterns of activity in order to become synchronized.

Two important assumptions about the synchronization of codes are the basis of the Code-Occupation Hypothesis. First, it is assumed that one code can fire in synchrony with different patterns of activity at the same time. Second, it is posited that it is harder to synchronize a code that is already firing in synchrony with a different pattern of activity than to synchronize a code that was not firing in synchrony with any pattern of activity. This can more easily be understood with an analogy. Suppose that you have to tap with your finger in synchrony to a signal compared to the situation where you are tapping in synchrony with a signal and you have to synchronize to an additional signal. The former case can be compared to the situation in which an unsynchronized code has to become synchronized to a certain pattern of activity. The latter case can be compared to the situation where an already synchronized code has to become synchronized to an additional pattern of activity.

From the second assumption regarding the integration of action codes into action plans the Code-Occupation Hypothesis is derived: It is more difficult to integrate an already integrated action code into a new action plan than to integrate an action code which is not already part of a different action plan. This hypothesis is called Code-Occupation Hypothesis, because the integration of an action code into an action plan corresponds to an occupation: The code is occupied by the action plan and therefore it is not so easily available for other processes of action planning. The slower access time in case of code occupation is the price that has to be paid for the ability to bind one code into more than one action plan.

In chapter 2 the temporal binding of response codes is studied by exploring the influence of an already formed action plan, held in memory, on the construction of another action plan. According to the Code-Occupation Hypothesis, the construction of a plan should be more difficult if the plan shares an action feature with an already formed action code: The construction requires the temporal binding of a code which is already synchronized with other codes. Therefore, the new synchronization process
necessary for the planning of the new action will take more time resulting in longer latencies of overt behavior. For example, assume that an action plan has to be constructed for pressing a key with the left index finger in a standard experiment. Codes such as LEFT, INDEX FINGER and PRESS can be assumed to participate in the representation of the action plan. Before execution, these codes have to be integrated, that is their patterns of activity have to be synchronized. Once integrated, these codes will be less available for other processes of action planning.

It is important to point out here that the prediction of impaired performance in the case of code sharing by coactivated action plans can not only be derived from the synchronization logic, but it is also consistent with the general idea underlying the object- and event-file approach. The results of object- and event-file studies suggest that feature integration leads to impaired performance if there is only partial overlap between the current and the previous event compared to complete overlap or complete mismatch (see previous section). Applied to the case of overlapping action plans, one should expect worse performance if the currently to be planned action partially overlaps with a previous planned action, for example, the planning of a left-hand upward response following the planning of a left-hand downward response. Complete overlap means the exact repetition of the planning of the previous action while complete mismatch could mean, for example, the planning of a left-hand upward response following the planning of a right-hand downward response.
Chapter 2
Testing the Code-Occupation Hypothesis

This chapter introduces a paradigm for testing the Code-Occupation Hypothesis. Then, four experiments based on this paradigm are reported. The theoretical implications will be discussed in the General Discussion at the end of the chapter.

2.1 The Experimental Approach

The general logic behind the paradigm I developed for testing the Code-Occupation Hypothesis is that it measures how the construction of an action plan is affected by the representation of an already formed action plan temporarily held in memory. In the paradigm, two tasks are involved, Task A and Task B. The stimulus and response of Task A will be called Stimulus A and Response A. And similarly, the stimulus and response of Task B will be called Stimulus B and Response B.

Task B is embedded in Task A in the following way: First, Stimulus A signals Response A. Participants have to delay the response up to a defined point in time. The delay between Stimulus A and B is so long that one can plausibly assume that the representation of the to-be-delayed Response A has been formed before Stimulus B is presented (4 s in Exps. 1, 2, and 4, 2 s in Exp. 3). The representation formed directly after the presentation of Stimulus A is called Plan A. Stimulus B signals Response B, which has to be performed immediately. Altogether, this timing of events ensures that Task B is performed while Plan A is held in memory. Response A has to be performed after Response B. The sequence of events in this ABBA paradigm is shown in Figure 3.

Of course, it is just an assumption that Plan A is really formed between the presentation of Stimuli A and B. Alternatively to planning Response A before the onset of Stimulus B, the participant could memorize Stimulus A itself until the end of the trial and then prepare Reaction A. This problem will be treated in Experiment 3.
The participants sat in front of two response panels, one to the left and one to the right of their body midline. On each panel, there were three vertically arranged, touch-sensitive metal plates, the central one serving as the home key (at least in Experiment 1; deviations in this and other aspects will be pointed out in the Method sections). As shown in Figure 4, Task A was signaled by a left- or right-pointing arrow, accompanied by the digit 1 above or below it, or by a 1 and a 2, with the second digit always occupying the location opposite to the first. Arrow direction indicated whether Response A was to be performed with the index finger of either the left or the right hand. The digits indicated the direction of the response and the number of movement elements. The movement always started with releasing the home key with the hand specified by the arrow. Then, the target key corresponding to the location of the 1 had to be touched (i.e., the uppermost key if the digit was above the arrow, the lowermost key if it was below). When the 1 appeared alone, participants moved back to the home key and Response A ended here. When the 2 was also shown, participants were required to
proceed by touching the target key corresponding to the 2 (i.e., the key opposite to the first), and then move back to the home key. Response B was signaled by the presentation of a red or green square, which called for a brief release and retouch of the home key with the left or right index finger (or foot, as in Exp. 2).

![Diagram of the display](image)

Figure 4. Example of the displays used in Experiments 1 and 2.

The most significant feature of the design was that, although Response A was signaled before Response B, it was carried out only after Response B was completed (i.e., an ABBA design). Figure 4 shows that after a 350-ms fixation asterisk, Stimulus A was presented for 2 seconds. This rather long interval should give plenty of time to plan Response A, and the participants were strongly encouraged to do so. Following a 1-sec blank interval and another 350-ms fixation mark (this time the outline of the upcoming color stimulus), Stimulus B appeared for 200 ms. Importantly, Stimulus B should be responded to immediately by performing Response B, followed by the already planned Response A.

Such a design allows to study the impact of an already formed action plan—the plan to perform Response A—on the efficiency of planning another action—Response B. According to the Code-Occupation Hypothesis, planning Response B should be more difficult with feature overlap between the action plans for A and B than without a match. Therefore, I expected the reaction time (RT) for Response B to be higher with A-
B overlap, that is, if A and B were to be carried out with the same effector (or with different effectors on the same side) than without A-B overlap.

2.2 Experiment 1

The first experiment was based on the design and procedure described above and aimed at testing the Code-Occupation Hypothesis. Response-feature overlap was manipulated by having participants perform Response A and Response B either with the same hand (and on the same response panel) or with different hands (and on different response panels).

Some degree of response variation was introduced by asking the participants to touch either one or two target keys between home-key release and retouch, and by having them start sometimes with the upper and sometimes with the lower target-key position. In doing so, it was made sure that participants actually planned Response A on the spot, instead of just retrieving an overlearned action plan from memory. If, for an extreme example, the same Response A were repeated over and over again, participants would be very likely to learn the particular combination of response features and store this combination in long-term memory. Consequently, they would no longer need to temporally bind the corresponding action codes, hence form a dynamic action plan, but may just retrieve the overlearned action pattern as a static whole. If so, I would not expect temporal binding processes to play a major role. Indeed, the findings of Murthy and Fetz (1992) suggest that they actually do not, so that the effects of response overlap may no longer be observed. Therefore, I reasoned that dynamic binding in action planning, the process the Code-Occupation Hypothesis aims at, may only be expected with some degree of response variation, and this variation was realized by varying the number of response elements and the starting position. However, in Experiment 3 it will be shown that reducing the number of Response A alternatives to two does not change the outcome.
2.2.1 Method

Participants. Eighteen volunteers between 18 and 32 years of age were paid for their participation in a single session of about 60 min. Two more participants were excluded because they exceeded the error criterion of 20% overall.

Apparatus and stimuli. The experiment was controlled by a Hewlett Packard Vectra QS20 computer, interfaced with a digital I/O card, and attached to a 16-inch Eizo Flexscan monitor. Two wooden boards served as response panels, each supplied with three touch-sensitive metal plates of 1.0 x 1.0 cm with 0.5 cm in between. A white asterisk served as the first fixation mark that appeared at the center of the black screen. Stimulus A consisted of a white arrow appearing at screen center and the digit 1 plus, if applicable, the digit 2, appearing above and/or below the arrow. From a viewing distance of about 60 cm, each symbol cell measured about 0.3° in width and 0.4° in height. A white square outline, measuring 0.3° in width and 0.8° in height, was used as the second fixation mark, which also appeared at screen center. Red and green filled squares of the same size and location served as Stimulus B.

Design and procedure. There were 16 conditions resulting from the orthogonal variation of four within-participant factors with two levels each: side of Response A (left or right), side of Response B (left or right), complexity of Response A (1 or 2 target keys to be touched), and direction of first movement of Response A (back or forth). Participants worked through a practice block of 32 trials (16 conditions x 2 replications) and an experimental block of 320 error-free trials (16 conditions x 20 replications), both randomly ordered (with the constraint that conditions were not immediately repeated). Error trials were recorded and repeated at some random position in the remainder of the block. The two possible mappings of Response B on the color of Stimulus B were counterbalanced across participants.

Following an intertrial interval of 1,250 ms, the order and timing of events was exactly as shown in Figure 4. Participants were instructed to plan Response A as soon as Stimulus A appeared, to wait for Stimulus B and to respond to it as fast as possible, and then to carry out Response A immediately. In case of an error in response type or if the
sequence of responses was not completed within 5,000 ms, a brief error message was displayed before the next trial started.

For each response, three measures were taken: For Response B, the first to-be-emitted reaction, RT was measured from the onset of Stimulus B to the first release of the home key, and movement time (MT) was measured from the first release to the following retouch of the home key. RTs and MTs refer to error-free trials only, while percentages of errors (PEs) refer to lifts of the incorrect finger as proportion of all in-time responses. For Response A, the second reaction, interresponse times (IRTs) were measured from the end of Response B, hence the first retouch of the home key, to the second release of the home key, while MTs measured the interval from the second release to the final retouch of the home key. Error rates for Response A refer to all incorrect in-time responses following a correct Response B. All measures underwent ANOVAs for repeated measures and the significance criterion was set to $p < .05$.

2.2.2 Results

Response B. For each participant, mean RTs, mean MTs, and PEs were computed as a function of response-feature overlap (i.e., same vs. different sides of Response A and Response B; see Table 1 for group means). The RT analysis yielded a highly significant effect of overlap, $F(1,17) = 18.42, p < .001$, indicating that the latency of Response B was longer with overlap between Response B and the prepared Response A than with no overlap. The MTs and error rates showed similar effects that, however, failed to reach significance, $F(1,17) = 3.87, p < .07$, and $F(1,17) = 1.97, p < .20$, respectively.
Table 1:
Mean Reaction Times (RTs), Movement Times (MTs), Interresponse Times (IRTs), and Proportion of Errors (PEs) for Experiment 1 as a Function of Feature Overlap Between Response A and B.

<table>
<thead>
<tr>
<th></th>
<th>Response B</th>
<th></th>
<th></th>
<th>Response A</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT(508)</td>
<td>MT(132)</td>
<td>PE(2.0)</td>
<td>MT(138)</td>
<td>PE(5.7)</td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Overlap</td>
<td>475</td>
<td>126</td>
<td>1.4</td>
<td>153</td>
<td>576</td>
<td>6.2</td>
</tr>
</tbody>
</table>

Response A. For each participant, mean IRTs, mean MTs, and PEs were computed as a function of response-feature overlap (see Table 1 for group means). IRTs and PEs did not differ significantly. However, MTs were shorter with feature overlap between Response A and B than with no overlap, $F(1,17) = 7.18, p < .05$.

2.2.3 Discussion

The results clearly confirm the prediction from the Code-Occupation Hypothesis. The RTs of Response B are negatively affected by an overlap between the to-be-planned Response B and the already prepared Response A. Also MTs and PEs seem to be negatively affected by response-feature overlap, although the differences between the group means did not reach significance. At any rate, a speed-accuracy tradeoff can be ruled out because MTs and PEs went into the same direction as RTs. Rather, the pattern of results suggests that the planning of Response B was not always completed at the time the home key was released, so that some portion of the actually planning-related effect of overlap showed up in MTs. Taken together, the results obtained for Response B provide first, substantial support for the idea that action planning involves the temporal binding of response-feature codes.
Response A was also influenced by response-feature overlap. Interestingly, for Response A the influence was positive, and negative for Response B. This was only reflected in the MTs of Response A. The IRTs and the PEs of Response A went in the same direction, but did not reach significance. This result is in accordance with the effect of response repetition (cf. Bertelson, 1965; Rabbitt, 1968). In a choice reaction task paradigm, a response is faster if the same response was also selected in the previous trial. Thus, in the present experiment, Response A is facilitated because partially the same response features were part of the previously executed response. Thus, at the moment Response B has to be executed, all its action-feature codes are activated over some threshold. After the execution, all the codes are somewhat higher activated. If the representation of Response A overlaps with the representation of Response B, Response A benefits from this rest activation of Response B. It is important to point out that this interpretation is completely compatible with the Code-Occupation Hypothesis, which predicts a difficulty for the access to action codes only if these are already part of an action plan. At the moment Response A has to be executed, Plan B may already have been disintegrated, because it had no representational function anymore after its execution.

2.3 Experiment 2

Experiment 1 showed that action planning is more difficult when the action codes that need to be accessed are already part of another action plan. This effect will be called the cost of response- or action-feature overlap. In Experiment 1, overlap of action codes meant that Response A and B were both performed with the same hand. It is this aspect of Experiment 1 that can be taken up for an alternative explanation of the observed overlap cost: It might be argued that the longer latencies of Response B in the overlap conditions were the result of some kind of remapping an effector-specific motor program. According to this argument, planning Response A might result in the construction of a motor program that, in non-overlap conditions, just needs to be maintained until Response B is completed and then released. With overlap, however, a
conflict arises as Response B has to be performed with the same hand as Response A is prepared for. In order to solve this conflict, the original plan has to be remapped, that is, it needs to be reprogrammed to allow for the execution of Response B before A.

It is true that such a remapping account would not be too different from the binding approach proposed here. After all, both accounts agree that planning results in some kind of linkage that needs to be undone before a new link can be made. However, it is also obvious that an account of overlap costs in terms of effector-specific motor programs would be much less general and generalizable than the suggested binding approach. Therefore, it was important to test whether overlap costs are restricted to feature overlap between actions performed with the same effector, or whether effector-unspecific overlap costs are demonstrable.

Experiment 2 replicated Experiment 1 with a slight, but theoretically important, modification: While Response A was again to be made with the left or right index finger, Response B now had to be performed with the left or right foot. In the overlap condition, Response A and Response B were performed with hand and foot of the same body side, and in the non-overlap condition, they were performed with hand and foot of opposite sides.

From a temporal-binding perspective, this modification should not matter, or at least not so much. It is true that the feature overlap of two actions performed with the same effector should be greater than the overlap with two different effectors on the same body side. After all, a left-hand Response A and a left-hand Response B in Experiment 1 do not just share the feature of being made on the LEFT side or with the LEFT hand; they are also made with the same finger, hand, and arm; start from the same location; and so forth. Many of these features are not shared by responses given with the left hand and the left foot, which limits the degree of feature overlap and, thus, might well reduce overlap costs. Nevertheless, at least some degree of feature overlap should remain, so that substantial overlap costs should be demonstrable even with combinations of hand and foot responses.

From an effector-specific remapping account, however, there is no reason to assume that overlap costs show up if Response A and B are performed with different
effectors. If, for instance, Response A is prepared for the left foot, remapping would not
be necessary, irrespective of whether Response B is made with the left or the right
finger. Consequently, the disadvantage of overlap observed in Experiment 1 should
disappear in Experiment 2.

2.3.1 Method

Participants. Twenty-eight volunteers between 17 and 39 years of age were paid
for their participation. Three more participants were excluded because they exceeded the
error criterion of 20% overall.

Apparatus and stimuli. These were the same as in Experiment 1, with one
exception: For measuring foot responses two touch-sensitive plates of 20 x 20 cm,
mounted on wooden boards of 25 x 40 cm and placed under the participant’s feet, were
connected to the I/O interface card.

Design and procedure. The procedure was exactly as in Experiment 1, except
that Response B was made with the left or right foot, as indicated by the color of
Stimulus B. Participants were asked to lift their foot briefly from the touch-sensitive
plate and retouch it without lifting their heel.

2.3.2 Results

Response B. Mean RTs, MTs, IRTs, and PEs were computed as in Experiment 1
(see Table 2). The effect of response-feature overlap was significant for the RTs,
$F(1,27) = 5.92, p < .05$, which were prolonged if Response B, the foot reaction, was on
the same side as Response A, the prepared hand reaction. Likewise, MTs were
significantly longer and errors were more frequent with response-feature overlap than
with no overlap, $F(1,27) = 12.83, p < .001$, and $F(1,27) = 10.73, p < .01$. 

Testing the Code-Occupation Hypothesis
Table 2:
Mean Reaction Times (RTs), Movement Times (MTs), Interresponse Times (IRTs), and Proportion of Errors (PEs) for Experiment 2 as a Function of Feature Overlap Between Response A and B.

<table>
<thead>
<tr>
<th></th>
<th>Response B</th>
<th></th>
<th>Response A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT</td>
<td>MT</td>
<td>PE</td>
</tr>
<tr>
<td>Overlap</td>
<td>565</td>
<td>161</td>
<td>6.1</td>
</tr>
<tr>
<td>No Overlap</td>
<td>553</td>
<td>158</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Response A. IRTs and PEs were virtually identical in overlap and no-overlap conditions and, thus, did not differ significantly. However, the MTs were significantly shorter with overlap than with no overlap, $F(1,27) = 5.09, p < .05$.

2.3.3 Discussion

The purpose of Experiment 2 was to contrast the Code-Occupation Hypothesis, which predicted response-feature overlap costs, with the remapping hypothesis, which predicted no such costs. Although Response A and B were always performed with different effectors, clear evidence of performance decrements with feature overlap was found in the RTs, MTs, and in the PEs of Response B. This outcome supports the Code-Occupation Hypothesis: Although, as expected, the size of the overlap effect is reduced as compared to Experiment 1, its mere occurrence suggests that overlap costs have to do with correspondence or noncorrespondence of response or action features, but less with the identity or nonidentity of the effectors used for Response A and B.

The results of Response A are similar to those of Experiment 1. In the present experiment, the MTs of Response A were significantly shorter in case of response-feature overlap and the RTs and PEs went in the same direction. The observation that this kind of repetition benefit is found even if different effectors are used for Response
A and B rules out possible accounts in terms of anatomical identity or effector activation. Thus, response priming seems to occur if the primed response overlaps with the previous response only in an abstract fashion.

2.4 Experiment 3

Experiment 1 and 2 showed that Task B performance is worse in case of response-feature overlap with Task A, this strengthening the Code-Occupation Hypothesis. Nevertheless, neither Experiment 1 nor Experiment 2 proved that Response A is really planned before Task B. The results of the first two experiments gave indirect indications that the important assumption that Response A is really planned before participants start with Task B is indeed true. First, it is implausible that Response A can be correctly performed most of the time without having prepared it before Task B. Second, the mean IRTs between Response B and A were all under 156 ms. If Response A had to be prepared after Response B, latencies over 250 ms, which is the minimal time for choice-reaction tasks, should be expected. However, one cannot be sure exactly what IRT level would be sufficiently low to indicate true and complete action planning. Perhaps, some participants, or all participants sometimes, merely memorized Stimulus A until Response A had to be emitted and only then translated the stimulus into an action plan. If so, some portion of the obtained overlap costs were due to an interaction between Response B and Stimulus A (or some memory code of it), but not to the proposed interplay of Response B and the plan of Response A.

The purpose of Experiment 3 was to test somewhat more directly whether it is really the prepared Response A that interferes with Response B, by manipulating the probability that Response A is planned before Response B is prepared and executed. To achieve this goal, participants ran through two different sessions: In a “planning” session, participants were encouraged to plan Response A before the onset of Stimulus B and, as in Experiments 1 and 2, there was plenty of time to do that. Moreover, a less memorizable format of Stimulus A was used (the arrows were replaced by the letters X and O), to further encourage immediate planning. In a “no-planning” session, run on a
different day, the participants were asked to plan Response A only after the execution of Response B instead. Because in this session the stimulus-onset asynchrony between Stimulus A and Stimulus B was just 100 ms and Stimulus A stayed on the screen until Response B was completed, it was not only practically impossible to plan Response A before Response B, but unnecessary as well. Obviously, even such an experimental design cannot assure that planning of Response A was always perfect in the planning session and completely absent in the no-planning session. Nevertheless, it does make sense to assume that the degree of planning would be considerably higher in the former than in the latter case. If so, and if overlap costs are really produced by the interplay between action plans, overlap costs should be expected to appear in the planning session only, but not in the no-planning session.

2.4.1 Method

Participants. Sixteen volunteers between 17 and 39 years of age were paid to participate in two sessions of about 15 minutes each, run on different days. Three more participants were excluded because they exceeded the error criterion of 20% overall.

Apparatus and stimuli. The same apparatus was used as in Experiment 1. Task A was signaled by a black uppercase X or O, positioned in a white, filled square. From a viewing distance of about 60 cm, the letter measured about 0.3° in width and 0.4° in height and the square subtended 1.0° in width and 2.2° in height. To signal Response B, the square turned green or red. The fixation point and error messages were as in Experiment 1.

Design and procedure. There were eight conditions, resulting from the orthogonal variation of three within-participant factors with two levels each: side of Response A (left or right), side of Response B (left or right), planning of Response A (encouraged = “planning” or discouraged = “no planning”). In either session, participants worked through a practice block of 20 trials (4 conditions x 5 replications) and an experimental block of 80 error-free trials (4 conditions x 20 replications). Error trials were recorded and repeated at some random position in the remainder of the block.
In each session the general method was as in Experiments 1 and 2, that is, Stimulus A was followed by Stimulus B and then Response B was performed, followed by Response A (i.e., the ABBA design). Stimulus A, the letter, indicated whether Response A was to be performed with the index finger of the left or the right hand. Response A required participants to release the home key, touch the upper target key, touch the home key again, touch the lower target key, and then return to the home key. That is, the complexity and the direction of the first movement of Response A did not vary. Stimulus B and Response B were the same as in Experiment 1, except that Stimulus B was larger. The two possible mappings of Response A on the identity of Stimulus A, the two possible mappings of Response B on the color of Stimulus B, and the two possible orders of the sessions were all counterbalanced across participants.

In the planning session, each trial started with a white square at the center of the screen. After 350 ms, Stimulus A (i.e., a black uppercase X or O) was presented for 1,000 ms at the center of the white square and then disappeared. 1,000 ms later, the square turned green or red, indicating the hand for Response B. After another 200 ms the square turned white again and completely disappeared as soon as the participant began performing Response A. To encourage the planning of Response A, participants were instructed to perform Response A as quickly and as accurately as possible after completion of Response B. After each ten error-free trials, visual feedback was given about the mean RT and the number of errors for Response A. An integrated index of performance in Task A was also shown (= mean IRT + number of errors x 20), including information about improvements relative to the preceding block. The remaining procedural details were as in Experiment 1.

In the no-planning session, each trial also started with the onset of a white square at screen center. After 1,350 ms, Stimulus A appeared within the square, where it stayed until Response A began. 100 ms after the onset of Stimulus A, the square signaled Response B by turning green or red for 200 ms. To discourage the planning of Response A, participants were asked to react as quickly and accurately as possible to the color, that is, to Stimulus B. Moreover, in contrast to the planning session, the feedback referred to Response B, not A. The remaining procedure was as in the planning session.
2.4.2 Results

Response B. Mean RTs, MTs, IRTs, and PEs were computed as a function of response-feature overlap and planning (see Table 3 for group means). For the RTs, there was only a highly significant interaction, \( F(1,15) = 21.46, p < .001 \), due to that overlap influenced RTs negatively in the planning session, but positively in the no-planning session. Separate comparisons revealed that both overlap costs in the planning session and overlap benefits in the no-planning session were significant. For the MTs no significant effects were obtained. The analysis of the PEs revealed one highly significant main effect: Less errors were made with overlap than with no overlap, \( F(1,15) = 25.03, p < .001 \).

The fact that the factors overlap and planning did not interact in the PEs might indicate some kind of accuracy tradeoff. To check this possibility a product-moment correlation was computed between overlap-effect sizes in RTs and in PE, separately for each planning session. However, both correlations were far from significance and, if anything, positive (plan: \( r = .24, p > .4 \); no plan: \( r = .03, p > .9 \)), which rules out a tradeoff account.

### Table 3:

Mean Reaction Times (RTs), Movement Times (MTs), Interresponse Times (IRTs), and Proportion of Errors (PEs) for Experiment 3 as a Function of Planning and Feature Overlap Between Response A and B.

<table>
<thead>
<tr>
<th>Overlap Type</th>
<th>Response A</th>
<th>Response B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT</td>
<td>MT</td>
</tr>
<tr>
<td>Planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td>503</td>
<td>123</td>
</tr>
<tr>
<td>No Overlap</td>
<td>474</td>
<td>126</td>
</tr>
<tr>
<td>No Planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td>560</td>
<td>120</td>
</tr>
<tr>
<td>No Overlap</td>
<td>595</td>
<td>125</td>
</tr>
</tbody>
</table>
Response A. IRTs produced only a highly significant effect of planning, $F(1,15) = 15.75, p < .001$, with faster initiation of Response A in the planning session than in the no-planning session. The main effect of response-feature overlap, indicating shorter RTs in the overlap condition, only approached the significance criterion ($p < .09$). For the MTs, the main effect of overlap was significant, $F(1,15) = 7.55, p < .05$, indicating that Response A was completed more quickly with response-feature overlap than without. The PE analysis yielded no significant effects.

2.4.3 Discussion

The purpose of this experiment was to determine the role of action planning in producing the response-feature overlap costs observed in Experiments 1 and 2. To do so, the preparedness of Response A before Task B was manipulated. Participants were encouraged to plan Response A before performing Response B in the planning session, but were discouraged to do so in the no-planning session. The manipulation worked, indeed: Response A was initiated much more quickly in the planning session than in the no-planning session, where the IRTs approached a level one would rather expect in a binary-choice task. This shows that the participants were able to prepare Response A to a much higher degree under conditions that were thought to favor planning processes. The observation that the accuracy of Response B was negatively affected by Response-A planning goes in the same direction. However, as this may also be an effect of forward masking, due to the shortened interval between Stimulus A and B in the no-planning session, this interpretation should be treated carefully. At any rate, there is reason to assume that Response A was planned at least more often or with greater efficiency in the planning session than in the no-planning session.

If it really was the planning of Response A prior to B that hampered performing Response B in the overlap conditions of Experiments 1 and 2, similar overlap costs should be expected in the planning session of the present experiment, but not in the no-planning session. And this is exactly what the results show: Response B was initiated
less quickly with response-feature overlap than with no overlap when participants were encouraged to plan Response A, while even the opposite pattern was obtained when participants were discouraged from planning. Obviously, it is planning that matters, not just perceiving or memorizing Stimulus A, or being prepared to respond to the latter.

Nevertheless, the positive overlap effect indicated that perceiving Stimulus A was not without any effect in the no-planning condition. Obviously, Stimulus A was not just presented too briefly before Stimulus B to be encoded at all or to be translated into its corresponding response. Note that letters instead of arrows were used as Stimulus A, hence stimuli with no intrinsic spatial meaning, so that overlap effects can only be attributed to an interaction between (codes of) Response A and B. That is, Response A must have been activated before the completion of Response B even in the no-planning condition. Otherwise the relationship between Response A and B could not have affected performance of B. This means that response (code) activation and response planning are different processes, a conclusion that fits well with the introductory considerations and with recent observations from other dual-task studies of Hommel.

Hommel (in press b) found that the response of a secondary task was activated before completion of the primary task in a standard dual-task paradigm. Nonetheless, there were dual-task costs, that is, there was an increase in the latency of the secondary response at shorter SOAs between both imperative stimuli. Thus, although there is a kind of bottleneck, S-R translation can precede the bottleneck, which is contrary to the assumption that the bottleneck reflects serial processing (e.g., Pashler, 1994; Welford, 1952). For the present experiment, this implies that perceiving Stimulus A led to the automatic application of the memorized S-R mapping rule and, thus, to the activation of the corresponding response codes. In the planning session, this was immediately followed by the planning of Response A, hence the binding of those codes, with the resulting decrease in their accessibility from overlapping action plans. In the no-planning session, however, no binding followed before Response B was completed. Therefore, the codes were already activated, but not yet occupied, so that planning an action with overlapping codes was facilitated.
2.5 Experiment 4

The previous experiments reported feature-overlap costs when an actual to-be-planned action shared a feature with an already prepared action plan held in memory. According to the Code-Occupation Hypothesis, this effect depends on the preparedness of Reaction A. Because the degree of readiness of Plan A depends on the SOA between Stimulus A and B, this SOA was manipulated. At very short SOAs, a facilitatory effect of response overlap instead of an interfering effect was found in Experiment 3.

The originally proposed Code-Occupation Hypothesis does not expect any effects of a delay between Responses B and A. However, if Response A is indeed positively influenced by the remaining activity in the overlapping response feature, the delay time between Response B and A should affect Response A. After a delay between Responses B and A, such a facilitation effect should not be found anymore because the remaining activity decays automatically (cf. Hommel, 1994). This was tested by manipulating the delay between Responses B and A. It was expected that only if the delay between Response A and B would be short, Response A would be influenced positively by an overlap with Response B.

According to the Code-Occupation Hypothesis, the variation of the ROA (Response Onset Asynchrony) between Response B and Response A should not influence the effect of feature-overlap costs itself. Still, it could be argued that the costs of feature overlap depend on the fact that Responses B and A are fused into one single motor program. Of course, this is not a valid argument, because such a fusion would expect a facilitatory effect of response-feature overlap: If the same response code has to be used twice for the same program, it only has to be retrieved once, which saves time (Rosenbaum, Inhoff, & Gordon, 1984). Thus, although there is no need to prove that the Code-Occupation Hypothesis does not depend on the fusion of two responses, the theory would show its strength if the Code-Occupation Hypothesis were also be found when Responses A and B are clearly separated. This separation was realized by an unpredictably long delay between Responses B and A.
Altogether, it was predicted that the costs of feature-overlap reflected in Response B are independent of the delay between Responses B and A. The benefit of feature overlap in Response A should only be found if the delay between Responses B and A is short.

2.5.1 Method

Participants. Twenty volunteers aged 20 to 30 years were paid to participate in a single session of about 30 min. One more participant was excluded because he exceeded the error criterion of 20% overall.

Apparatus and stimuli. The same equipment was used as in Experiment 1. Stimulus A was a letter, \( \text{X} \) or \( \text{O} \), measuring about 0.3° in width and 0.4° in height. The letter was presented at screen center. Stimulus B was the same as in Experiment 1. A go-signal for Response A followed Response B after a short or a long delay. The delay time was randomly chosen in each trial. The short delay lasted between 200 to 300 ms (mean 250 ms). The long delay lasted 1200 to 1300 ms with a mean of 1250 ms. The go-signal was beep-tone lasting of 50 ms, coming from two loudspeakers, which were positioned left and right of the computer screen.

Design and procedure. There were eight conditions resulting from the orthogonal variation of three withing-participant factors with two levels each: Side of Response A (left or right), side of Response B (left or right), and delay time (short or long). Participants worked through a practice block of 16 trial (8 conditions x 2 replications) and an experimental block of 160 error-free trials, both randomly ordered (with the constraint that conditions were not repeated immediately). Error trials were recorded and repeated at some random position in the remainder of the block. The two possible mappings of Response B to the color of Stimulus B were counterbalanced across participants.

The timing of events was the same as in Experiment 1, except for the delay time between Response B and the go signal for Response A. After an error-free Response B and the delay time, the go-signal followed, indicating that Response A should be
performed. Further, Response B had to be performed within 1,500 ms after the onset of Stimulus B, but not within the first 50 ms. Response A had to be performed within 3,500 ms after the presentation of the go-signal.

If participants released the home-key during the delay time, the trial ended and an error message was displayed.

2.5.2 Results

Response B. For each participant, mean RTs, mean MTs, and PEs were computed as a function of response-feature overlap (i.e., same vs. different sides of Stimulus A and Response B). The RT analysis yielded a significant effect of overlap, $F(1,19) = 6.46, p < .05$, indicating that the latency of Response B was longer with overlap between Responses B and A. Neither the MT nor the PE analyses of Response B yielded significant effects.

Response A. For each participant, mean RTs, mean MTs, and PEs were computed as a function of response-feature overlap (i.e., same vs. different sides of Stimulus A and Response B). The RT analysis yielded a significant effect of overlap, $F(1,19) = 8.12, p < .01$, indicating that the latency of Response A was shorter with overlap between Response B and Response A. There was a significant effect of delay, $F(1,19) = 48.62, p < .001$, indicating that the latency of Response A was shorter when the delay between Responses B and A was longer. There was a significant interaction between delay and overlap, $F(1,19) = 10.11, p < .005$, indicating that feature-overlap benefit was significant in the short-delay conditions only. A separate analysis of the short delay trails showed that in the short delay conditions, IRTs were shorter in the overlap condition than in the no-overlap condition, $F(1,19) = 16.17, p < .001$.

The MT analysis revealed a highly significant effect of delay, $F(1,19) = 33.40, p < .001$, showing that MTs in the short delay condition were shorter that in the long-delay condition.

The PE analysis yielded no significant effects.
Table 4:
Mean Reaction Times (RTs), Movement Times (MTs), and Proportion of Errors (PEs) for Experiment 4 as a Function of Feature Overlap Between Stimulus A and Response B and Delay of Response A.

<table>
<thead>
<tr>
<th></th>
<th>Response B</th>
<th>Response A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT</td>
<td>MT</td>
</tr>
<tr>
<td>short delay</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td>410</td>
<td>110</td>
</tr>
<tr>
<td>No Overlap</td>
<td>399</td>
<td>111</td>
</tr>
<tr>
<td>long delay</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td>412</td>
<td>110</td>
</tr>
<tr>
<td>No Overlap</td>
<td>394</td>
<td>112</td>
</tr>
</tbody>
</table>

2.5.3 Discussion

The results suggest that the remaining activity of the response-feature codes of Response B facilitate the initiation of an overlapping Response A. Thus, when the delay is long, there is no remaining activity anymore, and Response A does not benefit from an overlap with Response B.

The delay is responsible for two other effects in Response A. First, the initiation time (measured from the go-signal) of Response A is shorter after a longer delay. This is not very surprising, because participants could use the delay to better prepare Response A. Furthermore, the MT is longer in the long-delay condition. This suggests a tradeoff between the RTs and MTs: The participants had faster RTs in the long-delay condition, but this advantage was lost through longer MTs. Even if this post hoc explanation is true, it has no consequences for the hypothesis, because the MTs do not interact with the essential independent variable, that is the overlap.
The results clearly show that feature-overlap costs are also found when Responses A and B are temporally separated. Thus, even when Response B and Response A are unlikely to be part of the same motor program, Response B is clearly affected by feature overlap with Response A. This result is a rejection of the argument that feature-overlap costs result from the fact that Response A and B are part of a single motor program.

2.6 General Discussion

The main purpose of the experiments reported in this chapter was to find out whether action planning is associated with a feature-binding problem. First of all, I introduced the new ABBA Paradigm to test the Code-Occupation Hypothesis. As predicted by the Code-Occupation Hypothesis, Experiment 1 showed that planning an action is impaired if it shares features with another, already planned action plan held in memory. The benefit of overlap on the MT of Response A is interpreted as the result of the remaining activity in the overlapping code. This is not incompatible with the Code-Occupation Hypothesis and in accordance with the effect of response repetition.

Experiment 2 indicated that feature-overlap costs can be found even if the overlapping features do not refer to the same effector, hence when overlap is fairly abstract. Experiment 3 demonstrated in a more direct way that feature-overlap costs depend on intentional planning. If an action overlaps with a merely cued, but not intentionally prepared action plan, it is no longer interfered with but rather facilitated. Experiment 4 showed that after the execution of an action plan, the temporal bindings are disintegrated, but that the remaining activity of the overlapping code still influences the following action, under the condition that the delay between the first and second action is short enough. In the following, I will discuss some more general theoretical implications of the results.
2.6.1 Abstract Coding of Action Plans

An important question about action planning concerns the level of abstraction on which the basic components of action plans are defined. According to Keele (1968), a motor program is a structured set of muscle commands, hence a relatively concrete and peripheral entity. Later approaches, such as Rosenbaum’s (1985, 1987), assumed that programs are built from abstract codes, hence from movement parameters that need not directly map on specific muscle commands. The results of Experiment 1 to 4 support the assumption that abstract codes are involved in action planning. In Experiment 2 it was shown that planning a left- or right-hand action hampers foot responses on the same body side. That is, body side or egocentric location of the effectors must have been coded in the corresponding action plans independent of the effector’s identity. This clearly rules out the idea that planning occurs in terms of muscle commands.

The findings fit well with other observations that also point to the abstract nature of action planning. For instance, Rosenbaum and colleagues have repeatedly shown that the time it takes to plan manual actions decreases with the amount of advance information the actor receives about abstract, outcome-related action features (see Rosenbaum, 1987, for an overview). That is, action plans seem to take abstract parameters. In the same vein, Ulrich, Moore, and Osman (1993) found that actions can be planned up to the level of a measurable lateralized readiness potential (LRP) even if the actual effector is not yet known. In their experiment, participants were given advance information about the hand (left or right) of an upcoming manual action, but not about the finger to be moved, so that the information was too abstract to allow for preparing a specific set of muscles. Yet, partial information about the hand significantly speeded up LRP onset, suggesting that the LRP reflects the implementation of abstract response parameters.
2.6.2 A Two-Process Approach to Action Planning

According to the common view, the gap between stimulus processing and action control is bridged by a single process, usually called “stimulus-response translation” or “response selection” (e.g., Meyer & Kieras, 1997; Sanders, 1990; Teichner & Krebs, 1974; Theios, 1975). The basic assumption underlying most current models of human information processing is that after the stimulus is analyzed to at least some degree, some kind of selection process is called for, or some automatic translation device is activated, which then makes sure that the correct motor program is launched.

However, this single-step translation view has been challenged by the outcome of the already mentioned study of Hommel (in press b), who worked with a standard dual-task design involving a primary manual keypressing task and a secondary vocal color-naming task. On the one hand, secondary-task performance decreased with the time interval between primary and secondary stimulus, hence it decreased with increasing temporal task overlap. Of course, such an outcome is hardly surprising and merely replicates the well-known finding that people are unable to select two responses at the same time (see Pashler, 1994, for a recent overview). On the other hand, however, the compatibility between the secondary response (the word “red” or “green”) and the primary stimulus (a red or green color patch) strongly affected primary-task performance. This shows that the secondary response was activated before the primary task was completed and, thus, suggests that S-R translation does not provide a bottleneck in dual-task performance. To account for this pattern of results, Hommel (in press b) suggested to distinguish between response activation, which typically results from stimulus-response translation, and eventual response selection, which involves recruiting those already activated response codes that belong to the same response.

The present findings fully support and extend this two-process view of response preparation. In all those conditions where participants were encouraged to plan Response A, a substantial impairment of compatible B responses was found, an effect attributed to the binding of response features in the process of planning Response A. If planning was neither encouraged nor likely, as in the no-plan condition of Experiment 3,
a positive compatibility effect was found. This is exactly what one would predict if S-R translation and the resulting activation of response features were automatic, as Hommel (in press b) assumes. Therefore, it is concluded that action planning consists of two processes or stages: an automatic one responsible for activating those action-feature codes that are signaled by the stimulus, and another, more effortful one that binds the activated features. That is, constructing an action plan requires not just specifying the features of the intended action, but also temporarily binding them to form an integrated whole.

2.6.3 Feature Binding as a General Principle in Cognitive Representation

Altogether, the costs of feature overlap seem to depend on the consequences of a binding process. Planning an action involves the integration of several action-feature codes that specify the intended action. If action-feature codes are integrated, it is harder to integrate them in different action plans. Although the finding of overlap costs in the action domain is new, it fits well with a number of previous observations of similar effects in the perceptual domain. Therefore, I will propose a more general form of the Code-Occupation Hypothesis, which will be the central theme of the next chapter.
Chapter 3

The Generalized Code-Occupation Hypothesis

This chapter extends the theoretical framework expressed in the previous chapters. The extension is based on the common-coding approach, which states that action and perception share one representational domain (Prinz, 1992, 1997a, 1997b, 1997c). This means that action representations and object representations rely on the same set of representational building blocks (also called codes). According to this approach, feature codes can represent both action parameters and features of perceptual events. In section 3.1, I will discuss this approach and in 3.2, I will argue that the basic assumptions of the common coding approach imply that if a feature code is occupied, it can cause overlap costs in perceptual processes as well as in action planning. This assumption is the core of the Generalized Code-Occupation Hypothesis. In section 3.3, it will be argued that there already is empirical evidence available for the Generalized Code-Occupation Hypothesis. At the end of the chapter, it will be held that this hypothesis explains both normal and inverted compatibility effects.

3.1 The Common-Coding Approach

The common-coding approach deals with the processes underlying goal-directed action (Prinz, 1997a, 1997b, 1997c). The key characteristic is the assumption that action representations contain not only information about movements but also about the goal to be reached, and that the representation of this goal state is coded in terms of sensory events. This approach rests upon the work of Lotze (1852) and James (1890). These psychologists assumed that there are no fundamental differences between the representational content of perception and action, because the environmental changes, caused by actions, are perceived like any other perceivable event. The only difference between perceptual representations and action is that perceptual representations are caused by events in the environment, whereas action representations rely on intended future events in the environment. For example, the action representation of switching on
the light in a dark room relies on the known imagination of the same room in an illuminated state. In other words, action planning is the conception of future sensory states. Lotze and James assumed that the imagination of a future situation would automatically lead to the required movements.

According to Prinz (1997c) the emergence of action concepts is a learning process. In this process, it is learned which movements eventually realize an intended situation: Sensory events are associated with certain movements codes. One can easily imagine how this process takes place. If we encounter a new object (e.g., a kitchen device) we make random movements towards the object in order to find out what will happen. At first, we press some keys at random and soon we have found out which keypresses or other movements towards the new device lead to which effects. Thus, contingencies between intended effects and movements are learned. These associations between goal concepts, built from sensory codes, are connected to movement concepts. Action codes are viewed as such connections between sensory codes and movement codes.

The above described common-coding approach rests on two notions. First, event codes on the common-coding level represent the close relationship between objects and the organism. Thus, objects are not only coded in terms of sensory information like color and form, but also in terms of how they interact with the behaving person. One could imagine that an event code is a very complex network of associations between sensory and behavioral information which evolved over a long time of interactions between an object and organism. Connectionist models show how such codes emerge through the interaction between the sensory units and the motor units. In these models, internal representations are a complex combination of incoming sensory information and outgoing motor information. Because behavioral aspects can again be fed into the sensory units via recurrent connections, the system learns the relation between motor behavior and the to-be-expected outcomes on the sensory units (cf. Jordan, 1986).

A second important notion of the common-coding approach is the action-effect principle (Prinz, 1997c): Actions are planned and controlled in terms of their effects. This means, that action representations are not only coded in terms of pure motor codes,
that is, in terms of activation levels related to muscle movements, but that they are also
coded as associated sensory and movement codes.

The common-coding approach can be seen as a reaction to the simplified
conception of the processes between sensory events and responses, based on the widely
accepted idea of separate coding. The separate-coding approach rests on the work of
Donders (1868), who distinguished 12 stages of processing between stimulus reception
and response execution. The separate-coding approach is, like the common-coding
approach, an idea concerning the relationship between sensory and motor events,
posing that the contents of actions and perceptual representations are incommensurate.
A translation mechanism maps sensory codes onto motor codes (Massaro, 1990;
Sanders, 1980; Welford, 1968). However, this approach toward information processing
is far from complete, because humans are able to decide whether they react to a stimulus
and they are able to decide how to react to a certain stimulus category. Thus, in order to
describe all aspects of action planning, one need not only consider the cognitive
processes taking place between the presentation of the stimulus and the execution of a
response, but also the processes of intention formation: Before perceiving imperative
stimuli, humans form intentions that determine the information processing after the
stimulus presentation, because in most cases humans do not react in a reflex-like way to
stimuli. The common-coding framework especially stresses the structural aspects of
intentions, described as networks from sensory and movement information.

The common-coding approach implies that there are certain abstract codes that
can represent both action features as well as perceptual features, like position, direction,
movement, or speed. This implication follows from the simple idea that actions are
coded as action effects. For instance, if I intend to press a key on the left keyboard, the
representation can best be viewed as an imagination of the results such a keypress would
have, for example certain tactile feedback or an audible click.
3.2 From Common Coding to the Generalized Code-Occupation Hypothesis

The Code-Occupation Hypothesis was initially thought of as describing regularities in the domain of action planning alone. According to the common-coding approach, however, the Code-Occupation Hypothesis should be very closely related with the representation of sensory events, because there exists no real separation between action codes and sensory codes, as has been described above.

The Generalized Code-Occupation Hypothesis is the combination of the common-coding principle and the Code-Occupation Hypothesis and can be expressed as follows: *The construction of a cognitive representation will be impaired when part of its representational building blocks are part of an already formed representation.*

In other words, the function of a representation does not matter for the costs of overlap between an already constructed representation and a to-be-constructed representation. The Code-Occupation Hypothesis of the previous chapters was about interference in the action domain, therefore I will now refer to the costs of feature overlap in chapter 2 as R-R overlap costs. Additional to the R-R overlap costs predicted by the Code-Occupation Hypothesis, three more types of overlap costs are predicted:

1) S-R overlap costs, that is, an impairment of action planning because the to-be-constructed action plan partially overlaps with already constructed object representation(s).

2) R-S overlap costs, that is, an impairment of object recognition because the to-be-constructed object representation partially overlaps with already constructed action plan(s).

3) S-S overlap costs, that is, an impairment of recognition because the to-be-constructed object representation partially overlaps with already constructed object representation(s).
3.3 Evidence for the Generalized Code-Occupation Hypothesis

Each of the four types of overlap cost have been observed, although with
different paradigms and by different research groups. Evidence for R-R overlap costs
has been presented in chapter 2, so I will restrict myself to the other three types.

Evidence for S-R overlap costs is scarce. Hommel (in press a) reported that a
partial match between a particular combination of stimulus and response features
produces worse performance than a complete mismatch (see chapter 1 for a more
detailed discussion). This suggests that stimulus and response features are integrated
into a common representation, so that the features involved are temporarily unavailable
for the formation of other representations.

Evidence for R-S overlap costs comes from several studies. Recent
findings of Müseler and colleagues demonstrate that feature overlap between a manual
action about to be performed and a to-be-identified visual stimulus impairs perceiving
the latter. Müseler and Hommel (1997a, 1997b) showed that identifying or detecting an
arrow pointing to the left or right is more difficult if that stimulus appears at about the
same time a spatially compatible response is made. Wühr and Müseler (1997) observed
that this kind of “blindness” to response-compatible stimuli sets in as early as 2 seconds
before the planned manual response is actually emitted, revealing that it is not the
execution, but the planning of a feature-overlapping action that hampers perception.

Evidence for the S-S overlap costs has been observed in negative-priming tasks
(e.g., Allport et al., 1985) and with preview designs (Hommel, in press a; Kahneman et
al., 1992). As already discussed in chapter 1, the evidence available so far shows that a
partial match between the features of succeeding stimuli produces worse performance
than a complete mismatch. In other words, feature overlap between two event
representations formed in close temporal succession impairs performance as compared
to no overlap.

The explanations of the mentioned effects of feature-overlap costs have in
common that feature binding is conceived as the limiting factor: It is assumed that a
binding conflict arises when a feature has to be bound to a feature configuration other
than the one it actually belongs to. The similarity between costs of feature overlap in perception and in action planning suggests that both the binding of action features for action planning and the binding of perceptual features for object representation are similar.

3.4 Facilitatory and Interfering Effects in Compatibility Paradigms

An overlap between two events may lead to facilitated performance or, in other cases, to impaired performance. The Simon effect is the prototypical example of performance facilitations in case of an overlap between stimulus and response (Simon & Rudell, 1967). In experiments using the Simon paradigm, participants usually react to a stimulus, whose relevant and irrelevant features are varied. For example, participants react to the color of a stimulus whereas its position irrelevant. When position and response correspond, performance is enhanced.

The Simon effect is regarded as supporting the common-coding approach, because response categories seem to be activated automatically by corresponding stimuli. As the imperative stimulus and response have overlapping dimensions, their building blocks are partially the same. That is, the to-be-programmed response is already partially selected just because the building blocks are already activated for the stimulus representation.

Nevertheless, performance is not always facilitated under the condition of S-R overlap. The studies discussed in the previous section are good examples of situations under which S-R overlap leads to worse performance. At first sight, these studies can be viewed as problematic for the common-coding approach. Why should there be a problem of constructing responses when they already partially exist in the form of already established stimuli representations? The answer is the following: Representations do not only rely on activity in a set of codes, but also on the relationships between the codes. This answer brings us back to the binding problem discussed in chapter 1. If activation were the only distinguishing dimension among codes, the cognitive system could not distinguish different groups of codes with the
same level of activation, that is, coactivation would lead to unsurmountable problems as is illustrated in Figure 1.

Altogether, the common-coding theory is confronted with the fact that overlapping feature dimensions sometimes lead to facilitation and sometimes to interference. This asks for a necessary extension of the common-coding theory. I suggest that the introduction of temporal bindings between codes is the solution. In fact, the two-stage model of the construction of action plans as postulated in chapter 2 can be applied to the construction of event representations on the common-coding level. According to that model, the integration of codes into a representation cannot take place until the codes have been activated before. In other words, codes are first activated and then integrated. Feature-overlap costs are only expected when codes are integrated. Feature-overlap advantages are expected when codes are activated but not integrated. From these two simple principles, exact predictions about facilitation and interference can be derived. Suppose there are two overlapping representations, A and B. The construction of Representation B is facilitated under the following conditions; 1) The integration of A is not completed; 2) A and B are exactly the same. Under all other conditions interference is expected.

From this follows that there is an interaction between the overlap of two events and their timing. The assumptions are in accordance with the temporal relationships found between stimulus and response in experiments with a Simon paradigm. When RT is short, there is a clear facilitatory effect which becomes smaller with higher RTs (Hommel, 1993). In some experiments, an interfering effect of overlap is found in the slow RTs (Hommel, 1993). And when participants had to delay their response, the irrelevant stimulus position did not positively influence the compatible response if the delay was 300 ms or longer (Simon, Acosta, Mewaldt, & Speidel, 1976). This is exactly what the two-stage model of action planning predicts. When identity has been coded but the position of the imperative stimulus is not integrated together with the identity, the position code has been activated but not integrated, this resulting in response facilitation in case of overlap. Conversely, with longer RTs the code has been integrated and is
therefore not available for action planning. Nevertheless, this is a post-hoc interpretation of the data, and further research is surely necessary.

One could think of the following experimental setup. Suppose that stimuli have two dimensions like in a standard Simon experiment, but that these two dimensions are both relevant. If participants integrate stimulus color with the position because this conjunction is necessary for response selection, a reversed Simon effect should be expected, because the code necessary for the reaction is occupied by the stimulus representation.

In the following chapter, I will report new evidence for the Generalized Code-Occupation Hypothesis: S-R compatibility effects will be reversed if the stimulus features are integrated before the response selection starts.
Chapter 4

Testing the Generalized Code-Occupation Hypothesis

In the previous chapter, four predictions have been derived from the Generalized Code-Occupation Hypothesis and I have discussed support for each prediction. Nevertheless, only scarce and indirect evidence exists for the S-R overlap costs prediction, which states that action planning will be impaired if the to-be-planned action shares features with an already constructed representation of an object. Therefore, the experiments reported in this chapter are restricted to testing the S-R overlap-costs prediction.

In 4.1, I will describe how the experimental paradigm of chapter 2 was modified accordingly. Then, three experiments will be reported that corroborate the S-R overlap prediction. Finally, the theoretical implications of the results will be discussed.

4.1 Looking for S-R Overlap Costs with the ABBA Paradigm

The ABBA paradigm was also used to test the S-R overlap-costs prediction of the Generalized Code-Occupation Hypothesis. Instead of manipulating the overlap between Responses A and B, the overlap between Stimulus A and Response B was manipulated. To do this, not the prepared Response A, but the object “Stimulus A” was held in memory while Response B was planned (see Fig. 5). Stimulus A and Response B either shared a feature or not.

The sequence of events in the present experiments was different from the experiments in chapter 2: Reaction A could not be prepared after the presentation of Stimulus A. Stimulus A had to be held in memory until a forced-choice question regarding Stimulus A was presented. The forced-choice question was always presented after Response B, so that there was no possibility to prepare Response A before or during Task B. In other words, Task B could only be influenced by the representation of Stimulus A.
The Generalized Code-Occupation Hypothesis would predict that the performance of Response B is impaired when it overlaps with Stimulus A, because the overlapping feature is occupied by the object file of Stimulus A.

In the following experiments, the overlapping feature will always be the position of Stimulus A and Response B. According to the Generalized Code-Occupation Hypothesis, costs of feature overlap are only expected when the features of Stimulus A are integrated before the preparation of Response B begins. If SOA is very short, Response B should be facilitated by an overlapping Stimulus A: The overlapping feature will be activated but not occupied. This hypothesis will be further discussed in section 4.4 (Exp. 7).

In all the experiments reported in this chapter, Stimulus A had three features: Position (left or right from the screen center), Color (red or green), and Form (rectangular or round). There are two reasons for this number of features: First, more than one feature is necessary because feature integration requires the existence of more than one feature. Second, the memorization task should not be too hard for the
participants. In a pilot study, participants had great difficulties with memorizing four Stimulus A features, this leading to high error rates.

4.2 Experiment 5

The effect of an already constructed object representation on action planning is investigated, using the above described paradigm.

4.2.1 Method

Participants. Twelve volunteers aged 20 to 33 years were paid for their participation in a single session of about 15 min. One more participant was excluded for exceeding the error criterion of 20% overall.

Apparatus and stimuli. The same equipment as in Experiment 1 was used. Of each of the two response panels, only the lower sensor key was used. A white asterisk served as the first fixation appearing at the center of the black screen. Stimulus A had three features: Position (left or right), form (circle or square), and color (red or green). From a viewing distance of about 60 cm, Stimulus A measured 1.7° in width and in height. The distance between the center of Stimulus A and screen center measured 7.0°. A white dot of 0.1° in width and height was used as the second centered fixation. The letters X or H served as Stimulus B, measuring about 0.3° in width and 0.4° in height. Three forced-choice questions, one for each feature of Stimulus A, served to test whether the participants remembered the features of Stimulus A. For each question one of the two possible feature values was centrally displayed as a word (on each trial one of the two values was chosen at random). Left and right from the respective feature-value, the German words ‘Ja’ (yes) and ‘Nein’ (no) appeared. Half of the participants were presented with the word ‘Ja’ on the left side and ‘Nein’ on the right side. The other half saw ‘Nein’ on the left and ‘Ja’ on the right side. The order of the three questions was randomly chosen in each trial.

Design and procedure. There were four conditions resulting from the orthogonal variation of two within-participant factors with two levels each: Position of Stimulus A
(left or right) and side of Response B (left or right). Participants worked through a practice block of 8 trials and an experimental block of 80 error-free trials (4 conditions x 20 replications), both randomly ordered.

Following an intertrial interval of 1,250 ms, a fixation asterisk was presented 350 ms, followed by Stimulus A. Stimulus A was presented for 4,000 ms, followed by a blank of 2,000 ms. Then a centrally presented fixation point was presented for 500 ms, followed by a 500 ms blank. Stimulus B followed for 200 ms and was followed by a blank. Response B was the same as in the previous experiments, that is, a left or right finger lifting and retouching the home key. If Response B was correct, the questions concerning Stimulus A followed. The participants had to decide whether the presented feature was a feature of Stimulus A (Yes) or not (No) (see Fig. 6). Response A was a left or right finger lifting from the home key, corresponding to the answering categories “Ja” (Yes) and “Nein” (No), which appeared left and right from the question. If a question was answered wrongly, no further questions were presented.

![Diagram of a trial in Experiment 5](image)

Figure 6. Example of a trial in Experiment 5.

Participants were instructed to memorize the features of Stimulus A first and to respond to Stimulus B as fast as and accurately as possible, and then to answer all
questions about Stimulus A. They were encouraged to be quick in Task A, but they should give priority to accuracy. For errors in response type or if Response B was an anticipation, that is, was faster than 100 ms, or not completed within 1,000 ms, or when a question regarding Stimulus A was anticipated, that is, answered faster than 100 ms, or not answered within 5,000 ms, a brief error message was displayed before the next trial started.

Error trials were recorded and repeated at some random position in the remainder of the block. The two possible mappings of Response B on the identity of Stimulus B were counterbalanced across participants.

For Response B, two measures were taken: The first to-be-emitted reaction, RT was measured from the onset of Stimulus B to the first release of the home key, and MT was measured from the first release to the following retouching of the home key. For each of the three questions (as to color, form, and position) RT was measured from its onset. RTs refer to error-free trials only, while PEs refer to liftings of the incorrect finger as proportion of all in-time responses.

After each 10 error-free trials, visual feedback was given about the mean RT of Response B and the number of error trials. An integrated index of performance was also shown (= mean RT of Response B + mean RT of Response A [i.e. mean for all three questions] + number of errors x 50), including information about improvements relative to the preceding block. Participants were paid a small bonus according to the mean index of performance.

The three forced-choice questions were presented after each other. A maximum answering time of 5 seconds was given. When an answer was wrong, the trial ended.

4.2.2 Results

Response B. For each participant, mean RTs, mean MTs, and PEs were computed as a function of response-feature overlap (i.e., same vs. different sides of Stimulus A and Response B; see Table 5 for group means). The RT analysis yielded a significant effect of overlap, \( F(1,11) = 6.35, p < .05 \), indicating that the latency of
Response B was longer with overlap between Response B and the Stimulus A position than without overlap.

Neither the MT nor the PE analysis yielded significant effects for Response B.

Response A. For each participant, the mean RTs for each question was computed as a function of response-feature overlap (i.e., same vs. different sides of Stimulus A and Response B). The PEs referred to an error in at least one of the forced-choice questions.

An additional analysis was computed with the feature type of the question as an independent variable. The main effect for the feature type of the question was highly significant, $F(2,22) = 8.27, p < .01$. T-Tests revealed that the color question was answered more quickly than the form question, although significance was not reliable at the .05 level, $t(11) = 1.94, p < .079$. The form question was answered quicker than the position question, although significance was just above the .05 level, $F(1,11) = 2.17, p < .053$. The color question was significantly faster answered than the position question, $t(11)=3.93, p < .01$.

Table 5:
Mean Reaction Times (RTs), Movement Times (MTs), and Proportion of Errors (PEs) for Experiment 5 as a Function of Feature Overlap Between Stimulus A and Response B.

<table>
<thead>
<tr>
<th></th>
<th>Response B</th>
<th></th>
<th>Response A</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$RT$</td>
<td>$MT$</td>
<td>$PE$</td>
<td>$RT_{Color}$</td>
</tr>
<tr>
<td>Overlap</td>
<td>482</td>
<td>91</td>
<td>2.0</td>
<td>767</td>
</tr>
<tr>
<td>No</td>
<td>469</td>
<td>92</td>
<td>2.5</td>
<td>771</td>
</tr>
</tbody>
</table>

4.2.3 Discussion
The results show an impairment of action planning with feature overlap between the action to be performed and the memorized object. This is reflected in the RTs of Response B, but not in the error rates or MTs. The error rates are very low overall, most probably because the participants were encouraged to be accurate and errors had a negative influence on the ultimate score. Altogether, the results are in favor of the Generalized Code-Occupation Hypothesis: Action planning can be impaired when the codes required for it are occupied by an object representation.

The analyses of Response A suggest that color is easier to remember than form, and that form is easier to remember than position. However, importantly, none of these effects interacted with the factor overlap.

4.3 Experiment 6

The results of Experiment 5 showed that action planning is impaired when some of the action features are part of an already formed object representation. This suggests that the influence of an object file on action planning (Exp. 5) and on perception (Kahneman et al., 1992) is similar (see chapter 1 for a detailed discussion of their paradigm). However, there are two crucial differences between both studies: First, in the Kahneman et al. experiments, the object features in the preview field were not task-relevant. In Experiment 5, however, the overlapping feature, that is the position of Stimulus A was task relevant, because it had to be reported later. Second, in the Kahneman et al. (1992) study the previewed stimulus was not intentionally held in memory whereas in Experiment 5 Stimulus A was intentionally held in memory during Task B.

Thus, the question remains whether the costs of feature overlap result from the relevance of the overlapping feature in Task A. So far, in all experiments of this thesis (Exps. 1 to 5) the overlapping feature was relevant in both Task A and B. Therefore, in Experiment 6 it was investigated whether the costs of S-R feature overlap depend on the relevance of the object features for Task A. According to the object-file approach, object features will be integrated irrespective of their relevance. To study this, the individual
Stimulus A features are made irrelevant. Although the stimulus material is the same as in Experiment 5, participants had only to decide whether Stimulus A was present or not. If it was present, they continued the trial with Task B (go); if not, they withheld from reacting to the following Stimulus B (no go). In other words, the mere presence of Stimulus A functioned as a go-signal, its absence as a no-go-signal. The go-signal was presented for 4,000 ms for two reasons. First, this ensured that Stimulus A was attended and its individual features could be integrated. Second, in order to make the experiment as comparable as possible to Experiment 5, the same timing of events was used.

It was expected that the planning of Response B would be impaired in case of overlap with the go-signal features, because it is the object-file formation and not the active memorization that causes the overlap costs. If costs of feature overlap are caused by the process of active memorization, one should not expect them in this experiment, since no individual feature of Stimulus A had to be memorized at all.

4.3.1 Method

Participants. Ten volunteers between 21 and 30 years of age were paid for their participation in a single session of about 15 min.

Apparatus and stimuli. The same stimuli and equipment was used as in Experiment 5.

Design and procedure. The same design as in Experiment 5 was used, except for the fact that, additional to the standard trials, 12 catch trials were included, these serving as no-go conditions.

The standard trials differed from the trials in Experiment 1 in one respect (see Fig. 7). Stimulus A functioned as a go-signal, that is, only when Stimulus A appeared, participants had to react to Response B. The trial ended with Response B, that is, there were no questions concerning the features of Stimulus A.

Catch trials differed from standard trials in one respect: Stimulus A was not presented. In those trials the interval between the fixation asterisk and Stimulus B was so long as if Stimulus A was presented. If participants reacted to Stimulus B in a catch trial, that is, lifted a finger from one of both response keys within 1000 ms after the
onset of Stimulus B, they were warned with two loud beeps from the loudspeaker. If a participant falsely reacted in two catch trials the experiment ended.

After every 10 error-free trials, visual feedback was given in the same manner as in Experiment 5. Participants were paid a bonus according to their mean index of performance at the end of the experiment.

![Diagram](image)

**Figure 7.** Left: Example of a standard trial in Experiment 6. Right: Example of a catch trial.

The participants were correctly told that they had to pay full attention, otherwise they would be at risk to make errors in the catch trials. They also knew that the experiment would be broken off if they made more than two errors in the catch trials.

In the standard trials, for Response B the same measures were taken as in Experiment 5.

### 4.3.2 Results

For each participant, mean RTs, mean MTs, and PEs were computed as a function of response-feature overlap (i.e., same vs. different sides of Stimulus A and
Response B; see Table 6 for group means). The RT analysis yielded a significant effect of overlap, $F(1,9) = 7.64, p < .05$, indicating that the latency of Response B was longer with overlap between Response B and Stimulus A than without. Neither for the MTs nor for the PEs of Response B significant effects were found.

Table 6:

Mean Reaction Times (RTs), Movement Times (MTs), and Proportion of Errors (PEs) for Experiment 6 as a Function of Feature Overlap Between Stimulus A and Response B.

<table>
<thead>
<tr>
<th>Overlap</th>
<th>RT</th>
<th>MT</th>
<th>PE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlap</td>
<td>430</td>
<td>100</td>
<td>3.0</td>
</tr>
<tr>
<td>No Overlap</td>
<td>420</td>
<td>101</td>
<td>2.5</td>
</tr>
</tbody>
</table>

4.3.3 Discussion

The results show that the costs of feature overlap are also found when the overlapping feature is not relevant for Task A. This strongly suggests that if an object is attended, all object features are (automatically) integrated.

There is a similarity between this result and the non-specific preview effect Kahneman et al. (1992). In their experiment, the spatial feature of the previewed stimulus was irrelevant, but still this feature was coded and interfered with the spatial feature of the target object. This similarity suggests three conclusions concerning object files. First, all features of an object are integrated into its object file. Second, and more important, object files are long-lasting: Although the stimulus is no longer perceivable, its object file still exists. This follows from the fact that although the feature conjunction of the stimulus was useless for the task, it still influenced Response B. Conversely, action files seem to vanish apart when the action is completed (shown by the overlap benefit for Response A). The possible function of this difference will be discussed in the
General Discussion at the end of this chapter. Third, object files definitely influence action planning.

4.4 Experiment 7

The results of Experiment 5 and 6 are strong evidence for the S-R overlap-cost hypothesis. Furthermore, Experiment 6 reveals that the effect of object representation on action planning is very similar to the effects of object files. Experiment 7, then, addresses the question whether the dynamics of action planning are comparable to those of object-file formation. Experiment 3 suggested two stages in the construction process of action plans: The individual feature codes are activated, and then bound together.

The design of Experiment 7 is analogous to that of Experiment 3. Participants took part in two sessions, one for studying the first stage of object-file construction, the other for studying the second stage. In the session for studying the first stage, the SOA between Stimulus A and B was 100 ms. Additional to the short interval, Stimulus A stayed visible during Task B. Thus, in this session Stimulus A did not have to be memorized before the onset of Stimulus B. It was expected that the object features are activated but not integrated within 100 ms. Hence, the planning of Response B should be facilitated in case of feature overlap. In the session for studying the second stage of object-file construction, a procedure similar to that in Experiment 5 was used. Stimulus A was presented for 1,000 ms, followed by a mask. The SOA between Stimulus A and B was 2,000 ms. Hence, costs of feature overlap were expected because the action planning took place after the integration of the Stimulus A features was completed.

In this experiment, Stimulus B was not presented centrally, as in all experiments so far. Stimulus B now was the change of brightness of the whole display except for the mask covering Stimulus A. Thus, in order to attend Stimulus B, no additional eye-movements or shifts of attention were necessary, which excludes all alternative
explanations of the costs of feature overlap based on incompatible eye movements between the presentation of Stimulus A and B.

In summary, an interaction between the factors memorization and overlap was expected. Costs of feature overlap were expected when the to-be-planned action shares a feature with the already established object file of Stimulus A, whereas benefits of feature overlap were expected when the to-be-planned action shares a feature with the not fully completed object file.

4.4.1 Method

Participants. Twelve volunteers between 19 and 47 years of age (mean age 26) were paid for their participation in two sessions of about 15 min each.

Apparatus and stimuli. Additional to the equipment in Experiment 6, the numerical keyboard of the PC was used. Stimulus A was the same as in Experiment 5. Stimulus B was a change in brightness of the background (from grey to black or from grey to white). Stimulus A was covered by a mask of small randomly ordered red and green rectangles (76 x 12 rectangles). The mask measured 15.1° in width and 2.9° in height and was always presented centrally. The question concerning Stimulus A after Response B was presented in the form of eight yellow rectangles, each representing a possible Stimulus-A display, from which one had to be chosen. Each rectangle measured 2.1° in width and 1.4° in height. The eight rectangles were ordered as a square, that is, there were three rows, the upper and lower row containing three rectangles and the middle row two (see Fig. 8). The formation of the eight rectangles on the screen corresponded to the eight keys around the 5 of the numerical part of the PC keyboard. These eight keys were marked with blue stickers. The PC keyboard was always positioned left from the left response panel.

Design and procedure. There were eight conditions resulting from the orthogonal variation of three within-participant factors with two levels each: Object-position (left or right), side of Response B (left or right), memorization of the object (yes or no). In either session, participants worked through a practice block of 16 trials (4 conditions x 4 replications) and an experimental block of 80 error-free trials (4 conditions x 20
replications). Error trials were recorded and repeated at some random position in the remainder of the block.

The two possible mappings of Response B on the color of Stimulus B and the two possible orders of the sessions were counterbalanced across participants. After each 10 error-free trials, visual feedback was given about the mean RT of Response B and the number of error trials. An integrated index of performance in Task B was also shown (= mean RT + number of errors x 20), including information about improvement relative to the preceding block.

The order of events in the memorization session is illustrated in Figure 8. Each trial started with a yellow fixation asterisk at the screen center, which was presented for 350 ms and followed by a 50 ms blank. The background color was grey. Then, Stimulus A was presented for 1,000 ms and was then covered by a mask containing small red and green squares. 1,000 ms later, the screen, except for the mask, turned black or white, indicating the hand for Response B. After another 200 ms the background turned grey again. When Response B was completed, the whole screen turned grey. Then the 8 small rectangles were presented, each containing a small object with three features, that is, color, position in the rectangle, and form. The participant had to search the rectangle that corresponded to the previously memorized Stimulus A. This task was not speeded.

In the no-memorization session, each trial also started with the onset of a yellow fixation asterisk for 350 ms at screen center and was followed by a 50 ms blank. Thereafter, Stimulus A appeared and stayed visible up to 685 ms after Response B had been completed. 100 ms after the onset of Stimulus A, the whole screen except for the object itself turned black or white for 200 ms to signal Response B. When Response B was given, Stimulus A was masked for 500 ms in the same way as in the other session. The remaining procedure was as in the memorization session.

In both sessions, Response B had to be performed within 1,000 ms, but was not allowed to occur in the first 100 ms after the onset of Stimulus B, in order to exclude anticipations.
4.4.2 Results

Response B. Mean RTs, MTs, IRTs, and PEs were computed as a function of response-feature overlap and memorization (see Table 7 for group means of Responses A and B). For the RTs, there was a highly significant interaction between memorization and feature overlap, $F(1,11) = 78.56, p < .001$, due to that feature overlap influenced RTs positively in the no-memorization session, and negatively in the memorization session. Separate analyses of both sessions revealed significance for feature-overlap benefits in the no-memorization session and feature-overlap costs in the memorization session. In the no-memorization session $F(1,11) = 18.68, p < .001$, in the memorization-session, $F(1,11) = 5.10, p < .05$. Analysis of MTs yielded no significant effects.

For the PEs, there was a significant interaction between memorization and feature overlap, $F(1,11) = 7.72, p < .05$. Separate ANOVAs of both sessions showed
that only in the no-memorization session there was a significant difference between overlap and no overlap, $F(1,11) = 8.21, p < .05$. This indicates, that in that session less errors were made in case of overlap than with of no overlap.

Response A. For the RTs there was a significant main effect of feature overlap, $F(1,11) = 9.12, p < .05$, showing that the time to find the right answer was shorter when there was an overlap between Stimulus A and Response B. This was the case in both sessions. In the no-memory session $F(1,11) = 6.61, p < .05$, in the memory-session, $F(1,11) = 5.46, p < .05$.

For the PEs, there was a significant main effect of memorization, $F(1,11) = 6.44, p < .05$, showing that in the memorization session the error-rate was higher than in the no-memorization session.

**Table 7:**
Mean Reaction Times (RTs), Movement Times (MTs), and Proportion of Errors (PEs) for Experiment 7 as a Function of Feature Overlap Between Stimulus A and Response B and the Memorization of Stimulus A.

<table>
<thead>
<tr>
<th></th>
<th>Response B</th>
<th></th>
<th></th>
<th>Response A</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT</td>
<td>MT</td>
<td>PE</td>
<td>RT</td>
<td>PE</td>
</tr>
<tr>
<td>no memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td>359</td>
<td>88</td>
<td>3.0</td>
<td>1749</td>
<td>1.0</td>
</tr>
<tr>
<td>No Overlap</td>
<td>376</td>
<td>86</td>
<td>6.0</td>
<td>1832</td>
<td>1.5</td>
</tr>
<tr>
<td>memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlap</td>
<td>384</td>
<td>88</td>
<td>4.0</td>
<td>1888</td>
<td>5.1</td>
</tr>
<tr>
<td>No Overlap</td>
<td>374</td>
<td>87</td>
<td>4.0</td>
<td>1998</td>
<td>5.4</td>
</tr>
</tbody>
</table>

4.4.3 Discussion

The results supply clear evidence for the hypothesis that the construction process of object files passes through the same two stages as that of action plans. The first stage is the activation of the individual feature codes, the second stage the integration of these
feature codes. According to this hypothesis, it was expected that action planning is facilitated when the construction of an overlapping object file is in the first stage, but hampered when it had already been in the second stage, because after the second stage, the codes required for the object file are occupied.

In the no-memorization condition the SOA between Stimuli A and B was only 100 ms. Response B was clearly facilitated in case of an overlap with Stimulus A. In the memory condition, however, where the SOA amounted to 2,000 ms, action planning was faster and less errors were made in case of no overlap. The interaction between these two conditions was highly significant, thus, the effect of facilitation turned into a cost effect.

The lower search times in Task A when Stimulus A and Response B overlapped indicate that Stimulus A was better recognized when the position of Stimulus A and Response B were the same. This is in accordance with the interpretation of the response priming in chapter 2. It was concluded that after the execution of Response B the remaining activity of the involved codes facilitates subsequent similar actions. The moment Stimulus A has to be recognized, this process seems to be facilitated by the remaining activity from the execution of Response B. In other words, not only the initiation of an action can benefit from the remaining activity in a previously performed action, but also the process of recognition.

The pattern of errors hardly needs any special explanation. The finding that Response A was more often wrong in the memorization session than in the no-memorization session simply follows from the fact that Task A was much easier in the no-memorization session (since there was no interfering Task B).

4.5 General Discussion

The main purpose of the experiments reported in this chapter was to find out whether the S-R overlap-costs prediction from the Generalized Code-Occupation Hypothesis indeed holds. The prediction is that action planning is impaired when part of the codes necessary for the construction of the action plan are already integrated in an
object representation. In order to test this prediction, the ABBA paradigm used in chapter 2 was changed in minor points. The most important change was that now the overlap between Response B and Stimulus A instead of the overlap between Response B and Response A was manipulated.

In accordance with the Generalized Code-Occupation Hypothesis, Experiment 5 showed that planning an action is impaired when it shares a feature with a previously memorized object. Thus, object representations occupy codes in the same vein as action plans. Experiment 6 showed that even when object features are irrelevant for the task at hand, they are occupied by the object representation and, hence, are less available for the following action planning. Experiment 7 showed that object files pass through the same two construction stages as do action plans. First, the individual feature codes are activated. Second, the activated feature codes are integrated.

Moreover, Experiment 7 yielded evidence for the response priming hypothesis, because the answer to the question about Stimulus A was faster when Stimulus A and Response B shared a feature.

Two theoretical implications will be discussed in more detail. First, after having focussed on the similarities between object and action representations, I will discuss the differences between these. Second, I will discuss in more detail the benefit of overlap on Response A.

4.5.1 Differences Between Action Plans and Object Files

Altogether, the results of the experiments 5 to 7 suggest that object files and action plans are similar with respect to both structure and dynamics. Still, there are some important differences, which will be discussed in the following.

Response-related costs of feature overlap, on the one hand, seem to be bound to active planning, as shown in Experiment 3, and are rather short-lasting. The latter is indicated, first, by Wühr and Müßeler’s (1997) observation that difficulties in perceiving response-compatible stimuli disappear as soon as the overlapping response is executed, and, second, the finding in this thesis that overlap costs are confined to
Response B, but do not occur in the following Response A. Apparently, planning an action is associated with feature binding, but the respective feature codes are released and are then available for other bindings soon after the action plan is realized.

Stimulus-related overlap costs, on the other hand, seem to be due to automatic processes and are rather long-lasting. In Kahneman et al.’s (1992) original preview design or in Hommel’s (in press a) adaptation the mere presentation of a task-irrelevant stimulus is sufficient to cause overlap costs in perceptual processes that run off one or more seconds later. The same is true for coincidental stimulus-response conjunctions, whose automatically formed representations are stored for at least one second (Hommel in press a). These long-lasting effects of object files were even more pronounced in Experiment 6, where there was a delay of 3,000 ms between the offset of Stimulus A and the onset of Stimulus B. These findings suggest that the binding of perceptual features (with other perceptual or action features) is much more automatic than the binding of action features (with other action features). Moreover, the integrated event representations are maintained much longer for perceptual than for pure action events.

One may tentatively speculate about the function of this difference between object- and action files: It seems efficient to hold in memory the object file a little longer than the physical object is actually visible. In the natural environment, objects do not simply vanish as is the case in the psychological laboratory. They may disappear from the visual field when they are hidden behind other objects. We would not be able to represent the world surrounding us if we represented perceivable objects only. Thus, the object file may be used as representing the environment, rather than the retinal map. This is different in the case of action plans. After performing an action it makes no sense to hold the action representation in memory any longer. This may even be counterproductive, as it could lead to action perseveration.

4.5.2 Response Priming is Based on Abstract Response Codes

The results of Experiment 7 showed that Response A was faster when Stimulus A and Response B shared the spatial feature code. In that experiment, for performing
Response A, the participants had to compare the remembered Stimulus A with all eight Stimulus A possibilities. The RT was shorter when the position of the remembered Stimulus A was the same as that of Response B, which was performed shortly before, this suggesting that Response B influenced the memory trace for Stimulus A. This finding is similar to the findings of chapter 2, which showed that Response A benefits from an overlap with Response B.

Altogether, the benefit of feature overlap on Response A has been found in a number of experiments in this thesis. The effect cannot be accounted for by an anatomical overlap between the priming (Response B) and the primed (Response A) Response, because Response A and B did not always overlap in an anatomical sense. In Experiment 2, response priming was found when there was similarity between a foot and a hand reaction. In Experiment 7, a visual-search task was speeded when the object to be looked for shared a feature with the previously performed response. This suggests that it is the overlap of the responses on the central level of action representation causing the overlap benefit, and not just the overlap on the peripheral level of the effectors. This conclusion is also in accordance with the findings of Campbell and Proctor (1993), who found that response priming can occur between different hands if the responses share a property.
Chapter 5

Conclusions

Section 5.1 structures the essential conclusions drawn from the previous chapters. In section 5.2, I will discuss some remaining theoretical implications of the empirical findings. Subsequently, in section 5.3, I present a formalized feature-integration model of action planning. The final section reports the most essential contributions of this work.

5.1 Basic Conclusions

The findings of this work can be summarized in five conclusions. First, it has been demonstrated that feature codes are more difficult to access when they are already integrated into an action plan or object file. All the experiments of chapter 2 showed that planning an action was impaired if at the time of planning an already prepared action plan was being held in memory that overlapped with intended action. The experiments of chapter 4 showed that the planning of an action was impaired if at the time of planning an already constructed object file existed that overlapped with the features of the to-be-planned action. These findings support the Generalized Code-Occupation Hypothesis, which states that feature codes are less accessible for cognitive operations (like action planning or object-file formation) if the codes are already part of other representations (e.g., other plans or object files). Most importantly, it does not matter what the functional role of the occupied codes is. In other words, it is irrelevant whether an occupied code is part of an object or of an action representation.

Second, it has been shown that the action-feature codes are of abstract nature. Especially the results of Experiment 2 and those reported in chapter 4 show that the construction of an action plan can be impaired if the only overlapping code represents response side. As has been pointed out in detail in the general discussion of chapter 2, this finding supports the assumption that actions plans take abstract parameters and are part of a higher level of abstraction than pure muscle commands. Furthermore, the
finding that Response A can be primed by a similar but anatomically not overlapping response suggests also that action plans are built from abstract codes.

Third, it has been shown that planning an action as well as constructing an object file is a two-stage process. In the first stage the individual feature codes of the to-be-constructed representation are activated, and then integrated in the second stage. Experiment 3 showed that if action codes are activated but not integrated into a specific action plan, they facilitate the preparation of a different plan. Only if the codes are integrated are they less available for different processes of action planning. Experiment 7 revealed that if object features are activated but not integrated into an object file, they facilitate the preparation of an action plan that overlaps with the object. Only after the integration of the object features is completed, the planning of a subsequent action plan is hampered if the action planning requires features that are part of the object representation. As has been argued in chapter 3, this model has important implications for the description of the processes underlying S-R compatibility phenomena. Facilitation of S-R overlap on the construction of an action plan is only to be expected if the integration of the stimulus codes has not been completed.

Fourth, it has been demonstrated that the bindings between the feature codes of an action plan fall apart after execution. In the experiments of chapter 2 and in experiment 7 of chapter 4 it has been shown that if an action plan has been executed, the remaining activity in the action codes facilitates the following action if that action overlaps with the first executed action plan. This is an effect of response priming. Thus, as long the codes of an action plan are integrated, they are less available for different actions, but directly after the execution the action codes are given free and then facilitate overlapping different actions. It has been shown in Experiment 4 that the remaining activity in action codes automatically decays after the execution of the plan they were part of.

Fifth, it has been shown that object files are long-lasting and automatically established. These characteristics object files were most clearly demonstrated in Experiment 6. The individual values of the features had no task relevance, but
nevertheless they influenced subsequent action planning, even with 3 s between the offset of the object and the target stimulus of the subsequent task.

5.2 Implications of the Results

Additional to the general discussions at the end of chapters 2 and 4, I will discuss some more general implications of the findings in the following. First of all, I will discuss that the binding problem in action planning poses a problem for those theories of action planning that exclude the possibility of coactivated action plans. Second, I will compare the costs of feature overlap with the effect of reversed response priming, as reported by Rogers and Monsell (1995). This comparison is especially interesting if one considers the results of both chapter 2 and 4.

5.2.1 Consequences for Theories of Action Planning

The study of action planning includes several aspects: The most important are the way actions are coded (cf. Hommel, 1997), how action plans are constructed (cf. Rosenbaum et al., 1984), how action planning is related to perception (cf. Prinz, 1997a), and the way in which action planning is acquired and improved by learning and feedback (cf. Adams, 1971; Schmidt, 1976). As has been discussed in chapter 1, studying the binding problem in action planning is new, making it difficult to contrast the present findings with other research in the domain of action planning. Nevertheless, the present work points to a shortcoming in some concepts of action planning. Almost all research focusses on performance in a single task. Even in studies of dual-task performance, it is not considered how two tasks can be handled at the same time, because it is, a priori, assumed that the tasks are handled serially (for an exception see Hommel, in press b). This work, however, has revealed that more than action plan can be active at the same time and shows how different active action plans can be distinguished by the cognitive system.
The study of the construction of action plans has been strongly influenced by the idea of motor programming. Motor programming was first defined by Keele (1968) as “as set of muscle commands that are structured before a movement sequence begins, and that allows the sequence to be carried out uninfluenced by peripheral feedback” (p. 387). The concept of motor programming was picked up and modified by Rosenbaum and colleagues. Rosenbaum et al. (1984) proposed the Hierarchical Editor Model, which can be considered as a well established theory of action planning. This theory suggests that action plans are organized in a hierarchical fashion and that serial behavior (like the key-pressing tasks of the present work) is based on serial processing. Although the model, based on the programming of a sequence of movements, is good at explaining RTs and MTs in discrete finger movement tasks, this model explicitly excludes the parallel activation, maintenance, and preparation of action plans. Therefore, it is impossible to predict or explain results collected in an ABBA paradigm, because here two tasks are handled in parallel.

A similar problem posits the idea of one single motor buffer, holding the output for the motor system. Such a buffer was originally proposed by Henry and Rogers (1960), and presupposes that only one prepared motor program can be maintained. The present work, however, clearly show that one prepared action plan can be maintained while another action is prepared and executed. Logically, this means that there should at least be two buffers. The results show that the cognitive system is able to maintain more than one action plan, and that a binding mechanisms is responsible for the separation of the different plans.

In summary, theories of action planning should be able to explain how several action plans can be prepared and maintained in parallel, especially since the involvement in multiple tasks is not an exceptional situation for humans, but rather characterizes our usual behavior. Theories of action planning based on the serial character of motor programming and theories that assume just one motor buffer do not allow for the parallel maintenance of different actions plans, and thus are not appropriate for describing processes underlying the binding aspects of action planning.
5.2.2 Overlap Costs and Reversed Response Priming

The reported costs of overlap seem to be counterintuitive if compared with the phenomenon of response priming. Response priming is a well-known effect (cf. Bertelson, 1965; Rabbitt, 1968): In a choice-reaction task paradigm, a response is faster if the same response was also selected in the previous trial. In the experiments reported in chapter 2 there exists a somewhat similar situation as in response priming experiments: In the overlap conditions Response B was selected for the second time (because having been selected for the preparation of Response A). In other words, in the case of overlap Response B should be primed by Response A, rather than being inhibited due to their similarity. Of course, it can be argued that in the ABBA paradigm Response B is the first response to be executed response, and is therefore not really preceded by a similar Response A. Still, there exists an interesting similarity between the costs of feature overlap and the effect of inverted response priming, reported by Rogers and Monsell (1995). These authors found that the effect of response priming interacts with task switching. They obtained a normal response priming if both response and task are repeated, but inversed response priming if the response was repeated but the task was switched.

The costs of feature overlap in the present experiments are similar: First of all, Response B is slower when shortly before a similar, rather than a very much different Response A was selected had been selected. Second, Response B is given after a task switch, that is, after the switch from Task A to Task B. Thus, one could argue that the costs of feature overlap are an effect of reversed response priming through task switching.

Rogers and Monsell (1995) offer three possible explanations for the inversed response priming. First, the S-R association is strengthened after a response. This strengthening is accompanied by a weakening of those associations that did not contribute to the response. This explains why there is facilitatory effect if a response on the same stimulus category has to be given, and an interfering effect if the category is different. Second, after a task switch, the last reaction is inhibited. This is thought as a
kind of control function, that not only deactivates the “old” task, but also the corresponding active responses. Third, there must exist a mechanism that prevents the perseverative reexecution of responses. Under no-switch conditions, this effect might be overtuned by the enormous facilitatory effect of repeating the same stimulus category. Under task-switch conditions, however, the advantage of stimulus-category repetition does not exist and therefore the inhibitory effect becomes measurable.

It is hard to apply any of these three hypotheses to the costs of feature overlap reported in this chapter. The first explanation cannot account for the costs of feature overlap: A weakening of S-R relationships in Task A cannot influence the performance in Task B, because in the ABBA paradigm there are different stimuli ensembles. The second explanation could explain the impaired performance in Task B, but is inconsistent with the benefit of overlap for Response A: If Response B were inhibited after the switch from Task B to Task A, why then should Response A profit from overlap? The third explanation cannot be applied to the costs of feature overlap, as it depends on a control mechanism for preventing the reexecution of Responses. Since Response A is not executed before Response B, this explanation cannot deal with the impaired performance of Response B. It could only be applied if Response B followed the execution of a response.

So, none of the hypotheses are applicable in the context of the ABBA paradigm. However, it could well be that the reversed-response priming depends on a binding conflict, just as the costs of feature overlap do. After each task switch, the same stimulus and response codes have to be connected in a different fashion. That is, if a response has to be repeated but the S-R relationship is different, there is a partial match between the two trials. Such a partial match between two trials leads to worse performance than no match, already been discussed in section 1.4.

Furthermore, the results of chapter 4 show that the costs of feature overlap are more likely to result from a binding conflict than from inverted response priming. In the paradigm used, Task B is accompanied by a Task A representation held in memory, but no response had to be selected before Task B was done. In other words, the experiments
of chapter 4 show that feature overlap alone entails costs of feature overlap and, thus, that the costs of feature overlap cannot be explained by inversed response priming.

In sum, there are several similarities to the effect reported by Rogers and Monsell (1995). Nevertheless, none of their suggested explanations can be applied to the present findings. It is suggested that the effect of inversed response priming is a special situation of a binding conflict.

5.3 A Feature Integration Model of Action Planning

The findings of this work revealed that action planning depends on both the activation and the binding of feature codes. In the following, I will propose a model directly inspired by the neurophysiological concept of synchronization (Singer, 1994; Von der Malsburg, 1981). Although the model functions only as an illustration of how the discussed mechanisms could be realized, it has important advantages. First, both the activity and the binding with other codes are derived from the firing of codes. Second, the model shows that the two-stage model of action planning can be realized in an extremely simple manner. In other words, no complex extra assumptions are necessary for the binding of feature codes.

5.3.1 Assumption 1: Action plans have a distributed representation

The model assumes that action plans are represented in a distributed way: Their representations consist of a set of elements, each of which can be part of different representations. In Figure 9, an example of two overlapping action plans is illustrated, the circles representing the elements or codes and the dotted ellipses the action plans. The elements of action plans are abstract action codes, like direction or force. The model assumes these elements are interconnected. There are two types of such connections, temporal and fixed ones. It is assumed that temporal connections are based on synchronous firing of codes, whereas fixed connection are based on hard-wired connections between codes which developed through the repeated coactivation of two
codes. I will not discuss the development of fixed connections in further detail. Most connectionist models are based on the slow development of such connections, principally based on the Hebbian Rule of Learning, which states that the association of two neurons will grow stronger if their activity is correlated (Hebb, 1949).

![Figure 9. Each action plan (dotted ellipses) is built from a set of codes (circles).](image)

5.3.2 Assumption 2: Feature codes have two properties: activation and temporal pattern

The central properties of codes are activity and the temporal pattern of activity. It is assumed that each action code can be in two states: firing or not firing. This firing state is comparable to the spiking of a single neuron. If the “spikes” of a code are considered on a certain time interval, two properties can be derived.

First, there is the activity of codes. The activity is also called “rate coding” because this property is a derivation from the mean firing rate in a time interval, independent from the distribution of the spikes over the interval (see Fig. 10). The activity of a code represents its “involvement” in future actions.

Second, there is temporal pattern of firing of codes. The temporal pattern of firing refers to the distribution of the spikes over the time interval, and represents the relationship with other codes. This assumption is inspired by the temporal binding hypothesis, first expressed by Von der Malsburg (1981). The basic idea, which has already been expressed in chapter 1, is that the activity pattern of a neuron contains
information about its relationship to other neurons. There is evidence for this hypothesis in the visual areas, but there are also data indicating that this type of coding is used all over the cortex; thus, also in action planning (Murthy & Fetz, 1992).

![Figure 10](image)

Figure 10. The property Activity refers to the number of spikes over a time interval. The temporal pattern of firing refers to the information that lies in the distribution of the spikes over the interval.

5.3.3 Assumption 3: Integration through synchronization

The temporal binding of codes is represented by synchronous firing. This assumption is based on the temporal binding hypothesis, postulated by Singer (1994). It is important to stress here that two codes can be on the same level of activation with different temporal patterns, and that two codes can have different activation levels but still fire in synchrony (see Fig. 11). Thus, activation and synchronous firing are rather independent properties.
5.3.4 Assumption 4: Feature integration costs time

The process of feature integration implies that two or more feature codes, which are not bound, are about to be bound. As temporal binding is realized by synchronous firing, the process of binding can be viewed as a tuning process (cf. Figure 12). It is assumed that this tuning costs time because a kind of communication process between the two spatially remote codes has to occur.

Figure 11. Codes belonging together fire in synchrony.

Figure 12. In order to integrate code C3 into Plan 1, the temporal pattern of C3 has to be tuned to the pattern of C1 and C2.
5.3.5 Assumption 5: Code sharing leads to higher activation

Feature codes can be integrated in more than one representation. As each representation has its own individual action pattern, shared codes fire synchronous to all representations they belong to. This means that the activation pattern of a code that is part of more than one representation is equivalent to a superposition of the activation patterns of all the representations. This implies that the activation of a code correlates with the number of representations it belongs to (see Fig. 13).

![Figure 13. Code C2 is part of 2 plans, and its temporal pattern is the superimposition of C1 and C2. C2 has thus a higher activity.](image)

5.3.6 Assumption 6: Action plans fall apart after the action goal is reached

As discussed in chapter 2 and 4, it is assumed that action plans fall apart as soon as the action goal has been reached. In this regard, action plans are different from object files, which seem to exist even when the represented object is no longer present in the surrounding world. In the general discussion of chapter 4, some evolutionary functional reasons for this difference are given. The falling apart of a group of feature codes means that they no longer fire in synchrony. It is assumed, however, that the decay of activity in each of the individual feature codes takes some time.
5.3.7 Assumption 7: Actions are represented on the common-coding level

It is postulated that action plans are represented in terms of future sensory states (see chapter 3). The feature codes always refer to external events and not to some peripheral patterns of excitation in the muscles. This assumption is important to explain the interactions between object and action representations. It is put forward that it does not matter for the first six assumptions presented here, whether a feature code is involved in object representation or in representing a future action.

5.4 Final Conclusions

The central question of the present work was whether human action planning is faced with the binding problem. The empirical findings suggest an unequivocal answer: Yes, action planning is clearly affected by the binding problem in a similar way as perception is affected by this problem.

Moreover, the results give rise to a new model of the construction of action plans, that is, a two-stage model. Essential for this model is that action plans are represented by activated temporally integrated action features. It can be generalized to the construction of object files, as has been theoretically and empirically underpinned in the chapters 3 and 4. The model explains how the coactivation of several plans at the same time is possible and what the consequences of this coactivation are.

Altogether, this work contributes to our understanding of an important aspect of human performance. It offers a solution to the question of how humans are able to handle multiple tasks at the same time, without confounding the parts of the tasks. This is not trivial: Our ability to handle two or more tasks in parallel is not an exceptional state but is very representative for our daily functioning.

In Kapitel 2 wird ein neues Doppeltätigkeitsparadigma zur Überprüfung der im ersten Kapitel postulierten Kodebesetzungshypothese vorgestellt. Das wichtigste


Gemäß der Kodebesetzungshypothese wird Handlungsplanung beeinträchtigt, wenn die für die Handlungsvorbereitung benötigten Handlungsmerkmale Teil eines bereits erstellten Planes sind. Deshalb wird erwartet, daß Handlung B im Falle der Merkmalsüberlappung beeinträchtigt wird. Vorhergesagt wird, daß die Reaktionszeit (RT), Bewegungszeit (MT) und Fehlerrate (PE) der Reaktion B im Falle der Merkmalsüberlappung höher sein werden.

Experiment 1 fand gemäß dem oben beschriebenen Paradigma statt. Reaktion A war eine kleine Sequenz von Tastenberührungen mit der linken oder rechten Hand. Stimulus A war ein Pfeil, der die zu benutzender Hand andeutete. Der Pfeil wurde immer von einer oder zwei Zahlen begleitet, welche die Richtung der Sequenz (oben oder unten) und die Länge der Sequenz (kurz oder lang) angaben. Diese beiden Faktoren
(Richtung und Länge) besaßen die Funktion, die Reaktionsvariation zu erhöhen, so daß Reaktion A nicht als vorgefertigtes komplettes Programm direkt vom Langzeitgedächtnis abgerufen werden konnte und der Handlungsplan A immer erst nach Darbietung von Stimulus A konstruiert werden mußte. Stimulus B war ein zentral dargebotenes Rechteck in roter oder grüner Farbe, auf die mit dem rechten oder linken Finger reagiert werden sollte. Dazu sollte der betreffende Finger von der mittleren Ausgangstaste abgehoben und sofort wieder darauf zurückgelegt werden.


Experiment 2 lehnte sich Experiment 1 an. Der einzige Unterschied bestand darin, daß Reaktion B mit dem linken oder dem rechten Fuß ausgeführt wurde. Dieses Experiment sollte eine alternative Erklärung von den in Experiment 1 gefundenen Ergebnissen widerlegen. Alternativ zu der Kodebesetzungshypothese kann behauptet werden, daß die Merkmalsüberlappungskosten von einem Remapping-Prozeß verursacht werden, der nur im Falle der Merkmalsüberlappung notwendig ist. Im Rahmen dieser Idee reicht die Zuweisung einer Reaktion A zu einer Hand hin, um Reaktion A mit der ausgewählten Hand auszuführen. Muß Reaktion A allerdings aufgeschoben und mit der gleichen Hand eine andere Handlung ausgeführt werden, muß
die ursprüngliche Zuweisung gelöst und nach der Erledigung der hinzu gekommenen
Handlung neu installiert werden, d.h. der ursprüngliche Plan für die Hand muß neu
organisiert werden (engl. remapped). Wenn es keine Merkmalsüberlappung gibt
(Reaktion A und B mit unterschiedlichen Händen), ist diese Reorganisation nicht
notwendig, weil die eingeschobene Reaktion B mit der anderen Hand ausgeführt wird,
und die Hand, der Reaktion A zugeordnet worden ist, einfach warten kann. Wenn dieses
Argument stimmt, sollte der im ersten Experiment gefundene Effekt nicht auftreten,
wen Reaktion A und B in allen Bedingungen mit unterschiedlichen Effekten
ausgeführt werden müssen.

Die Ergebnisse zeigten eine signifikante Beeinträchtigung der Fußreaktion in
sowohl RT, MT und PE, wenn bereits eine Handreaktion auf der gleichen Körperseite
vorbereitet wurde. D.h. daß das Remapping-Argument nicht richtig sein kann. Darüber
hinaus zeigten diese Ergebnisse, daß sehr abstrakte Merkmale wie LINKS oder
RECHTS Teil eines Handlungsplans sein können, und dann für die Vorbereitung
anderer Handlungen nicht zugänglich sind, obwohl diese anderen Handlungen mit
anderen Effekten ausgeführt werden sollen. Weiterhin wurde wie im ersten
Experiment Reaktion A im Falle einer Merkmalsüberlappung mit Reaktion B
signifikant schneller ausgeführt als wenn die Merkmalsüberlappung fehlte.

Aus den Ergebnissen der ersten zwei Experimenten können drei wichtige
Schlußfolgerungen gezogen werden: Erstens, das Remapping-Argument gilt nicht als
alternative Erklärung der in Experiment 1 gefundenen Merkmalsüberlappungskosten.
Diese werden nämlich auch gefunden, wenn Reaktion A und B mit unterschiedlichen
Effekten auf der gleichen Körperseite ausgeführt werden, also wenn die
Merkmalsüberlappung sich nur auf die Ausführungsseite bezieht. Zweitens,
Handlungspläne können sehr abstrakte Merkmalskodes besitzen, denn die
Merkmalsüberlappungskosten werden sogar gefunden, wenn das einzige überlappende
Merkmal die Reaktionseite ist. Drittens, Reaktion A scheint bei Merkmalsüberlappung
mit Reaktion B kürzer zu dauern als ohne Merkmalsübereinstimmung. Dies ist im
Einklang mit der schon nach Experiment 1 vermuteten Annahme, daß die Merkmale
nach Ausführung eines Planes nicht mehr besetzt sind, jedoch noch eine Restaktivität besitzen, die ähnliche Handlungspläne bahnt.

In Experiment 3 wurden zwei mögliche Gegenargumente gegen die von mir vorgestellte Erklärung der Merkmalsüberlappungskosten behandelt. Erstens, es wurde weder in Experiment 1 noch in Experiment 2 sichergestellt, daß es wirklich die vorbereitete Handlung A war, die mit Handlung B interferierte. Es könnte beispielsweise der Fall sein, daß sich die Vpn Stimulus A merkten, aber erst nach Beendigung von Reaktion B interpretierten. In anderen Worten, es könnte sein, daß sich nicht Reaktion A, sondern Stimulus A im Kurzzeitgedächtnis befand, während Aufgabe B ausgeführt wurde. Deswegen wurde in Experiment 3 die Wahrscheinlichkeit manipuliert, daß Handlung A vor Aufgabe B vorbereitet wurde. Hierzu wurde die unabhängige Variable (UV) "Planung" mit den Ausprägungen "geplant" und "nicht-geplant" eingeführt.

Zweitens, da Stimulus A in bezug auf räumliche Eigenschaften nicht ganz neutral war, könnte es die inhärente Bedeutung von Stimulus A gewesen sein, und nicht der Handlungsplan A selbst, die den Merkmalsüberlappungseffekt verursacht. Um dieses Argument entgegenzukommen, wurde für Stimulus A ein Buchstabe gewählt, der keine inhärente Richtungsbedeutung hat.

umgekehrte Effekt gefunden: Reaktion B war deutlich schneller, wenn es eine Merkmalsoverlappung mit Reaktion A gab.

Es gibt folgende wichtige Schlußfolgerungen. Erstens, eine Handlung wird beeinträchtigt, wenn die Handlungsmerkmale, auf die zugegriffen werden muß, schon von einem anderen Plan besetzt worden sind. Wenn die Handlungsmerkmale jedoch aktiviert worden, jedoch nicht integriert sind, dann zeigt sich eine bahnende Wirkung. Es gibt also zwei Phasen in der Handlungsvorbereitung. In der ersten Phase werden die einzelnen Handlungskodes aktiviert. In der zweiten Phase werden sie integriert, wodurch sie für andere Prozesse schwieriger zugänglich werden.

Die Ergebnisse bezüglich Reaktion A bestätigen die vorhin schon beschriebene Vermutung, daß nach Ablauf einer Handlung die Merkmale nicht mehr besetzt, jedoch noch aktiviert sind und mit der Restaktivität nachfolgende Reaktionen bahnen können: Die Bewegungszeit von Reaktion A war signifikant kürzer im Falle der Merkmalsoverlappung mit Reaktion B.


Das gewählte Design erlaubte die Beantwortung einer zusätzlichen Frage: In den Experimenten 1 und 2 wurde Reaktion A sofort nach Beendigung der Reaktion B ausgeführt. Dies legt die alternative Vermutung nahe, daß die Verschmelzung beider
Reaktionen zu dem Effekt der Merkmalsüberlappungskosten führt. Da es in diesem Experiment keine Möglichkeit für derartige Verschmelzungen gab, konnte die Vermutung leicht überprüft werden.


In Kapitel 4 war die empirische Überprüfung der generalisierten Kodbesetzungshypothese der zentrale Punkt. Das für die ersten vier Experimente benutzte Design wurde geringfügig modifiziert, um die generalisierte Kodbesetzungshypothese zu testen. Entweder befand sich während der Ausführung von Aufgabe B eine bereits gebildete Objektrepräsentation anstelle eines Handlungsplans im Gedächtnis (Experiment 5 und 7) oder es wurde vor der Ausführung
von Aufgabe B ein Objekt (Stimulus A) als Go-Signal betrachtet, aber mußte nicht behalten werden (Experiment 6). Gemäß der generalisierten Kodebesetzungshypothese kommt es zu einer Beeinträchtigung der Handlungsplanung, falls es eine Merkmalsüberlappung zwischen den von der Objektrepräsentation integrierten Merkmale und die für die Handlungsvorbereitung notwendigen Merkmale gibt.


Nach der Ausführung von Reaktion B wurden alle drei Eigenschaften von Stimulus A abgefragt. In der Mitte des Bildschirmes erschien eine Eigenschaft als Wort (z.B. "rot"), und die Vp mußte mit einer "Ja"- oder "Nein"- Taste angeben, ob die Eigenschaft Teil des gemerkten Objekts war. Die Ergebnisse zeigten, daß die RT von Reaktion B im Falle der Merkmalsüberlappung signifikant größer war. Darüber hinaus zeigte sich, daß für die Beantwortung der Positionsfrage mehr Zeit gebraucht wurde als für die Beantwortung der Formfrage, und daß für die Beantwortung der Formfrage mehr Zeit gebraucht wurde als für die Farbfrage. Diese Unterschiede interagierten jedoch nicht mit dem Faktor "Merkmalsüberlappung".

Experiment 6 unterschied sich in zwei Aspekten von Experiment 5. Erstens, die drei Objekteigenschaften von Stimulus A mußten nicht behalten werden, denn Stimulus A diente als Go-Signal: Nur wenn Stimulus A erschien, sollte auf den darauf folgenden Buchstabe (Stimulus B) reagiert werden. Zweitens, es gab 12 CATCHtrials, in denen Stimulus A nicht erschien. Wenn die Vpn in mehr als zwei CATCHtrials reagierten, wurde
das Experiment abgebrochen, weil sonst nicht sichergestellt war, daß das Go-Signal aufmerksam betrachtet wurde. Diese strenge Regel mußte jedoch nie angewandt werden. Ziel der Änderungen im Versuchsplan im Vergleich zu Experiment 5 war herauszufinden, ob die Relevanz der Stimulus A-Eigenschaften eine Rolle bei der Merkmalsintegration spielt. Für die Aufgabe war bloß wichtig zu wissen, ob Stimulus A dargeboten wurde oder fehlte. Ein zweites Ziel war die Klärung der Frage, ob sich die Merkmalsintegration nach dem endgültigen Verschwinden von Stimulus A noch auf eine nachfolgende Handlung auswirkt. Gemäß der Theorie von Kahneman et al. (1992), sollte sich die Merkmalsintegration nicht sofort nach dem Verschwinden des Objektes auflösen.

Die Analyse der Daten ergab, daß gemäß der generalisierten Kodebesetzungshypothese die Reaktion B im Falle einer Merkmalsüberlappend mit der Position des Go-Signals signifikant langsamer war. Das heißt, daß die Objekteigenschaften des Go-Signals zeitlich integriert wurden, obwohl sie nicht relevant waren. Ein aktives Behalten ist also nicht notwendig für die Integration der Merkmale in eine zeitliche Repräsentation.

In Experiment 7 wurde die Dynamik der Merkmalsintegration untersucht. Wie in der Diskussion von Experiment 3 behauptet, werden Handlungsmale zunächst aktiviert und erst danach integriert. Nun sollte untersucht werden, ob das Gleiche für Objektmerkmale gilt und ob aktivierte, jedoch nicht integrierte Objektmerkmale eine bahnende Wirkung auf die Handlungsvorbereitung haben.

behalten werden, denn er blieb während und nach der Aufgabe B auf dem Bildschirm sichtbar. Zum anderen erschien Stimulus A nur 100 ms vor Stimulus B, während das SOA in der Bedingung "Objekt-behalten" 2000 ms betrug.


In Kapitel 5 werden die wichtigsten Schlußfolgerungen zusammengefaßt. Anschließend werden zwei weitere theoretische Implikationen der Ergebnissen besprochen. Erstens wird die Bedeutung der Tatsache, daß die Handlungsplanung von einem Bindungsproblem beeinflußt wird, für die Handlungsplanungstheorien besprochen. Es wird kritisiert, daß die existierenden von der Motorprogrammierung ausgehenden Theorien nicht den Einfluß der Bindungsproblematik umfassen, weil sie von vorn herein ausschließen, daß der Mensch gleichzeitig mehrere Handlungspläne
References


Lebenslauf