

## Dimensions of Coordination

§Peter Bøgh Andersen, \*Peter H. Carstensen, §Morten Nielsen

§Center for Human Machine Interaction, Aarhus University, Denmark  
{pba, [mnielsen](mailto:mnielsen)}@imv.au.dk, \*The IT University of Copenhagen,  
Denmark, [carstensen@it-c.dk](mailto:carstensen@it-c.dk)

### Abstract

Cooperative work is coordinated by means of many different media and codes: by means of oral or written language, by means of diagrams and maps, and by means of dedicated computer systems. The purpose of the paper is to set up a small set of dimensions that can be used to characterize and compare the different ways in which coordination is achieved.

The paper focuses on the relation between oral and artifact-based coordination. The former is often replaced by the latter when coordination complexity exceeds the limits of oral coordination, but the artifacts must still connect to the oral communication surrounding it. Therefore it is relevant to investigate the similarities and differences between the two distinct means of coordination, and to look for a general framework, encompassing both means of coordination that enables us to understand how they may work together. We set off from two field studies: One of oral coordination as performed aboard the world's largest container carrier and one of a coordination artifact used in a software development project. We conclude by proposing five dimensions for classifying media and codes for cooperation. One of them relates to the medium: *Persistency*. Two are semantic, since they relate to the *reference* of the representations: *Aktionsart (Process vs. State)*, and *Theme (Field of Work vs. Work Arrangement)*. The last one is pragmatic since it concerns the use of the representations and their interaction with the environment: *Locus of Agency (User, System and Environment)*.

### 1. Introduction

The coordination of complex cooperative work is an intricate matter that can impose a severe workload on the cooperating actors—so much so, in fact, that the magnitude of coordination work can become a general obstruction to work effec-

tiveness, flexibility, or even safety. In such cases computer-based coordination support systems are often engaged as a means of reducing the degree of complexity involved in coordinating the complex cooperative work.

There are many studies illustrating the rich and seamless nature of coordination performed by co-located actors (e.g., Harper et al., 1991; Heath et al., 1993). All these studies show that the coordination of cooperative work can sometimes be achieved effectively and effortlessly simply by means of our everyday modes of interaction. Other studies have shown however, that oral interaction becomes increasingly labor intensive and sometimes even inadequate as a means of coordination in high complexity cooperative work (Carstensen and Sørensen, 1996; Schmidt and Simone, 1996). The latter studies highlight the need for artifacts that support the actors in coordinating the complexly interdependent cooperative tasks which, when carried out by means of oral interaction, tend to generate a very high coordination workload. When designing computer systems with the objective of reducing the magnitude of coordination work, it will therefore often be necessary to 'transform' coordination work from the medium of oral interaction to artifact based interaction—this transformation is neither trivial nor well understood.

The paper investigates the relation between spoken and artifact based coordination. We report on two field studies: a study of maritime operations aboard the worlds largest container ship where coordination is primarily performed by means of oral interaction, and a study of software engineering where the coordination relied heavily on artifacts. We will analyze the means of coordination used within the two work settings, and we will propose a number of dimensions along which oral and artifact based coordination can be characterized and compared in order to elicit similarities and differences.

## **2. Communication as coordination**

The maritime operations with which we will be concerned in the following were observed during the most recent in a series of field studies concerned with time-critical cooperation and coordination as performed by the crew on board some of the world's largest container carriers. We have spent a total of four months on board three container carriers calling on 35 harbors on four continents. The field work has generated a rich data material comprising a large number of video recordings, interviews, and technical documentation on the vessels in general and the instruments on the bridge in particular.

The data presented below stem from a European roundtrip performed on board the carrier M/S Sally Mærsk during spring 1999. The fragments of communication that will be analyzed in the paper have been transcribed from video recordings.



Fig. 1. Sally Mærsk approaching the harbor of Algeciras

## 2.1 The need for coordination in maritime operations

Maritime operations are characterized by significant variations in the need for coordination. While operating in open waters the vessel is steered by one officer who will perform most tasks in solitude; he is alone on the bridge most of the time, and ship-to-ship and ship-to-shore communication is infrequent. However, when the vessel approaches a harbor and enters coastal waters, operations become highly cooperative. When operating in and around the harbor the master (captain), chief officer, helmsman, and pilot are working on the bridge; two groups of three and four actors—commanded by the first and second officer, respectively—are positioned on deck; the pilot station, vessel traffic management service, and dockers are located ashore and communicate via VHF; tugboats and other vessels operate in the immediate surroundings of the Sally Mærsk and will at times become part of the cooperative work arrangement involved in maintaining safe operations.

Cooperation grows in line with rises in work complexity. When the vessel moves from open waters through coastal waters on to the waters in and around the harbor, work constraints become increasingly tight. Basically, maneuvers have to be performed with an increasing degree of precision, while there is less time to make the navigational decisions.

The tighter work constraints are reflected in the ways the actors communicate and coordinate. In and around the harbor area—where work complexity peaks—communication is almost exclusively about work and proceeds according to highly structured patterns of interaction. It is this type of coordination that we shall be concerned with in the following. The theoretical basis for the analysis will be drawn from empirical and theoretical work on work communication (Andersen, 1997). We shall characterize instances of work related communication by means of the key concepts, *focus*, *background*, and *protocol*.

## 2.2 Focus and background

Consider the following situation. The Sally Mærsk is inbound for Rotterdam. There are four actors on the bridge, master, chief officer, pilot and helmsman. The auto pilot has been discharged and steering is performed manually. All are silent, then the pilot speaks: *starboard twenty*. The helmsman speaks: *starboard twenty*. And all are silent again.

1	Pilot	Starboard twenty
2	Helmsman	Starboard twenty

Fragment 1: Rudder command

In any communicative situation, some features are taken for granted, whereas others are the topic of discussion. We shall say that the former constitute the *background*, the latter the *focus* of communication<sup>1</sup>. Background information is either not expressed at all, referred to via unstressed pronouns or definite nouns that recur identically during the conversation, or referred to via certain backgrounding constructions, such as adnominal adjectives and clauses.

In the communication between the pilot and the helmsman, the background information is completely left out. The pilot does not state who is the receiver of his command (the helmsman) nor which action is to be done (turning the wheel); he does, in fact, not state that it is a command at all.

Particular combinations of backgrounded and focused information types define specific communicative functions. The communicative functions can be defined in a very precise manner by considering a typical sentential schema used to interpret actual utterances<sup>2</sup>.

---

1 Focus and background is well-known concepts in linguistics. They describe the information structure of the sentence, cf. the *textual metafunctions* in Halliday (1994). Bødker (1996) uses the notion of focus as a general tool of work analysis.

2 The approach is mainly inspired by Halliday (1994). According to Halliday, utterances must simultaneously fulfill three main types of functions: *ideational* (how should we structure the topic?), *interpersonal* (how do we interact?), and *textual* (how do we make the text cohere?), cf. note 1. The schema tries to capture the combination of functional features that are relevant in the data. *Ideational* functions: the schema only covers *Doings*, i.e. material processes implying a change of state. The structure *Subject (Actor) + verb + Object (Goal)* derives from the process type of *Doings*. Halliday's five other process types are not covered. The adjuncts *Manner*, *Time* and *Place* are added since they turned out to be important in the data. *Interpersonal* functions: A slot for *tense*, *aspect* and *modality* is included since we are concerned with co-operation, i.e. regulation of interpersonal relations. *Textual* functions: the distinction between *background* and *focus* is a textual function. It is included in order to describe the reduction of complexity so important in complex work settings. Thus, the schema is an assemblage of features that are expected to be important in cooperative, physical work of the type we are dealing with. The schema may not be able to extract important features of other types of work, e.g. work mainly consisting of communicating – “sayings” in Halliday's terms.

## Dimensions of Coordination

<i>Subject</i>	<i>Tense Aspect Modality</i>	<i>Verb</i>	<i>Object</i>	<i>Manner</i>	<i>Time, place</i>
Someone	do	acting on	Something	in some way	sometime

Schema A. The basic sentential schema.

A sentential schema consists of a set of slots following one another in a more or less fixed sequence. Each slot can be filled with a particular kind of linguistic material, e.g. nouns, verbs or adverbials. Such sets have been called *paradigms* since Saussure's days. In the following diagrams we shall shorten "Tense, aspect, modality" to TAM.

Below a set of communicative functions are defined by different combinations of background and focus (boldface) paradigms (the data is from the Swedish Postal Giro, cf. Holmqvist 1989; Andersen 1997, pp. 379 ff.):

<b>Someone</b>	should begin	acting on	Something	in some way	<b>sometime</b>
----------------	--------------	-----------	-----------	-------------	-----------------

Schema 1. Work distribution: *who* should begin doing it *when*?

Someone	<b>is</b>	acting on	Something	in some way	now
---------	-----------	-----------	-----------	-------------	-----

Schema 2. Supervision: *is* she doing it?

<b>Someone</b>	<b>has</b>	acted on	Something	in some way	now
----------------	------------	----------	-----------	-------------	-----

Schema 3. Worker Reporting: *has she* done it?

Someone	<b>has</b>	acted on	<b>Something</b>	in some way	now
---------	------------	----------	------------------	-------------	-----

Schema 4. Object Reporting: *has* it been done to *it*?

I	should	<b>act on</b>	<b>Something</b>	in some way	now
---	--------	---------------	------------------	-------------	-----

Schema 5. Work coordination: *what* should I do now?

She	should	act on	Something	<b>in some way</b>	sometime
-----	--------	--------	-----------	--------------------	----------

Schema 6. Instruction: *how* should she do it?

Someone	<b>has</b>	acted on	Something	<b>in the correct way</b>	sometime
---------	------------	----------	-----------	---------------------------	----------

Schema 7. Control check: *has* it been done to *it in the correct way*.

If we consider the rudder command communicated between the pilot and the helmsman in terms of the basic sentential schema, only the *manner* slot is focused (Schema 8) the rest of the slots are backgrounded. Thus it belong to the instruction type.

<i>Subject</i>	<i>TAM</i>	<i>Verb</i>	<i>Object</i>	<i>Manner</i>	<i>Time, place</i>
Helmsman	should	turn	the wheel	<i>Starboard</i> <i>twenty</i>	Now, at the helm stand

Schema 8: Instruction. Helmsman should turn the wheel *in some manner* now at the helmstand

The purpose of using background and focus paradigms is to reduce complexity. If the utterance is to specify an action that can be performed, then all paradigms must contain exactly one member that connects to work operations. One cannot make the ship sail “some” course. The course must be “001°, 002°, 003°, etc.”. So the purpose of the conversations is to fix all focus paradigms to exactly one of these. But if we can choose between, say 10 items in each slot, there are 10<sup>6</sup> possibilities to choose from, and we will never get around to do anything before the ship is grounded. Formally speaking, the technique of backgrounding reduces complexity exponentially.

The oral coordination in our data illustrates the importance of flexibility in the distribution between foreground and background paradigms. Paradigms, initially in the background during a conversation, may suddenly be brought to focus. In fragment 2 the shift from background to focus is performed by the master as he realizes that the pilot is in the midst of a dangerous misunderstanding.

1	S.Atlantic	Sally Mærsk, the Sealand Atlantic [VHF]
2	Pilot	Sally Mærsk [VHF]
3	S.Atlantic	Yeah, good afternoon Captain, are you <b>turning to port</b> now, are you, over ? [VHF]
4	Pilot	Yes, I’m turning <b>slowly to port</b> , yes [VHF]
5	S.Atlantic	Okay, we, we are, we will be steering our course of about two nine zero, and we will stay to the north of you, if that is agreeable with you [VHF]
6	Pilot	yeah fine, I will be following the deep draft route outside [VHF]
7	S.Atlantic	Yeah, and can you give us <b>a red to red</b> passing, please, <b>port to port</b> [VHF]
8	Pilot	<b>port to port</b> , yeah fine okay [VHF, talking to the Sealand Atlantic]

Fragment 2. Focus on parameters.

The fragment is from an approach to the Harbor of Rotterdam where the Sally Mærsk has been forced to perform a 360° turn in the deep-draft fairway some five miles from the harbor entrance. The maneuver is critical because no vessels can pass her while she is sideways in the canal. Traffic conditions are cleared prior to the maneuver. Yet, just after having initiated the turn, the Sally Mærsk is contacted

on the VHF by the outbound Sealand Atlantic that will meet with Sally Mærsk in the narrow canal.

The assumed background action is “passing head-on” and the focus paradigm contains two members, *red to red* and *red to green*, from which they have to choose. Sealand Atlantic chooses *red to red* because this is the normal choice. However, since Sally is turning port, the action will not be “passing head-on” but “overtaking” where *red to red* is a wrong choice.

Sealand Atlantic knows that Sally is turning port, but has not realized the consequences. The pilot does not realize the error either, but the captain does. He begins to focus the modality paradigm (how *can* we) and the previously backgrounded action slot (he will *overtake* us probably) and the Pilot quickly realizes his error (*not possible*) — although he does not admit to it!

1	S.Atlantic	Standing by zero two one six[VHF]
2	Pilot	okay [VHF]
3	Master	who was that?
4	Pilot	the outgoing ship, Sealand Atlantic, she want to pass <b>port to port</b>
7	Master	How, <b>how can</b> we do that? [pass the Sealand Atlantic port to port]
8	C. officer	(...)
9	1. officer	Yes
10	Pilot	We are steering around slowly, slowly
11	Pause	
12	1. officer	I presume he means that he will ...
13	Master	Yeah, but <b>how can</b> he ... (...), he <b>will overtake</b> us probably ....
14	Pilot	Yeah, I don't know
15	Master	...yeah, no, so um, <b>port to port</b>
16	Pilot	Sealand Atlantic, the Sally Mærsk [VHF]
17	S.Atlantic	Sally Mærsk, Sealand Atlantic [VHF]
18	Pilot	Um, you want to pass us on our <b>starboard</b> side, on the <b>north</b> side? [VHF]
19	S.Atlantic	Roger, Ill like to ...I, I heard you were <b>turning</b> to your <b>port</b> to go back in, I (...) in with the drencher, otherwise we are going <b>red to red</b> , over [VHF]
20	Pilot	I think <b>red to green</b> with us [VHF]
23	PilotSt	Sealand Atlantic ,Pilot Maas [VHF]
24	S.Atlantic	Pilot Maas [VHF]
25	PilotSt	The, the Sally Mærsk is <b>turning</b> to <b>port</b> , so you can proceed her north of her, over [VHF]
26	S.Atlantic	she is gonna <b>turn to port</b> , okay thank you, thank you, Sally Mærsk [VHF] [notice that the Sealand Atlantic thinks they are talking to the Sally Mærsk—when they are in fact talking to the Pilot Station Maas Approach].
27	Pilot	yeah (...) <b>port to port</b> , <b>red to red not possible</b>
28	Master	that is <b>not possible</b>
29	Pilot	his <b>red</b> side to our <b>green</b> side

Fragment 3. Shift of focus from parameters to modality.

Fragment 3 is a good example of how important it is that speakers are able to re-focus what was previously a background paradigm. On the one hand, it is necessary to reduce complexity by backgrounding information, but it is risky too if the backgrounded information is not valid.

### 2.3 Protocols

As is well-known, communication is realized as a sequence of utterances governed by certain turn-taking rules of the form: Questions are followed by answers, commands are repeated (fragment 1), etc. However, as in the case of oral commands other events than communicative ones are governed by conventions; physical events are governed too.

Ronald Stamper and his collaborators (e.g. Chong & Liu 2000) claim that such rules — or *norms*, as they call it — are a vital ingredient in the proper functioning of work organizations. They set up two main types of norms, *behavioral* norms and *intrinsic* norms, comprising *evaluative*, *perceptual*, and *cognitive* norms. Course commands (like fragment 1) are examples of behavioral norms that have the form:

- If <condition> then <agent> is <obliged|permitted|forbidden> to perform <action|speech act>

Intrinsic norms like ‘If someone hinders me doing something (e.g. berthing) then I *expect* him to notify me when the coast is clear (the berth is free)’ has the following general form:

- If <condition> then <agent> adopts <an attitude> towards <some consequences or proposition>

Norms in our data are *heterogeneous* since general and specific norms are mixed; they are *semi-conscious* since some of them are written down and others are verbalized when they are broken; and they are not designed as a system, but have *evolved* through history.

1	Master	And call out when you are on the new course, K, right (...), when it's there you say one one five – then we know it's there
2	Helmsman	I did say so last time
3	Master	Well, I just did not hear it
4	Helmsman	(..)
5	Master	okay (...) that's fine – keep up the good work

Fragment 4: Violating a rule.

The existence of norms in oral coordination of maritime operations can be verified in four ways: they can be abstracted from regularities in the data, they are taught explicitly during training; and some norms occur in the ship-owners manuals; and they are verbalized when they are violated:

Fragment 4 also indicates another important aspect of the observed norms, namely that they are self referential: The master *tells* the helmsman to *call* out.

## 2.4 Characteristics of oral coordination

The examples given above can be summarized into the following observations regarding oral communication and coordination:

- In oral communication, information is distributed in *background* and *focus* paradigms. This enables the actors to reduce complexity of their interaction.
- Coordination communication consists in closing the focus paradigms so that one executable member remains.
- Communicative functions can be defined by *combinations* of background and focus paradigms.
- Oral communication allows *smooth changes* between focus and background paradigms, enabling speakers to question tacit assumptions.
- Norms are *semi-conscious*, i.e., normally not conscious, but can be verbalized and discussed in case of violations. Formulations can also be found in maritime education and in ship-owner's manuals.
- Norms do *not form a system*, but are a set of concurrently working, heterogeneous, and relatively autonomous guidelines that have evolved through time.
- Protocols can communicate about themselves (are *self-referential*).

A special subset of norms are those concerned with regulating cooperation between two or more persons. We shall use the term *protocol* to denote a coherent set of norms of this type. All the preceding examples of norms belong to a protocol.

## 3. Artifact based coordination

In the following we will turn to a field study of artifact based coordination conducted at Foss Electric, a Danish manufacturing company developing, producing, and marketing equipment for e.g., measuring the compositional quality of milk.

The study to be discussed here concentrated on the software development in one large project: The S4000 project. It was a highly complex design project. It included facilities earlier provided by several instruments, measurements of new parameters in the milk were included, speed was to be improved significantly, and it was the first product with an Intel-based 486 PC built-in.

The software complex contained more than 200.000 lines of C-code and was organized in approximately 25 modules distributed in 15 different application. The S-4000 project involved more than 50 different people and lasted approximately 2 1/2 years. During the last 18 months the software design group had a stable size of approximately 8-10 designers, each having rather clear roles related to the design and implementation of one or more specific modules.

The most important roles related to the testing and correction activities included: (1) Software designers responsible for designing, implementing, maintaining, and correcting bugs in one or more of the software modules; (2) A group of three software designers called the *Spec-team* responsible for diagnosing reported bugs and deciding how to handle each of the bugs; (3) A Platform master responsible for managing and coordinating all the activities involved in integrating the outcome of one working period (called a “platform period”). He was, among other things, responsible for verifying the corrections of the software made by the designers, i.e., control that the reported bugs had been dealt with; (4) A Project plan manager responsible for maintaining a project plan spreadsheet; (5) Testers testing of the software embedded in the S4000 instrument; And (6) the central bugs file Manager organizing and maintaining the central bug file, a binder containing copies of all reported bugs and organized according to their status. It is interesting to notice, that there was no software group manager during this one and a half year period.

Early in the S4000-project, the software designers realized problems in coordinating, controlling, monitoring, and handling the testing activities. As a counter measure to these problems the software engineers invented and used a standardized bug form (figure 2) that all testers had to fill in whenever they identified an error (a bug). Subsequently, the findings from the study of the paper-based bug-report form formed the basis for a illustrative prototype implementing a computer-based bug-report form. In the following we will present the paper-based bug-report form and the subsequent computer-based bug-report form as cases of coordinating artifacts.

### **3.1 The paper-based bug-report form**

The development and correction work was organized in phases called “platform periods”. A platform period was typically 3–6 weeks work followed by one week of integration. All the work and the plans were structured in relation to these periods. For each period a designer was appointed Platform Master responsible for collecting all information on updates and changes made to the software, and for ensuring that software was tested and corrected properly.

When a bug was identified, the tester filled in a form and sent it to the spec-team. The spec-team diagnosed the problem and decided which developer should fix the problem. The responsible designer was notified (by receiving a bug form),

and estimated the correction time needed. When the problem was dealt with, the designer notified the Platform Master who could then verify the corrections.

At any state, the binder contained a copy of the form in its current status. The binder had seven entries reflecting the status of a specific bug: (1) Non-corrected catastrophic bugs (copies); (2) non-corrected essential bugs (copies); (3) non-corrected cosmetic bugs (copies); (4) postponed bugs (originals), (5) rejected bugs (originals); (6) corrected bugs not yet verified (copies); and (7) corrected bugs (originals). For each of the seven categories were the forms filed chronologically. The entries played a central role in stipulating the coordination by providing all involved designers and testers access to the state of affairs in the testing.

Initials: (1) Date: (1)	Instrument:	Report no: (2)	<p>The actors fill (or add information) in:</p> <p>The testers: (1), (2), (3), and (4) The Spec-team: (3), (4), (5), and (7) The designers: (6) and (8)</p> <p>The procedure for handling bugs:</p> <ul style="list-style-type: none"> <li>•A tester register and classifies a bug (field 1,2,3, and 4)</li> <li>•The tester sends the form to the spec-team</li> <li>•The spec-team diagnose and classify the bug (field 3, 4 and 7)</li> <li>•The spec-team identifies the responsible designer (field 5)</li> <li>•The spec-team estimates the correction time (field 5)</li> <li>•The spec-team incorporates the correction work in the work plans</li> <li>•The spec-team requests the designer to correct the problem</li> <li>•The designer corrects the bug and fills in additional correction information (field 6 and 8)</li> <li>•The designer sends the form to the central file</li> <li>•The CFM sends the form to the PM and insert copy in central file</li> <li>•The PM verifies the correction</li> <li>•The PM returns the form to the central file</li> </ul>
Description: (3)			
Classification: (4) 1) Catastrophic    2) Essential    3) Cosmetic			
Involved modules: (5) Responsible designer:                      Estimated time:			
Date of change:              Time spend:              Tested date: (6) <input type="checkbox"/> Periodic error - presumed corrected			
Accepted by:                      Date: (7) To be: 1) Rejected    2) Postponed    3) Accepted Software classification (1-5): ____ Platform:			
Description of corrections: (8)			
Modified applications:  Modified files:			

Fig. 2. A paper-based bug-report form annotated with the procedure followed when using it. CFM is central file manager and PM is Platform Master. The form is a sheet of A4 paper printed on both sides. The figure illustrates who fill in the information in the form.

The work plans were organized in a large spread-sheet containing information on: which tasks are to be accomplished and a reference to a detailed description of the task, the estimated amount of labor-time per module for each task, the responsibil-

ity-relations between modules and software designers, the relationships between the tasks and platform periods, and the total planned work hours per platform period for each software designer. The work plans were maintained by the Platform Master and a member of the spec-team.

The usage of the forms and the procedures can also be viewed as information flows between the involved roles and actors. Figure 3 illustrates the flow between the six different roles involved in coordinating the software testing and correction activities. In many situations the involved actors undertook more than one role.

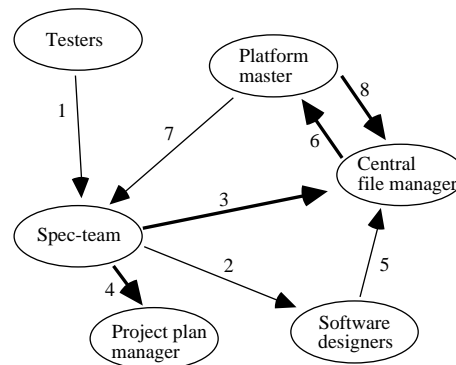


Fig. 3: A visualization of the roles involved in the software testing of the S4000 project, and the information flow between them.

The information flow described in figure 3 concerns only the stipulated (through organizational procedures) flow of the bug form mechanism. The thick arrows (3, 4, 6, and 8) indicate, that the flow is often a bunch of forms sent. The thin arrows indicate, that the forms are typically sent one at the time. Other types of information are frequently exchanged between the actors and there might be situations where one of the actors choose not to follow the stipulated information flow.

### 3.2 The computer-based bug-report form

BRaHS<sup>3</sup>—the computer-based bug-report form—was designed in order to illustrate certain principles for computer-based coordination support which had been derived from the study of the paper-based bug-report form. The design of BRaHS focused on illustrating how the registration of bugs (including an improved classification of bugs) could be handled, how overall requirements for flexibility (of the flow protocol and the structure of the form) could be reflected in the user interface, and how a protocol can be made visible and accessible to the users of the system. BRaHS was designed to support distributed registration of bugs, and automatic routing (forwarding and passing on) of report forms to the next actor (role) in the

<sup>3</sup> Bug reporting and Handling System, designed and implemented by Thomas Albert and Peter Carstensen. See Carstensen (1996) for a full presentation of the BRaHS system.

*tester spec-team designer Platform Master chain.* BRaHS implemented a changeable workflow protocol, and provided facilities enhancing the overview of bugs, the status, etc. To fulfill the requirements, the following overall functionality was designed and implemented in BRaHS:

- Log-in procedure where the user identifies himself to the system and specifies the role(s) he assumes when interacting with the system.
- Facilities for registering information about bugs. Both the testers, the spec-team, and the designers enter information about a bug.
- Facilities for specifying the classification of a bug.
- Procedures that automatically routes the form to the next actor in the workflow when a user indicates that his registration is finished.
- A graphical representation of the workflow protocol, and access for the users to make certain changes to the protocol.
- Facilities to search for bugs associated with certain characteristics, and facilities for browsing information on registered bugs.
- Procedures to select the current project, and to define or change the actors, modules, roles, instruments, etc. involved in a particular project.

BRaHS was implemented as a client/server structure using Borland's Delphi as the application development and runtime environment. The system runs on a Windows platform.

### **3.3 Characterization of artifact based coordination**

It can, of course, be questioned whether the bug form mechanism has genuinely “eliminated” coordination complexity. The coordination to be conducted is as complex as always, but it can appear simpler to the actors through improving the representations of the work domain, by enforcing a specific behavior of the actors involved, by “automating” certain activities of the coordination, and by establishing a division of labor minimizing the need for coordination. The claim here is, that the bug form mechanism supported the coordination of software testing and correction by providing several of these things:

- The bug form provided a standardized information structure by means of which all bugs were described. By allowing the diagnosis information to be included in the form in a standardized pre-structured manner, the mechanism made it easier for the spec-team to find the relevant information. The classification of the bugs made it easier for the spec-team to deduce the testers’ perception of the problem reported. The form can thus be seen as improving the representation of the field of work (the bugs) by establishing a “common standardized language”. This makes it easier for the actors to interact. “Standard-

ized languages” are, of course, problematic too. They constrain the actors, it takes time for the actors to become familiar with them, and they need to be maintained.

- The standardized format of the form, furthermore, supported the work of reporting, both from tester to spec-team, from spec-team to designer, and from designer to platform master. This is because it forces a specific behavior on the actors. Through a specific surface representation the form freed the actors from considering which information to include.
- The bug form mechanism also supported the coordination activities by stipulating the work flow for handling the reporting, diagnosing, and correcting process. Although it was not completely automated, the pre-specified flow minimized the need for communication and interaction among the actors when handing over the form (and thus the obligations) from one actor to the next. The pre-specified flow (the embedded protocol) afforded support through constraining the actor: He could just apply the pre-specified routing without further considerations.
- Finally, the bug form mechanism was a central tool for establishing a well-understood and well-defined division of labor. By establishing different roles, and very clearly defining their responsibilities, the mechanism reduced the need for communication and interaction among the actors. All actors in the reporting, diagnosing, and correcting process knew exactly, what their obligations were. When they had dealt with their part of the treatment, they could just pass on the form, and others would take care of the rest. Their need for coordination was reduced.

Coordination activities, like monitoring the state of affairs and negotiating classifications, allocations, etc., were supported by the bug form mechanism too. The establishment of the central file (the binder), including all registered bugs and their current status, made it easier to get a coherent picture of the state of affairs. Although the testers and designers found it difficult to achieve an overview from the content of the binder, the benefit was that they only had to search in one place. Furthermore, the standardized information structure in the forms, and the standardized index of the binder, made it easier for the actors to find the relevant information, for example the classification and status of a bug or the number of not-yet-corrected category 2 bugs. Finally, the bug form mechanism, and its related mechanisms made classifications and resource allocations visible and accessible to the actors. The classifications and allocations became easier to discuss.

#### **4. Dimensions of coordination means**

If we compare the coordination *functions* performed orally in maritime operations to those being stipulated by the paper- and computer-based bug-report forms we

find a high degree of equivalence. Parts of the bug-report can be analyzed as a medium for conducting a sequence of coordinative communicative functions with different focus and background paradigms. In addition, the focus paradigms in one function becomes background in the following. As shown in Schema 9, the verb is always backgrounded to *correct*; (1) focuses on the module identification, (2) on modality (*should/should not be corrected (rejected/accepted)*), time (*now/postponed*) and manner (classification of bug: *how should the correction be done?*), and (3) focuses on the subject (*who is going to do it*). Finally, (4) is concerned with aspect (*has it been done*) and manner (*has it been done in the proper way*).

<i>Subject</i>	<i>TAM</i>	<i>Verb</i>	<i>Object</i>	<i>Manner</i>	<i>Time, place</i>
1. Someone	should	correct	<b>which module</b>	in some way	sometime
2. Someone	<b>should/should n't</b>	correct	the module	<b>in which way</b>	<b>when</b>
<b>3. Who</b>	should	correct	the module	in that way	at that time
4. Someone	<b>has</b>	<b>corrected</b>	the module	<b>in the proper manner</b>	at some time

Schema 9. Coordinative functions in the bug-report in terms of foreground / background focus paradigms

If we compare Schema 9 to the different combinations presented in Section 2.2 we can see that (3) is Work Distribution (Schema 1), (2) is a variant of Instruction (Schema 6) where modality is focused too (*should/ shouldn't be done*), and (4) is Control Check where tense/aspect (*has it been done?*) and manner (here “correctly”) are focused.

The first function (1) is a new one (compared to those introduced in Section 4.1) where only the work-object is focused. Also the “seriousness-classification” (catastrophic, essential, cosmetic) lack analogs in the presented oral coordination, although it resembles the Work Priority Ordering defined in Andersen (1997).

We can conclude that communicative functions found in oral communication recur in coordination mechanisms. The paper-based bug-report collects a set of communicative functions that are normally applied after each other according to the oral protocol, and their sequence on the form to some degree stipulates their temporal sequence. For example, the Control Check in (4) will normally be performed after the Work Distribution in (3).

In our examples, a main difference between oral and artifact based coordination is that the present state of the field of work, its history and possible futures are made persistent and thus publicly accessible. Whereas the software developers at any time can inspect the forms to ascertain how far work has progressed, what has already been done, and what might probably happen in the future, the situation awareness of the maritime officers must continually be reproduced and updated. A

further difference is that the computer-based bug-report form is an active means of coordination—it performs automatic routing—whereas all routing is performed by the actors in the cases of oral and paper-based coordination.

In the following we will propose a set of analytical dimensions relevant for a more general and structured comparison between different means of coordination. Before we do this, however, we shall describe a computerized means of coordination from the maritime domain. The example (Fig. 4) is from the VTS station at Bremerhafen. It is used by the pilots to prevent ships passing each other at dangerous bends of the river Elb.

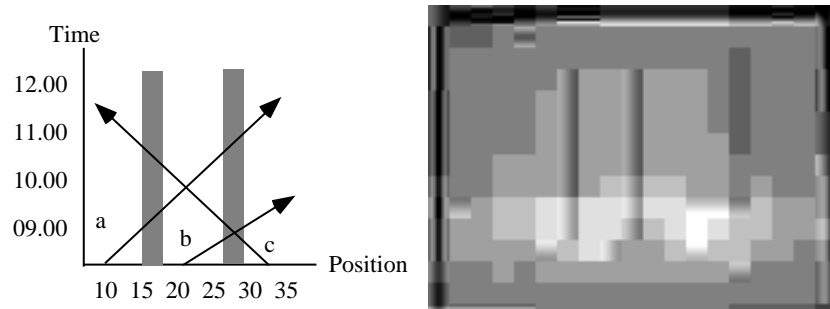


Fig. 4. Display used by the pilots at the Bremerhafen VTS station.

The y-axis represent time and the x-axis position along the river (in the actual systems, *tonnes* were used as units of measurements). The vectors represent the ship's future positions extrapolated by means of the present speed, and the shaded areas represent the sections of the river where encounters are prohibited. Vectors pointing left are ships sailing upstream, those to the right are downstream ships. The simple rule is that vectors crossing each other inside the shaded areas are prohibited, so *b* and *c* must be warned of each other, but no one else.

The example illustrates two things: (1) its reference is different from BRaHS in that it primarily refers to the field of work (the positions of the vessels) overlaid with representations of the cooperative work arrangement (the shaded sections denoting “two vessels may not be here simultaneously”). (2) The field of work changes the system in the sense that sensors automatically updates the vessel positions on the screen-image. The reason for this difference lies in the two domains: whereas the software system does not change except when the designers are active, in process control, the field of work changes autonomously without interference of the bridge personnel.

We now have described a set of means for enhancing cooperation: oral communication, the paper-based and the computer-based bug-report, checklists, and the Bremerhafen system. Although they share a common usage—enhancing cooperation within a common field of work—they are also quite dissimilar. How can we characterize their similarities and dissimilarities?

We have just touched upon an important *semantic* dimension, namely the *reference* of the representations: do they primarily refer to the work arrangement or to the common field of work? We shall call this distinction the *Theme* of the representation. The paper-based bug-report and the pilot-system primarily refer to the common field of work and presuppose cooperative protocols not explicitly described in the medium, whereas in the computer-based bug-report the work arrangement is also represented in the system. Oral communicative functions includes both varieties: in *Work Distribution*, *Supervision*, *Worker Reporting* and *Work Coordination*, the cooperative arrangement is focused, whereas the tasks and work objects are emphasized in *Object Reporting*, *Instruction*, and *Control Check*.

Thus, coordination may be achieved primarily by means of information about the field of work or primarily by reference to the cooperative work arrangement. Simple cooperative work can often be coordinated exclusively by reference to the state of the field of work. The coordination of complex cooperative work will however benefit from mechanisms which make direct reference to states or processes to be achieved within the cooperative work arrangement. Put another way: in complex cooperative work the actors will have difficulties coordinating the cooperative activities simply by observing the changing states in the common field of work. In order to reduce the coordination complexity they need means which explicitly stipulate the course of coordinative action.

Another important semantic dimension is that between references to *processes* and *states*, traditionally called *Aktionsart*. *Control Check* and parts of the bug-report refers to a state of the work-object, whereas oral *Instruction* and the pilot system (and, for that matter, maritime RADARs and VMS systems) refer to both states and processes in the field of work (the maritime systems primarily records states, but extrapolate future events and can make a history plot). The computer-based bug-report and the oral *Work Distribution* refer to processes in the work arrangement (protocols), whereas the oral *Worker Reporting* refers to a state in the work arrangement (*has she done it?*).

A third important dimension of the media is *persistency* that sets oral communication apart from all the other examples. By a *persistent* medium we mean a medium that maintains its information over longer periods of time. In *ephemeral* media, information is lost almost instantly and must be recorded elsewhere .

A more detailed classification is probably needed in reality: although it is true that oral communication is ephemeral, it can be recorded on tape, and this in fact done to VHF-communication in some ferries in order to provide legal documentation in case of accidents. In some cases, only parts of the representations are ephemeral, while others are persistent. For example, RADARs always offer a persistent representation of the present state which the crew can inspect at any time. However, it is possible to switch on a “history” facility so that the vessels’ posi-

---

A similar discrimination has been proposed in Schmidt and Simone (1996)

tions remain on the screen as a track of trailing dots. The decision as to which representations are to be persistent and which ones ephemeral is an important design issue (Andersen & May, to appear).

A fourth dimension relates to the *pragmatics* of the representations, i.e. their use and interaction with the environment. We shall term the dimension *Locus of Agency*. In our cases there are examples of three loci of agency: in the *user*, in the *system* and in the *field of work*. In principle, all combinations are possible but we have only found a few. For example, in all examples, the user is a locus of agency. There are no fully automated systems.

In the pilot system, the *field of work* is an additional locus of agency (the screen is continually updated via a network of RADARs placed along the river), and this is definitely not the case in the BRaHS system that has no access whatsoever to the software system constructed. On the other hand, the BraHS *system* is itself a locus of agency: after a field in the form has been filled in the protocol actively suggests who should work on the form next, and the protocol also implements a default flow structure that ensures that the report is actually routed to the chosen actor: However, the user has still the final word. Finally, in the oral or the paper-based media, the *user* is the sole locus of agency, since the user himself must create the representations as well as execute the tasks denoted. The maritime computer systems are somewhere in-between: they only notify the user of possible conflicts of cooperation, but do not offer suggestions of possible actions. However, advisory collision avoidance systems are presently being designed where the maritime protocols—the rules of the road—are implemented.

## **5. Conclusion**

In this paper we have argued that there is a common set of concepts that apply to any means of coordination, irrespective of medium. These concepts include textual functions, such as focus and background, interpersonal functions, such as communicative function, norms and protocols, and ideational concepts, such as work arrangement versus field of work. We have furthermore suggested that a four-dimensional grid, *Persistency*, *Aktionsart*, *Theme*, and *Locus of Agency*, can be used for characterizing and comparing the examples we have detailed knowledge of.

We believe that this line of research is useful for two reasons:

- Means of cooperation often change medium in the course of time. For example, in the software example, similar functions were performed orally first, then by means of written forms, and, possibly, in the future, by means of a computerized coordination mechanism. Creating the latter often means changing, supplementing, or replacing the former.

- Computerized coordination mechanisms must still function together with oral and paper-based means of coordination. The designers will not stop talking to one other nor writing notes, even if they adopt a BraHS-like system.

In the first case, a shared set of concepts may make it easier to make a systematic change from expressing coordination in one portfolio of media to doing it in another one; in the latter case, a shared classification system may help designers evaluate how well a computerized coordination mechanism will work together with the other media surrounding it.

## 6. Acknowledgements

The research documented in this paper could not have been conducted without the invaluable help of numerous people at Mærsk Sealand and Foss Electrics. We would like to thank the companies for their generous hospitality and valuable comments. We thank Steven Verjans for useful comments. The research has partially been funded by the Center for Human-Machine Interaction (The Danish National Research Foundation). All errors naturally remain the responsibility of the authors.

## 7. References

- Andersen, Peter Bøgh: *A theory of computer semiotics. Semiotic approaches to construction and assessment of computer systems*, (Paperback, 1997), Cambridge University Press, Cambridge, 1997.
- Andersen, Peter Bøgh & Michael May.: Tearing up interfaces. To appear in “Organisational Semiotics” (ed. Kecheng Liu). Kluwer
- Bødker, S. (1996). "Understanding Computer Applications in Use — a Human Activity Analysis". In *Signs of Work*, eds. B. Holmqvist, P. Bøgh Andersen, H. Klein & R. Posner. Gruyter: Berlin, 1996, 325 – 348.
- Carstensen, Peter H.: *Computer Supported Coordination*, Writings in Computer Science (No. 61), Department of Computer Science, Roskilde University, Roskilde, Denmark, 1996.
- Carstensen, Peter H., and Carsten Sørensen: “From the social to the systematic. Mechanisms supporting coordination in design,” *Computer Supported Cooperative Work. The Journal of Collaborative Computing*, vol. 5, no. 4, 1996, pp. 387-413.
- Chong, S. & Kecheng Liu: "A semiotic approach to increase the design quality of agent-based information systems". In: Proc. of the 3rd Workshop of Organizational Semiotics, Univ. of Stafford, 2000.

- Halliday, M. A. K.: *Language as Social Semiotics. The Social Interpretation of Language and Meaning*, Edward Arnold, London, 1978.
- Halliday, M. A. K. *An introduction to functional grammar*. London: Edward Arnold, 1994.
- Harper, R. R., J. A. Hughes, and D. Z. Shapiro: "Harmonious Working and CSCW: Computer technology and air traffic control," in J. M. Bowers and S. D. Benford (eds.): *Studies in Computer Supported Cooperative Work. Theory, Practice and Design*, North-Holland, Amsterdam etc., 1991, pp. 225-234.
- Heath, Christian, Marina Jirotko, Paul Luff, and Jon Hindmarsh: "Unpacking Collaboration: The Interactional Organisation of Trading in a City Dealing Room," in G. De Michelis, C. Simone, and K. Schmidt (eds.): *ECSCW '93. Proceedings of the Third European Conference on Computer-Supported Cooperative Work, 13-17 September 1993, Milan, Italy*, Kluwer Academic Publishers, Dordrecht, 1993, pp. 155-170.
- Holmqvist, B.: "Work-language and perspective," *Scandinavian Journal of Information Systems*, vol. 1, no. 1, 1989.
- Schmidt, Kjeld, and Carla Simone: "Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design," *Computer Supported Cooperative Work. The Journal of Collaborative Computing*, vol. 5, no. 2-3, 1996, pp. 155-200.