

Applying Organizational Routines in Analyzing the Behavior of Organizations

Markus C. Becker

Strategic Organization Design Unit
Department of Marketing & Management
University of Southern Denmark
Campusvej 55 - DK-5230 Odense M - Denmark
mab@sam.sdu.dk

&

Francesco Zirpoli

Università di Salerno, Facoltà di Ingegneria
Dipartimento di Ingegneria Meccanica
I-84084 Fisciano (SA), Italy
fzirpoli@unisa.it

Abstract

The concept of organizational routine can foster our understanding of the behaviour of organizations and of organizational change (Nelson and Winter 1982, March and Simon 1958, Cyert and March 1963), but since empirical studies employing organizational routines as analytical perspective are still relatively rare, how to conduct such an analysis and what are its benefits is not yet fully evident. We wish to shed light on how employing routines contributes to understanding the behavior of organizations and to demonstrate the potential of such analysis. The empirical analysis of the product development process at an engineering centre shows that using organizational routines presents advantages over alternative analytical approaches. The paper also contributes to shed light on *how* to fruitfully employ an organizational routines perspective in analysing the behaviour of organizations, providing the foundation for further empirical work.

Keywords: Organizational routines, organization behaviour, innovation processes, new product development

JEL Classifications: O31, O32, M10, L23

Applying Organizational Routines in Analyzing the Behavior of Organizations

1. Introduction¹

Understanding the behavior and change of organizations was one of the key motivations in introducing the concept of organizational routines (Simon 1947, March and Simon 1958, Cyert and March 1963, Nelson and Winter 1982). More recently, organizational routines have also been posited as a useful focus in analyzing how work is carried out in organizations (Hutchins 1991, Pentland 1995, Orlitzki 2000, Barley and Kunda 2001, Pentland 2003a and 2003b) and in showing how organizations change their operations (Feldman 2000, Feldman and Pentland 2003, Winter and Szulanski 2001, Zollo and Winter 2002). Other theoretical frameworks and approaches to empirical analysis of organizational behavior and change do, of course, exist, so why choose organizational routines as analytical perspective?

To start with, routines are ubiquitous in organizations (see the empirical literature reviewed in Becker 2004), as well as an integral part of their daily operation. A large part of the work carried out in organizations is accomplished in routinized ways. Thus, its routines can be considered ‘typical’ for an organization. In order to understand an organization and its behavior, analysing its routines thus seems an appropriate starting point since they capture systematic and endogenous (rather than exogenous or one-off) performance drivers. Moreover, routines play an important role in organizational learning and memory, and contribute to efficiency induced by such learning (Argote and Epple 1990). Accordingly,

¹ Authors' names are in alphabetical order because they have contributed to the paper equally. For purposes of formal assignment, Markus Becker wrote section 2. Francesco Zirpoli wrote section 3. Section 1, 4, 5 and 6 were written jointly. We are grateful to Thorbjørn Knudsen, two anonymous referees, and participants of the 2005 Academy of Management Meetings and the 2nd International Routines Conference, Nice 2005, for helpful comments and discussion of preceding versions of this paper, and to Pasquale Salvatore for research assistance. All remaining errors and omissions are our own responsibility.

using organizational routines means to be able to capture knowledge- and learning-related aspects and their effects.

The potential of organizational routines as analytical perspective has not yet been fully unlocked. Empirical studies of organizations focused on routines are still relatively rare (for some examples, see Cohen and Bacdayan 1994, Pentland and Rueter 1994, Knott and McKelvey 1999, Edmondson et al. 2001, Feldman 2000 and 2003, Narduzzo et al. 2000, Szulanski and Winter 2002, Winter and Szulanski 2001). Furthermore, the question of how to operationalize organizational routines has turned out to be more than trivial, making the task more difficult (recently, a number of papers on the topic have been published; see Pentland 2003a and 2003b, Becker et al. 2005 and the articles in the special section of *Industrial and Corporate Change* 2005, 14.5). Finally, precisely what the benefits are from such an analysis is yet not fully evident, mainly due to a lack of empirical studies.

While previous empirical studies employing an organizational routines approach have tackled specific research questions about organizations, we explore how using the analytical perspective of organizational routines contributes to understanding the behaviour of organizations and show how to carry out such an analysis, thus closing a gap in the literature. The paper is structured as follows. It first briefly introduces the notion of organizational routines. The case study is presented in section three. Section four explores the contribution of organizational routines in explaining the behavior of organizations, thus answering the research question. Section five presents conclusions.

2. Organizational routines

Three definitions of organizational routines can be found in the literature: (i) behavior patterns (recurrent interaction patterns), (ii) rules (standard operating procedures, heuristics, etc.), and (iii) dispositions.

(i) Currently, most scholars think of organizational routines as repeated behavior patterns for accomplishing tasks. For example, consider the task of taking an order by phone, which is often accomplished by using a particular sequence of phrases in a phone conversation between a call center agent and a customer (Pentland and Rueter 1994). It has now become standard practice to use the term ‘routines’ for collective (multi-person) and the term ‘habits’ for individual (single-person) behavior patterns (Dosi et al. 2000). The term ‘recurrent interaction patterns’ provides a more precise term for referring to stability on the level of behavior that involves multiple actors.

(ii) Viewing organizational routines as rules (standard operating procedures), on the other hand, captures a different phenomenon. At least implicitly, rules and standard operating procedures (such as codified ‘best practices’ and process handbooks like McDonald’s processes for providing fast food; Leidner 1993) give rise to recurrent interaction patterns. Note that rules do not necessarily, however, fully specify the causal mechanism, that is, precisely how rules contribute to generating recurrent patterns of behavior. As Feldman and Pentland (2003) have recently argued, the role of human agency in rule-following, and probably also the governance mechanisms that provide incentives and constraints for following rules would at least need to be considered.

(iii) Some recent articles argue that organizational routines should be understood as *dispositions* to engage in previously adopted or acquired behavior, triggered by an appropriate stimulus or context (Hodgson and Knudsen 2004a and 2004b). Rather than patterns of behavior, routines are ‘stored behavioral capacities or capabilities. These capacities involve knowledge and memory. They involve organisational structures and individual habits which, when triggered, lead to sequential behaviors’ (Hodgson and Knudsen 2004a, 9). Routines are

therefore repertoires of potential behavior that can be triggered (Hodgson 2004, 652), such as the capacity to take and process an order from a customer in a call center.

In our analysis, we use these different definitions as follows. First, since our interest is in understanding the behavior of organizations, a description of the manifest behavior is required. Repeated behavior patterns, as in definition (i), are important because *processes* as they are actually carried out directly generate performance implications (Penrose 1959). For these reasons, ‘recurrent interaction patterns’ should be part of the analysis. Second, describing only the ‘surface’ level of manifest behavior does not, however, reveal the causes for certain behaviors in organizations. For understanding the behavior of organizations, capturing the causes obviously is important as well. Martha Feldman and Brian Pentland recently proposed a framework that draws together the two most commonly used definitions (routines-as-behaviour [recurrent interaction patterns] and routines-as-rules [standard operating procedures]). They distinguish the ‘ostensive aspect’ of a routine, referring to abstract, cognitive regularities and expectations that enable participants to guide, account for, and refer to specific performances of a routine, and its ‘performative aspect’, which refers to actual performances by specific people at specific times and in specific places. The standard operating procedures represent the ‘ostensive aspect’ of a routine, and the recurrent interaction patterns that implements them is the ‘performative aspect’. Importantly, between the two lies human agency. Human agents interpret, ignore, or adapt rules, for instance, and give a particular stamp to the recurrent behavior patterns to be observed. They also contribute to transforming organizational routines slowly by adapting the performative to the ostensive and vice versa. What concerns us here is that both levels of analysis are required because, as Feldman and Pentland argue, their interplay matters. The third definition, dispositions, is the most recent one. We are not aware of any empirical studies in the management literature that have begun to identify dispositions, and empirical methodology has not yet been discussed in

management research.² For this reason, and because of the arguments just described for focusing on the other two levels, we start our description there, and later consider the level of dispositions after having completed the description.

3. The case study

3.1 Method

Current literature shows that existing research does not fully address how organizational routines can be operationalized so that they can contribute to the understanding of organizational behaviour through empirical studies. In this respect, we intend to offer both an empirical and an analytical contribution. These goals, the nature of the research question and of the gaps we found in the literature led us to choose the case study methodology (Eisenhardt 1989, Pettigrew 1990, Yin 1994, Eisenhardt and Graebner 2007). Since we are interested in both rules and standard operating procedures, and recurrent interaction patterns, we need data-gathering methods able to capture both. While standard operating procedures and eventually the gap between the firm's realized and expected performances could be traced using documental and archival sources, in order to identify the recurrent interaction patterns, observation and in-depth interviews were the only possible options available. No method other than case studies can gather the qualitative data necessary to study both aspects of organizational routines.

The empirical material presented in this paper was collected at a European research centre specializing in automobile design. It was established as a 'green field' research centre close to a plant of a major European car maker in 1988 and today has around 750 employees, mostly

² There is, however, a stream of empirical research on dispositions in psychology. For a pointers to that literature see Knudsen (2008).

engineers. Our field study involved employees responsible for the development of vehicle engineering (in particular all the mechanical systems except for the style).

Three data collection methods were used: (1) Archival sources (about 2000 pages of company official documents [norms and procedures] were analyzed); (2) Extensive semi- structured interviews (the authors carried out more than 60 hours of interviews, both to explore and to get a deeper understanding of particular issues; these interviews involved the head of the vehicle department, the head of project management for vehicle department, the vehicle development function manager, the head of human resources, the chief engineer of ‘packaging’ (the ‘packaging’ task will be explained in the following section), one ‘packaging’ team leader, three ‘packaging’ team members and the head of ‘project SUV’; (3) one researcher spent 10 months working at the research centre; during this time, he regularly took part in meetings, participated in some of the tasks, provided support, and so on. In order to increase the consistency of our data, we decided to focus our interviews and direct observations on one project. For confidentiality, we cannot use the real acronym of the project, so we have named it ‘project SUV’. The project was the most important one the research centre was carrying out during the observation.

The method was designed to observe the same units of analysis from different angles, confirming the reliability of the data when they were consistent and inducing a new round of interviews when inconsistent. The comparison of quantitative and qualitative data, and the fact that managers from different departments and functions were interviewed using the same unit of analysis, made triangulation possible and extremely effective. Due to the intrinsic limitations of case study research (Miles and Huberman 1994), the sample could be partially biased by the fact that only one project has been observed, which could represent a limit to the

generalizability of the findings despite the fact managers interviewed confirmed that the project chosen was highly representative of their usual projects.

3.2 The task selected for observation: packaging

To analyze organizational routines, first, we must select a task in order to analyze how it is accomplished by the organization. Following from the definition of a routine, it should be a task that is recurrent, repeatedly accomplished in ‘the same’ (or very similar) way. For purposes of observation, it would also be helpful if the frequency would be high rather than low so that numerous observations can be collected in a reasonable time frame. Furthermore, the task should be closely linked to the organization’s objectives in order to avoid sampling outliers such as activities that take place ‘at work’ but are not primarily concerned with accomplishing the organization’s objectives.

Next, we must select the level of granularity at which to describe how tasks are accomplished, and this choice can present a problem. If the task chosen is too ‘large’ (macro, high-level), then the description will be very generic and complex, making it difficult to analyze and to draw conclusions as, for example, when one analyzes the entire process of achieving the principal task of the organization (such as ‘produce and market high-quality cars for the upper market segment’). If the task chosen is too ‘micro’, such as how to push keys on the keyboard while writing a marketing plan, the object of the analysis becomes meaningless, and the analysis itself might thus become meaningless as well. In either case, the risk is high that a detailed description of the process will be of little use in understanding how accomplishing the task will lead to a certain level of performance.

Fortunately, the concept of organizational routines provides guidance to the appropriate level of granularity as an intrinsic part of it. First, organizational routines refer to the collective level (such as group or team), not the individual level, which is covered by the

concept ‘habit’ (Dosi et al. 2000). The analyst should thus select tasks that require task-related interaction and exclude individual habits that do not contribute to accomplishing organizational tasks (at least not immediately). Second, the organizational routine concept provides guidance to which tasks are very important for the organization; the qualifier ‘organizational’ in ‘organizational routines’ indicates which routines are important with regard to this question. As is well-known, the organization literature considers interdependence as the key problem that organization addresses (Thompson 1967), and coordinating interdependencies is a fundamental task of every organization. If organizational routines capture the behavior of an organization, they should therefore also play a role in interdependencies (for instance, by providing coordination of how interdependent tasks are accomplished).

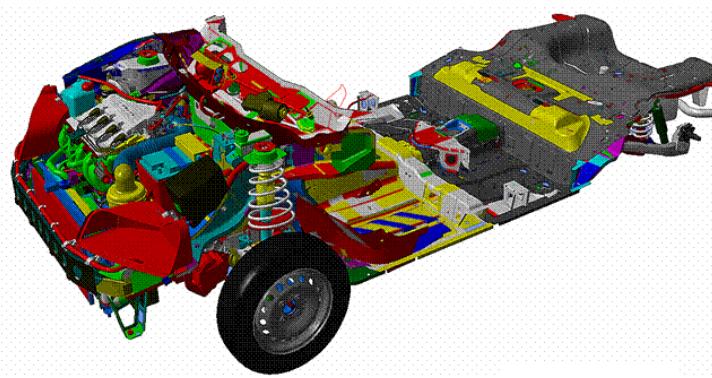
Different kinds of interdependencies between tasks can be distinguished. Thompson distinguishes generic, sequential, and reciprocal interdependence. Reciprocal interdependence is the strongest type, posing the most difficult coordination challenges, so focusing on routines an organization uses to accomplish a reciprocally interdependent task offers a very good chance of describing something that is meaningful to the organization. The criterion provided by the concept of organizational routine is thus to select *processes that implement reciprocally interdependent tasks*.

Given these criteria, we chose a task called ‘packaging’³. In the course of a new product development project, packaging refers to checking whether individual components actually fit together when assembled in their final positions. For example, imagine all the elements of an

³ As far as sampling is concerned, we were looking for an organizational task that was ‘...very special in the sense of allowing one to gain certain insights that other organizations would not be able to provide’ (Siggelkow 2007, p. 20). For this reason we employed theoretical sampling. This sampling approach reflects the fact that ‘cases are selected because they are particularly suitable for illuminating and extending relationships and logic among constructs’ (Eisenhardt and Graebner, p. 27) and that our research does not intend to perform theory testing (Eisenhardt and Graebner). As Eisenhardt and Graebner state ‘single-case research typically exploits opportunities to explore a significant phenomenon under rare or extreme circumstances’ (p. 27). We believe our sampling choice reflects this opportunity.

engine; packaging would check whether the proposed engine could actually fit in the available space under the hood (see figure 1).

Figure 1: Example of virtual packaging



In the packaging check, particular attention is on such details as minimal distances between components to assure they do not touch even at high speed or on rugged surfaces, resulting in noise and possibly damage of components. Packaging therefore contributes in a major way to a core objective of the organization and its product development activity: to assure a high quality standard for the product. Packaging also qualifies as a core task, its purpose being to eradicate sources of product failure that can generate negative customer reactions of very high impact (such as recall of car models from the market). The budget devoted to packaging makes up 5 to 10% of the overall budget of the development of a new car model at this engineering centre, making it one of the most costly activities in the development of the car model. Twenty employees are dedicated to the packaging activity on a permanent and full-time basis, constituting a team of an important dimension. Furthermore, packaging is an on-going activity during almost the whole development project, and most importantly, the packaging task is almost the paradigmatic example of a reciprocally interdependent task (our main criterion above). Figure 1 provides a graphical illustration of the highly reciprocal, technical interdependence: the objective of the packaging task is to optimize the technical performance of the overall system by ensuring that interactions between different

combinations of components conform to technical specifications. As explained in more detail below (see figure 2), such reciprocal interdependence on the technical level is also reflected on the organizational level: carrying out the full packaging check requires coordinating 26 engineering teams who develop the components and sub-systems in parallel and whose possibilities and constraints in doing so depend on the decisions taken in the other teams.

‘Virtual packaging checks’ (see below) are performed on a bi-weekly basis during the entire life of the development project, from concept development to the industrialization phase. Thus, the packaging activity recurs at least 48 times during a project (and 20 times during our 10 months observation). When our observations started, the project had run for two years and had been overseen by the packaging team from the start of the project. Since the people staffed to the team had previous experiences in packaging activities for other projects, their performance is not idiosyncratic to the projects under observation. In fact, the packaging tasks are governed by a set of company norms and procedures developed by the OEM that apply to all packaging activities performed in the firm. The key norm was established in 2003 by a dedicated team that included company experts for virtual tools. The team members we interviewed held the same positions in the company both during the period of observation and before, as regards their involvement in the “packaging” process. This stability was one of the keys to being able to observe the formation and role of organizational routines.

The firm’s management confirmed that this task was an adequate object of observation since packaging is a typical activity for the engineering centre and tightly linked to its objectives. Packaging check results are important inputs for several stages in the new product development process. In many instances, a green light from packaging is a necessary condition before moving into the next phase of development. This fact illustrates the high reciprocal interdependence of the packaging task, and indicates how essential a part of the

development process it is. Most importantly, moreover, the objective of the packaging task is to coordinate the reciprocal interdependencies in the development of complex products. To benefit from the lever of concurrent engineering in component design, individual component development teams need to align and fine-tune their development work through many gates (checks). The kinds of checks run in packaging can include the following:

- Checks of project objectives
- Checks of absence of static and cinematic interferences, compliance with minimum distances
- Functional and esthetical checks
- Checks on safety and ease of maintenance
- Ergonomic checks and checks of manufacturing ergonomics
- Checks of manufacturability

Figure 2 shows how development teams are divided into packaging for mechanics (engine, suspensions, etc.) and packaging for the chassis. For both mechanics and chassis packaging activities, two kinds of checks have to be performed: virtual checks and physical checks. In virtual checks, the individual components are described by CAD/CAM files stored in the Product Data Management system (PDM). These files then interact virtually by software that simulates their interaction (Digital Mock-Up, DMU). Virtual checks serve to check the feasibility and coherence of the designs early in the design process, either substituting for physical checks or complementing them. Note that the virtual prototype (Digital Mock-Up, DMU) constitutes, over most of the development project, the reference point and the object on which development efforts focus. Physical checks are usually carried out late in the process and involve physical prototypes. In what follows we refer to activities carried out via DMU for the packaging of mechanics. As this section should have made clear, packaging is a paradigm example for a highly reciprocally interdependent task.

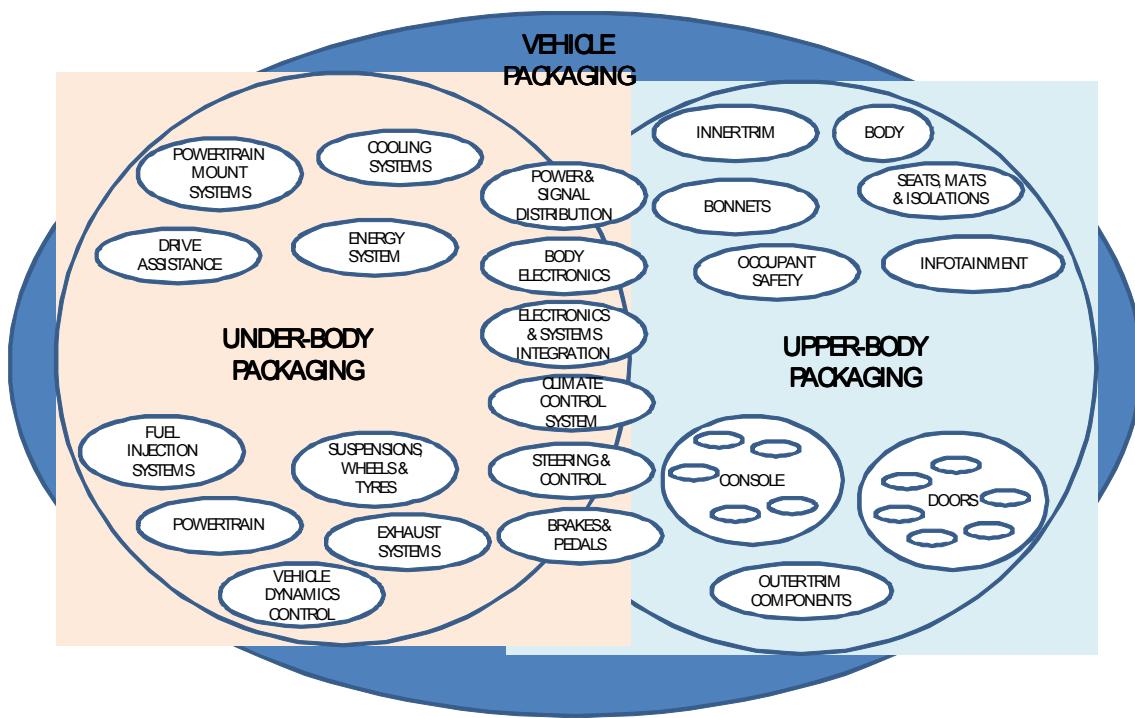


Figure 2: The structure of packaging

3.3 Procedures regarding the packaging task

Packaging activities are described comprehensively and thoroughly by a whole set of ‘procedures’ (which in this firm refer to how to organize tasks) and ‘norms’ (setting technological standards, such as threshold values). Amongst others, the following selection of ‘procedures’ and ‘norms’ are pertinent to packaging:

- ‘Norm on the product development process’
- ‘Procedure of vehicle packaging’
- ‘Procedure on virtual checks via DMU’
- ‘Check-lists of virtual check via DMU: chassis, mechanics, manufacturability, ergonomics’

- ‘Procedure on use of knowledge management system and use of CAD in the product development process’
- ‘Norm on the lay-out of drive-train and engine’

The list could be continued but illustrates that packaging is a highly ‘regulated’ activity in the sense that some very comprehensive and very precise instructions, some having close to 100 pages, exist for how to carry out packaging.

Briefly, to start the packaging process, the ‘raw material’ for the packaging calculations (we focus on virtual packaging checks), the CAD files of all the car’s components, are developed by development engineers grouped into 26 teams: chassis, drive-train, steering, and so on (see figure 2). The CAD files they produce are kept in a database that is called product data management system (PDM). Designing components often takes many months, considering adaptations required to accommodate changes in other components, so it is an on-going activity. Packaging is carried out by the packaging team, which consists of 20 employees who do nothing other than packaging. Once a packaging activity is triggered, they transfer the files that correspond to the components they are checking to another computer system (and file format) and use a DMU (Digital Mock-Up) virtual simulation software to run calculations that simulate interaction between the components to evaluate the effects in the various dimensions (ergonomics, etc.). All the performances listed above (manufacturability, space limits, etc.) produce a list of anomalies that specifies all instances in which the requirements were not met.⁴ One member of the packaging team then filters this list for anomalies that do not create practical problems or whose cause has already been eliminated, and all the heads of the different engineering units (engine, interiors, etc.) then meet to discuss the remaining anomalies. The anomalies list allows the packaging team and the engineers who are the

⁴ For every check to be carried out, there are lists of technical norms (threshold values such as minimum distances) as well as technical procedures for how to run the checks.

'owners' of the problem (i.e., the component that exhibits anomalies) to coordinate their efforts. The anomalies are also registered in the log-book, the official document used by the development platform to keep track of the project's evolution, record and monitor the resolution of the anomalies as well as the technical solutions adopted, and enable team members to discuss potential modifications of the project's objectives with the client. The packaging activity follows a cycle with a regular rhythm, described in the written procedures that prescribe fortnightly cycles of checks for most components (some follow weekly cycles, but the object we follow in this paper was subjected to bi-weekly cycles) (Figure 3).

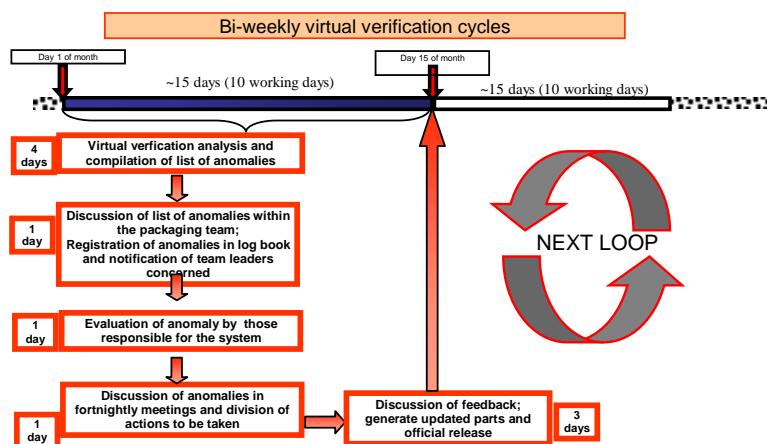


Figure 3: Flow chart of virtual check cycles

3.4 Recurrent interaction patterns in packaging

In this section, we first describe how the task is supposed to be carried out according to the procedures and norms and, subsequently, how it is carried out in practice. The description below follows the nine-step structure of packaging as described in the procedures.

Step #1: Releasing the CAD files for use by the packaging team

The procedures and norms prescribe that each engineer responsible for a component should release his or her CAD files periodically (for most components the rhythm is a fortnight).

There are 26 teams of engineers, whose team leaders are responsible for making sure their team-members release the CAD files on time. Typically, the delivery deadline is set a day or two before the virtual check is run (some transformation of the data has to take place, as the CAD system and the system used for the virtual checks are different and files have to be transferred into another format).

In reality, updated CAD files are often not delivered on time. This triggers quite an intense activity on part of the packaging team, who remind engineers to release their files. Two formal measures can be taken in case updated CAD files are not released to the packaging team on time, depending on the cause of the delay. In case of hardware or software problems, often the cause of delays, the problem is addressed by IT people (neither the engineers nor the packaging people are much involved) but if a delayed release is related to delays in the engineers' activities, no particular action is taken. No authorization for not releasing the updated files is required; not even a notice of the late release or lack of release needs to be made. Since there are no such requirements, the updated CAD files are often released late or not at all without any notice, preceded by an intense reminder activity on part of the packaging team.

Step #2: Creating a reference base of files for the packaging software

The second step consists of creating a homogenous reference (in terms of the state of the design) for purposes of the packaging check. To do so, (i) a point of time of reference has to be defined to which the designs of all components will be compared. The CAD files of all components are then (ii) transformed into files in the format of the packaging software, which results, in simple terms, in a 'snap shot' of the whole vehicle at that moment. Packaging check

calculations will be run on this ‘object’. According to the procedures and norms, the person in charge of packaging should, after the point fixed as reference, download the CAD files from the product data management (PDM) system (where all CAD files are kept centrally), identify the CAD files that correspond to the components of the vehicle (there might be more than one file per component, and several components to realize an assembly), and then upload them into the DMU software. The procedures also contain a back-up procedure so that in case of late release, an override folder is created. By default, the system then automatically downloads all CAD files, picking the last available version previous to the point of reference. In reality, due to the late delivery of updated CAD files (i.e., so that only ‘old’ files, already used for the last cycle of the packaging check, are available in the product data management system), the process in step two already diverges from that specified for the procedure. Another divergence also arises in step two: ignoring the procedure on the override folder, no one on the team creates the override folder. In fact, in our interviews, all members of the packaging team told us they had never used it, some even saying they had not even heard of such a thing. Rather, the packaging check is simply run with the last available files (as identified automatically by the system). At the same time, the packaging team attempts, by way of informal communication with the engineering teams, to assess how serious the omissions and their consequences are (and whether the overall packaging check might be feasible without the new files).

This step of the packaging process has a second aspect to it as well. There are two ways in which members of the packaging team can go about starting the packaging calculus. The direct way is by transforming the CAD files into packaging software format and saving them in the product data management system (PDM). The result is the same data in two formats in the PDM system. The packaging software then calls up the files in the PDM system. The indirect way is by transforming the CAD files into packaging software format, then saving them in the packaging system (VisMockUp), not the PDM system. Both ways are described in

the procedures. The direct way, however, provides a higher degree of control of the process because the packaging calculus, run on the central server, leaves a record on the system that can be useful. In practice, however, due to unstable IT performance, the packaging team prefers using the indirect way.

Step #3: Release of override folder

The procedures prescribe the release of the override folder and notification of its creation to the head of packaging. In practice, as the override folder is not created in the preceding step, it can obviously neither be released nor announced.

Step #4: Selection of the individual files to be used in the packaging calculus

In the fourth step the type of version of the product to be checked is selected (for example, the type of engine). Its correct execution is closely linked to the execution of the second step.

Step #5: Uploading the files into the packaging software

At this point, there are two possibilities. Either only the files required for a particular packaging calculus can be uploaded in the packaging system, or all files for the whole vehicle can be uploaded. In the procedures, the latter option is indicated as best practice for virtual packaging checks, and it is also what happens in practice. Obviously, as opposed to the partial upload, the demands on IT resources are heavier.

Step #6: Running the packaging calculus and generation of its output

The calculus itself is run by the software. Having run through the preparation described in the previous steps, the action demanded from members of the packaging team at this point is just to start the calculation. Packaging team members can choose between three modalities of running this session: interferences and anomalies of one (group of) component(s) against one, one against all, all against all. The norm on virtual packaging suggests utilizing almost

exclusively the modality ‘all against all’. This is precisely what we observed in practice, too: no gap was observed.

Step #7: Analysis of anomalies identified

The output of the analysis is a list of anomalies which, as described above, can be very long. In analyzing it, packaging team members therefore use filters. The norm on virtual packaging suggests filtering the results according to type, status and result. In practice, we observed that packaging team members filtered by sorting the column ‘result’, thus sorting by decreasing criticality. Going through this list from the top, they then compare each potential anomaly (the list identifies ‘theoretical’ anomalies, indicating where limit values have been violated) with a checklist to determine whether the anomaly is likely to be only of minor impact (for instance, a minimum distance could not be kept, but the two parts are supposed to touch each other).

For each line of the table, the result of this check is recorded. It is here that a very small difference between the procedures and practice can be noted. Employees do not usually note ‘p’ (positive) or ‘n’ (negative) outcomes of this confrontation with the checklist, but only mark the cases that are problematic.

Step #8: Following up on the relevant anomalies

For those anomalies that are considered relevant according to the check in the previous step, the packaging team staff needs to individuate the components responsible for the anomaly (remember there are always several components involved in producing an anomaly, but the cause could be one of them, such as a part that has an uneven surface and rubs against another one, causing damage). Once the component that caused the anomaly has been identified, the responsible engineer is also identified. Those anomalies that cannot be attributed to one component or team member are assigned to the team leader of packaging who then determines who is responsible for the anomaly. Furthermore, the packaging team member

compiles an anomaly record, in which the anomaly is described. A record of the anomaly is also put in the project log-book, and a link to the anomaly record is created in the log-book. The responsible team member is then notified by an email with the complete log-book attached, which he or she can filter to see the lines that concern him or her. At this point, a slight divergence from the procedures can be noted. The procedures prescribe a notification of the responsible party in the official record of the vehicle platform. Rather, only the email is sent, leaving less on record. The effect is that organizational memory is weakened.

Step #9: Reporting on packaging

This step consists in compiling indicators on the packaging activity, anomalies found, and so on. This phase is usually not carried out completely.

4. Analysis

The research question tackled in this paper is ‘What does employing routines as analytical perspective contribute to understanding the behavior of organizations?’ In the analysis presented in the previous section, employing organizational routines as an analytical perspective has allowed us to do a number of things.

Capture how tasks are actually accomplished in practice (performative level)

Describing recurrent interaction patterns has turned out to be useful for capturing and describing the behavior of organizations as *actually* carried out. Moreover, two further insights relating to this point have emerged from the case study. The *recurrent* interaction patterns not only describe how tasks are actually carried out, but they also capture how tasks are *usually* accomplished. They capture what characterizes the firm (for instance, how assembly on a production line takes place at Ford and at Toyota characterizes the ‘Ford way’

or ‘Toyota way’ of producing cars). Recurrent interaction patterns thus contribute a sharp and focused descriptive device. In step #1, for instance, describing the recurrent interaction patterns to be observed, one realizes that the average on-time delivery rate is almost always below 100%. It is typical for the organization we analysed to deliver late, and a whole spate of other tasks and routines were triggered by that (one person even had the official task to remind the others).

Second, this device can make visible what before remained invisible to managers. In the packaging process, for instance, management was largely unaware that no systematic record was available of which tests that should have been triggered by the list of anomalies had actually been carried out (step #7). This is an example of the knowledge-related issues that, as we discussed in the beginning, organizational routines can capture. In our example, analysing the packaging process in terms of organizational routines reveals that no systematic organizational memory is provided for regarding which of the tests that should have been triggered by the list of anomalies has actually been carried out (and in particular, in case the result was negative, i.e., where there was no problem). In consequence, management only perceives the effects (such as interferences), but cannot identify their causes. For instance, in our case, engineers ran certain tests but did not leave any trace of a negative test result (the problem was not linked to this test). Leaving no trace, however, made the diagnosis later much more difficult. Identifying how tasks are accomplished in practice (i.e., by which recurrent interaction patterns) therefore contributes at least two important things for managers: the description provides a record, an organizational memory and such a description, by uncovering aspects that before were hidden from managers’ view, can then be the basis of management efforts. There are two ways in which this is important. Knowing what *usually* is the case allows managers to focus their efforts on systematic rather than one-off events, errors, problems, biases, and performance drivers. Recall that much research supports the idea that many processes in organizations are routinized (Cyert and March) so

'acting on' and influencing those processes, rather than one-off events, should be an important focus for managers. (One could even argue that although sometimes strategic decisions and adaptations to external shocks need to be taken, such adaptations subsequently will involve precisely the adaptation of recurrent interaction patterns, specifying how tasks will be accomplished under the new circumstances.) Moreover, if managers are not aware of how tasks are usually accomplished in practice, it will be difficult to design and implement systematic interventions that will be effective. Instead, their acts will be based on the description contained in the rules (i.e., as the tasks should be accomplished), perhaps with a huge gap to the actual recurrent interaction patterns that could render managerial action incongruent and ineffective.

Identify the 'governance gap' by contrasting how tasks are actually carried out in practice with how they should be carried out according to the standard operating procedures

Once a description of how tasks are actually accomplished in practice is available, it can be compared with how these tasks should be carried out according to the standard operating procedures. The gap that often appears between these two levels can thus be described (see figure 4). (Note that if managers do not have a description of how tasks are actually accomplished, the gap by definition, but most likely not in practice, will be zero.)

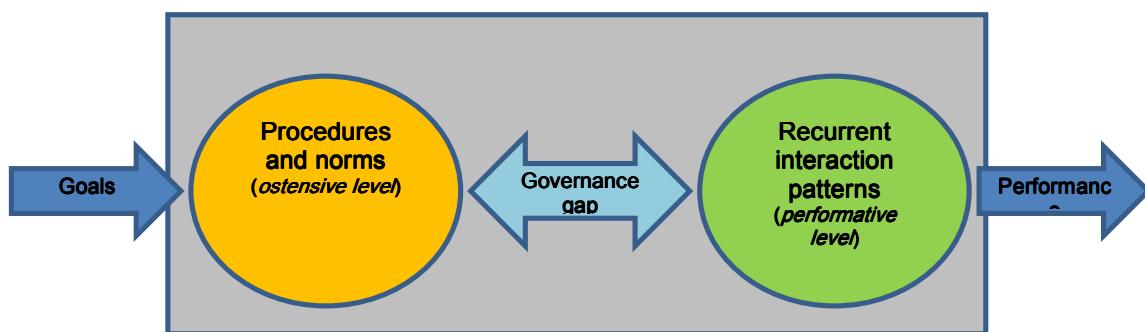


Figure 4: Gap between standard operating procedures and recurrent interaction patterns

Being able to describe the gap between the standard operating procedures and the recurrent interaction patterns prepares the ground for designing governance mechanisms that can address it. A description of organizational behaviour clearly boosts the possibilities for designing interventions of organizational change and organizational development that improve the structures and mechanisms that govern organizational behaviour. In the packaging task, we identified significant gaps in case of the release of CAD files late (step #1), how the packaging team used its time (increasingly, running to convince engineers to release their files) (step #1), and the (lacking) use of the override folder (step #2).

At this point, one could argue that *any* description of the actual behaviour allows a comparison between it and the rules and procedures to determine whether they were implemented faithfully. Note, however, that a description that uses the organizational routines framework enables to examine much more than just whether rules were followed or enforced; other frameworks (such as business process reengineering) could do this, too. Applying the concept of organizational routine in the analysis provides much more than a ‘rule-enforcing’ lens. It provides a vantage point that allows the analysis of the behavior of organizations more profoundly because the framework systematically takes into account a range of causes of behaviour (and performance drivers) that is wider than just the design and implementation of rules. (For instance, dispositions for helping colleagues catch up delays in the delivery of their files to the packaging team might explain why such delays are continuously occurring.) We will develop this aspect in more depth in section six.

Identify the performance effects of the recurrent interaction patterns, and of the ‘governance gap’

Providing a description of recurrent interaction patterns and identifying the governance gap provides the possibility of understanding the drivers of particular positive or negative

performance effects and linking these effects back to their causes. In packaging, for instance, the benefits realized from concurrent engineering are to a considerable extent determined by the release of the CAD files and whether they happen on time for all 26 engineering teams.

The gap in step #1 has the most significant consequences for performance of the packaging task: due to a lack of monitoring instruments and of the requirement to communicate lacking or delayed release, behaviours result that remain hidden or ‘invisible’ to management (which relies on summary indicators or results to follow the process). Moreover, the process is structured in such a way that it becomes the responsibility of the packaging team leader not only to chase the CAD files, but also to identify the cause of the delayed release. This structure confers a hidden *de facto* power onto the engineers who design components, as their (non-)compliance with the release deadline has immense organizational consequences. If one or more of the 26 engineering teams do not release their CAD files on time (maybe even after postponing the deadline of the virtual packaging by some hours or days), the effects are the following:

- The packaging check as a whole has a much lower significance, as it is based only in part on the latest version of all the components. In the extreme, it is close to useless or already outdated, as the design teams that have not released their latest set of files will most likely have changed the designs over the past two weeks.
- The teams who did not release their files (on time) will work for another two weeks without feedback on whether their designs interfere with any of the other components or show any other anomalies. If that should be the result of a later packaging check, time will be lost, and it will be much more costly to reverse the design changes made in the meantime. Furthermore, there is a risk that anomalies have ‘increased’ in the meantime and might now require changes on part of other design teams also.
- There are important effects on employees as well: tensions between the engineering teams and the packaging team, distraction of resources and time of the packaging

team, danger of de-legitimation of the packaging check's results, and the risk of demotivation of the packaging team.

The gap between the procedures and the recurrent interaction patterns in this step has serious consequences because of the importance of running packaging checks (their lack might cause serious quality problems, for instance) and because of the multiplicator effect of errors and deficiencies over the following iterations of the packaging check. In our case, there is not even a mechanism to know how many CAD files were not released on time, which aggravates the problem. This factor alone could seriously dampen the benefits of concurrent engineering.

Step #2 also creates significant performance effects. Not creating an override folder substantially reduces the capability to control the process in real time, of monitoring it over the whole life of the project, and to reconstruct it later from organizational memory to learn from earlier experience. Without the override folder, it is difficult to know on what basis the packaging was carried out (which files were updated and which ones were not). All of these details remain invisible to management because they lack profound and detailed knowledge of how tasks are carried out in practice (including informal variations on the official procedures, which are tolerated and have stabilized). Because packaging team members spend time communicating informally with engineering team members to convince them to release the latest version rather than use the override folder, the workload shifts to informal communication, which is less powerful in terms of organizational memory, remedying problems, and lead time. It also involves higher resource requirements, an increased exposure to potential sources of error, and possibly an increase in the number of anomalies that are identified by the packaging check (some might already have been taken care of in the meantime).

In step #6, as a consequence of checking interferences in ‘all-against-all’ mode, in combination with the choice of uploading the complete files describing the vehicle (step 5), the calculus becomes very time- and resource-consuming. It also produces a very long list of interferences and anomalies. The effect on the analysis of the anomalies is negative, since the packaging team must spend more time going through the list, increasing the rate of error in assessing it.

The description of the recurrent interaction patterns and their gap to standard operating procedures thus helps to identify performance effects. Our analysis of step #2 identifies some causes for employees choosing to create direct rather than indirect VisMockUp sessions for running the packaging calculations. The transfer of all CAD files into the packaging software takes a long time because the network is somewhat slow for such purposes. If there are problems, the workstations have to be rebooted, taking even more time. Furthermore, a software bug on the server does not allow recalling checks that were run in the direct mode. Similarly, in step #5, uploading all files into the packaging system, rather than only the files needed for the check to be run, enables running checks for things were not planned at the outset without leaving the active session and starting a new one (which requires starting again from step one).

exhibited and therefore cannot also describe its *causes* at the same time.⁹ Therefore, we find that in future research on organizational routines and behavior, we need to attempt to create a close match between empirical constructs on the one hand and a theoretical framework on the other hand that has something to say on the causes of espoused behavior. While that is not at all surprising, that is not always what one can find in the literature so far.

The empirical analysis has shown that organizational routines focus on phenomena of crucial interest to students of organizations since focusing on reciprocally interdependent tasks examines tasks that pose crucial organizational challenges. As remarked recently in organization literature, it is not always easy to pick units of analysis that capture the phenomenon under inquiry in a way that it is fruitful for an analysis by organization scholars (Heath and Sitkin 2001, Hackman 2003). The indications from our empirical analysis are therefore promising.

Finally, our empirical analysis makes an important contribution to the conceptual discussion of how to define an organizational routine: precisely the multi-level perspective that the organizational routines concept points to has turned out to be of great help for the empirical analysis and drawing conclusions from it. The most recent addition, the level of dispositions, points to a fruitful avenue to explore in the future. Because dispositions clearly refer to the causes of espoused behavior (Hodgson 2008, Knudsen 2008) and do so more immediately than rules or standard operating procedures, they seem to hold yet more potential for the concept of organizational routines as perspective in analyzing organizations.

⁹ We refer to the same point of time here. We do not want to imply that behavior in a previous time step can not influence, or cause or trigger, behavior in a subsequent time step.

Bibliography

- Adler, PS, Clark, KB, 1991. Behind the learning curve: A sketch of the learning process. *Management Science* 37, 267-281.
- Adler, PS, Goldoftas, B, Levine, DI, 1999. Flexibility versus efficiency? A case study of model changovers in the Toyota production system. *Organization Science* 10, 43-68.
- Argote, L, Epple, D, 1990. Learning curves in manufacturing. *Science* 247, 920-924.
- Barley, SR, Kunda, G, 2001. Bringing work back in. *Organization Science* 12, 76-95.
- Becker, MC, 2004. Organizational routines: A review of the literature. *Industrial and Corporate Change* 13, 643-678.
- Becker, MC, Lazaric, N, Nelson, RR, Winter, SG, 2005. Applying organizational routines in understanding organizational change. *Industrial and Corporate Change*, 14, 775-791.
- Cohen, M, Bacdayan, P, 1994. Organizational routines are stored as procedural memory: Evidence from a laboratory study. *Organization Science* 5, 554-568.
- Cyert, RM, March, JG, 1963 [1992]. *A Behavioral Theory of the Firm*. Oxford: Blackwell.
- Dosi, G, Nelson, RR, Winter, SG, 2000. Introduction: The nature and dynamics of organisational capabilities. In: Dosi, G, Nelson, RR, Winter, SG (Eds.). *The Nature and Dynamics of Organisational Capabilities*. Oxford: Oxford University Press, 1-22.
- Edmondson, AC, Bohmer, RM, Pisano, GP, 2001. Disrupted routines: Team learning and new technology implementation in hospitals. *Administrative Science Quarterly* 46, 685-716.
- Eisenhardt, KM, 1989. Building theories from case study research. *Academy of Management Review* 14, 532-550.
- Eisenhardt, K. M. and Graebner, M. E. (2007). 'Theory building from cases: opportunities and challenges', *Academy of Management Journal*, 50/1: 25-32.
- Feldman, MS, 2000. Organizational routines as a source of continuous change. *Organization Science* 11, 611-629.
- Feldman, MS, 2003. A performative perspective on stability and change in organizational routines. *Industrial and Corporate Change* 12, 727-752.
- Feldman, MS, Pentland, BT, 2003. Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly* 48, 94-118.
- Hackman, JR, 2003. Learning more by crossing levels: evidence from airplanes, hospitals, and orchestras. *Journal of Organizational Behavior* 24, 905–922
- Hammer, M, Champy, J, 1993. *Reengineering the Corporation – A Manifesto for Business Revolution*. New York: Harper-Collins
- Heath, C, Sitkin, SB, 2001. Big-B versus Big-O: What is organizational about organizational behavior? *Journal of Organizational Behavior* 22, 43-58.
- Hodgson, GM, 2008 forthcoming. The Concept of a Routine. Becker, MC, ed, *Handbook of Organizational Routines*. Cheltenham: Edward Elgar

- Hodgson, GM, 2004. Reclaiming habit for institutional economics. *Journal of Economic Psychology* 25, 651–660.
- Hodgson, GM, Knudsen, T, 2004a. The firm as an interactor: Firms as vehicles for habits and routines. *Journal of Evolutionary Economics* 14, 281-307.
- Hodgson, GM, Knudsen, T, 2004b. The complex evolution of a simple traffic convention: The functions and implications of habit. *Journal of Economic Behavior and Organization* 54, 19-47.
- Hutchins, E, 1991. Organizing work by adaptation. *Organization Science* 2, 14-39.
- Knott, AM, McKelvey, W, 1999. Nirvana efficiency: A comparative test of residual claims and routines. *Journal of Economic Behavior and Organization* 38, 365-383.
- Knudsen, T, 2008 forthcoming. *Organizational Routines in Evolutionary Theory*. Becker, MC, ed, *Handbook of Organizational Routines*. Cheltenham: Edward Elgar
- Leidner, R, 1993. *Fast Food, Fast Talk – Service Work and the Routinization of Everyday Life*. Berkeley: University of California Press
- Lomi, A, Larsen, ER, 1999. Evolutionary Models of Local Interaction – A Computational Perspective. McKelvey, W, Baum, JAC (Eds.). *Variations in Organization Science*. Thousand Oaks, CA: Sage, 255-278
- March, JG, Simon, HA, 1958 [1993]. *Organizations*. Oxford: Blackwell.
- Miles, MB, Huberman, AM, 1994. *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd edition. Thousand Oaks, CA: Sage.
- Narduzzo, A, Rocco, E, Warglien, M. 2000. Talking about routines in the field. In: Dosi , G, Nelson, RR, Winter, SG (Eds.). *The Nature and Dynamics of Organizational Capabilities*. Oxford: Oxford University Press, 27-50.
- Nelson, RR, Winter, SG, 1982. *An Evolutionary Theory of Economic Change*. Cambridge: Belknap Press/Harvard University Press.
- Orlikowski, WJ, 2000. Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science* 11, 404-428.
- Penrose, E, 1959 [1995]. *The Theory of the Growth of the Firm*. Oxford: Oxford University Press.
- Pentland, BT, 1995. Grammatical models of organizational processes. *Organization Science* 6, 541-556.
- Pentland, BT, 2003a. Conceptualizing and measuring variety in the execution of organizational work processes. *Management Science* 49, 857-870.

- Pentland, BT, 2003b. Sequential variety in work processes. *Organization Science* 14, 528-540.
- Pentland, BT, Rueter, H, 1994. Organisational routines as grammars of action, *Administrative Sciences Quarterly* 39, 484-510.
- Pettigrew, AM, 1990. Longitudinal field research on change: Theory and practice. *Organization Science* 1, 267-292.
- Reynaud, B, 1996. Types of rules, interpretation and collective dynamics: Reflections on the introduction of a salary rule in a maintenance workshop. *Industrial and Corporate Change* 5, 699-721.
- Siggelkow, N. (2007). 'Persuasion with case studies', *Academy of Management Journal*, 50/1: 20-24.
- Simon, HA, 1947 [1997]. *Administrative Behaviour*. New York: Free Press.
- Szulanski, G, Winter, SG, 2002. Getting it right the second time. *Harvard Business Review* 80/1: 62-71.
- Thompson, JD, 1967. *Organizations in Action – Social Science Bases of Administrative Theory*. New York: McGraw-Hill.
- Winter, SG, Szulanski, G, 2001. Replication as strategy. *Organization Science* 12, 730-743.
- Yin, RK, 1994. *Case Study Research*, 2nd edition. London: Sage.
- Zollo, M. and Winter, S. (2002). 'Deliberate Learning and the Evolution of Dynamic Capabilities', *Organization Science*, 13/3: 339-35.