Provisional Directionality: Unpacking the prospective dimension of organizational routines

Christian A. Mahringer[1], Katharina Dittrich[2], and Birgit Renzl[1]

[1]University of Stuttgart, Germany
[2]University of Warwick, UK

Paper accepted for publication in 'Organization Studies'.
Author Accepted Version

Provisional Directionality: Unpacking the prospective dimension of organizational routines

Abstract

While it is well known that organizational routines guide actions in practice, the prospective dimension of this guiding function remains underexplored. We examine the prospective dimension of routines through an ethnographic study of Scrum software development teams. Our findings show how actors continuously (re)create a tentative realm of possible paths in what we refer to as provisional directionality. The findings describe how provisional directionality is continuously (re)created through three patterning mechanisms: initiating possible paths, including or excluding possible paths, and reorienting possible paths. Provisional directionality contributes to routine dynamics research by, first, unpacking the prospective dimension of patterning in routines, second, emphasizing temporality in routine performances as flow, and, third, showing how heavily scripted routines enable action when it is unclear how to proceed. We also discuss how provisional directionality may inspire future research on routines in contexts such as product and service innovation, agile software development, and emergent strategizing.

Keywords

Agile, innovation, path, patterning, practice theory, routine dynamics, Scrum, software development, strategizing, strategy, temporality, wayfinding.

Introduction

Routine dynamics research highlights how organizational routines guide actions in practice (Feldman, Pentland, D'Adderio, Dittrich, Rerup, & Seidl, 2021; Feldman, Pentland, D'Adderio, & Lazaric, 2016; Mahringer, Pentland, Renzl, Sele, & Spee, 2024). Organizational routines entail task-oriented sequences of actions that actors collectively consider characteristic, that is, typical action patterns (LeBaron, Christianson, Garrett, & Ilan, 2016). These typical action patterns reflect what routine participants come to recognize as characteristic ways of getting things done. A crucial role of typical action patterns is *guiding*, which means that they provide orientation for what to do next in everyday situations. As Feldman and Pentland (2003, p. 105) emphasize, typical action patterns influence routine performances '*prospectively* [emphasis added], as a guide to what actions ought to be taken.' Existing research has shown that there are two ways of guiding. One way of guiding is when typical action patterns prescribe sequences of actions that participants are expected to follow (Danner-Schröder & Geiger, 2016; LeBaron et al., 2016). The second way of guiding occurs when they foster flexibility by enabling actors to select from a familiar repertoire of actions and by encouraging flexible adjustments to the situation at hand (Danner-Schröder & Geiger, 2016; Dönmez, Grote, & Brusoni, 2016; Pentland & Rueter, 1994; Sailer, Loscher, & Kaiser, 2023; Turner & Rindova, 2012).

However, even though the prospective dimension of guiding has been emphasized conceptually (Chia & Holt, 2009; Shotter, 2008; Wegener & Lorino, 2020), it has rarely been examined empirically. Conceptually, the notion of *prospective* refers to actors' experience of living forward from within specific situations. As Tsoukas (2017, p. 134) notes, '[p]ractitioners living their lives forward is less orderly, more fluid and uncertain, and of a different kind than what appears to scholars who study it backwards.' Yet how exactly prospective guidance works in routines remains underexplored.

Studying the prospective dimension of guiding is important for at least two reasons. First, it allows us to understand how actors orient to typical action patterns when it is not clear how to proceed. This applies especially in generative contexts—such as product and service innovation, agile software development, and emergent strategizing—where neither the outcomes nor how to achieve them are known in advance. In such situations, routine participants must align on a joint course of action, which comes into view prospectively as they move forward. Second, while routine dynamics research has emphasized that performing routines entails noticing and enacting possibilities (Feldman & Sengupta, 2020; Feldman, Worline, Baker, & Lowerson Bredow, 2022; Pentland, Mahringer, Dittrich, Feldman, & Ryan Wolf, 2020; Pentland & Rueter, 1994), research on guiding has tended to treat possibilities as generic features of typical action patterns. However, these accounts provide little insight into how typical action patterns enable actors to see possibilities for action. Considering the prospective dimension of guiding enables us to address this gap by showing how living forward entails seeing possibilities from within specific situations—where people currently are—and enacting possibilities to move forward. Thus, we ask the following research question: How do routine participants use typical action patterns to prospectively guide their actions?

We draw on an ethnographic study of a Scrum software development team, which is particularly well-suited to study the prospective dimension of guiding. Scrum is a widely used agile framework that splits software development into Sprints. Scrum routines are heavily scripted (Jarzabkowski, Lê, & Spee, 2017), providing actors with well-defined typical action patterns they can orient to. At the same time, however, it is often unclear how to proceed because, neither the outcomes nor the best way to accomplish them are clear a priori in the development of innovative software products. Thus, actors orient to typical action patterns prospectively, which guides what to do next. Therefore, Scrum offers a revelatory context to elaborate our research question.

To conceptualize our findings, we employ the notion of *paths* (Danner-Schröder & Ostermann, 2022; Feuls, Hernes, & Schultz, 2025; Goh & Pentland, 2019; Hansson, Hærem, & Pentland, 2023; Karnøe & Garud, 2012). A path refers to a time-ordered sequence of actions, which may represent either a possible path or an actual (i.e., enacted) path (Pentland et al., 2020). We observed that team members often experienced many, none, or divergent possible paths, making it unclear how to proceed, and requiring them to align around possible paths to enact. This is closely related to the notion of *wayfinding* (Bouty, Gomez, & Chia, 2019; Chia & Holt, 2009; Ingold, 2011; Sminia & Corvalán, in press), which refers to a mode of existence in which direction emerges in and through the act of going. The concepts of path and wayfinding enable us to see how actors orient to typical action patterns to figure out the paths they could enact.

Our findings show that team members oriented to their Scrum-based, typical action patterns to (re)create a tentative realm of possible paths that the team could collectively use to proceed—what we label *provisional directionality*. Provisional directionality enabled the team to align their actions in a coordinated fashion. We also find three mechanisms through which participants oriented to typical action patterns and continuously (re)created provisional directionality: initiating, including or excluding, and reorienting possible paths.

Provisional directionality contributes to routine dynamics research in multiple ways. First, it enhances our understanding of *patterning* in routines (Danner-Schröder & Geiger, 2016; Feldman & Pentland, 2022; LeBaron et al., 2016), by unpacking how it prospectively guides actions. Routine-based patterning creates a realm of possible paths that actors can enact to proceed. Guidance, in this sense, is not about what ought to be done but about what could be done. Second, provisional directionality advances our understanding of *temporality* in routines (Howard-Grenville, 2005; Ritter, Danner-Schröder, & Müller-Seitz, 2024; Turner & Rindova, 2021) by conceptualizing temporality as a continuous flow rather than a set of temporal

orientations. Third, provisional directionality sheds light on how heavily scripted routines help in situations where it is unclear how to proceed. Rather than merely prescribing sequences of actions (Danner-Schröder & Geiger, 2016; LeBaron et al., 2016) or allowing for flexible recombination and adjustment (Danner-Schröder & Geiger, 2016; Dönmez et al., 2016; Pentland & Rueter, 1994; Sailer et al., 2023; Turner & Rindova, 2012), typical action patterns can also guide actions by (re)creating provisional directionality, which provides a coherent yet open-ended way forward. Moreover, we discuss how provisional directionality could inspire future research on routines in contexts such as product and service innovation, agile software development, and emergent strategizing.

Theoretical background

The guiding function of organizational routines

Organizational routines are defined as 'repetitive, recognizable patterns of interdependent actions, carried out by multiple actors' (Feldman & Pentland, 2003, p. 95). Routine dynamics research conceptualizes routines as processes of performing and patterning (Feldman, 2016). While performing refers to actions carried out in specific times and places, patterning denotes 'the impact of patterns on actions and the ongoing creation of patterns through actions' (Feldman & Pentland, 2022, p. 850). Although patterning is essential to routines, it can also be accomplished in other coordination mechanisms, such as objects or plans. In this paper, we focus specifically on routine-based patterning.

The concept of patterning entails two analytical emphases. First, *patterning as re-creation* highlights how actors produce recognizable patterns in practice. This view emphasizes the situated actions through which recognizability is accomplished (Feldman, 2016), particularly when situations shift (Sele, Danner-Schröder, & Mahringer, 2025) or when actors hold divergent understandings of the routine (Turner & Rindova, 2012). Recreating patterns is thus

an effortful accomplishment (Pentland & Rueter, 1994), requiring actors to figure out how to sustain recognizability in specific situations.

Second, *patterning as impact* (the focus of our study) refers to how actors orient to typical action patterns in situ. Many routines are heavily scripted, meaning they 'are designed for a particular purpose and are characterized by sequences with a clear beginning, several milestones or specified steps, and a clearly defined end' (Jarzabkowski et al., 2017, p. 241). We use the term 'typical action pattern' to describe task-oriented sequences of actions that actors collectively consider characteristic of routines. These patterns may integrate multiple elements, such as timing (Geiger, Danner-Schröder, & Kremser, 2021), roles (Kremser & Blagoev, 2021), and sequence (LeBaron et al., 2016). Actors can draw on these typical action patterns to prospectively guide actions, retrospectively account for actions, or refer to actions (Feldman & Pentland, 2003). We focus on how the sequential aspect of typical action patterns guides actions prospectively.

While patterning is less problematic in familiar situations, where actors can follow well-established sequences of action, it becomes far more difficult when it is unclear how to proceed. Such situations are typical for generative settings—such as product and service innovation (Deken, Carlile, Berends, & Lauche, 2016; Deken & Sele, 2021; Mengis, Nicolini, & Swan, 2018; Salvato & Rerup, 2018), software development (Baham & Hirschheim, 2022; Benlian, Pinski, & Adam, in press; Mahringer & Danner-Schröder, in press; Ritter et al., 2024), and emergent strategizing (Bouty et al., 2019; Jarzabkowski, Lê, Seidl, & Vaara, 2025; Sminia & Corvalán, in press)—where neither the desired outcomes nor the paths to achieve them are known in advance. In such contexts, it is often unclear how to proceed, making patterning challenging. This is where the paradoxical nature of routines becomes evident (Birnholtz, Cohen, & Hoch, 2007): from a bird's-eye view, the routine (e.g., product development) appears

stable, yet at the level of enactment, each situation is unique, requiring considerable effort to figure out how to proceed.

Routine dynamics research has identified two ways in which typical action patterns can guide action when it is unclear how to proceed. First, actors may adhere to a prescribed, highly detailed sequence. For example, Danner-Schröder and Geiger (2016) describe a case where rescue workers, in the process of setting up a camp, followed their typical action pattern rather than immediately assisting a victim who approached for help. Trained to suppress distractions, rescue workers prioritized establishing the camp. When deviations from prescribed sequences become necessary, actors must actively render it intelligible by linking it to situational cues (LeBaron et al., 2016). In these instances, typical action patterns enable actors to retrospectively account for deviations rather than guiding actions prospectively.

Second, typical action patterns can encourage flexible recombination and adjustment. Danner-Schröder and Geiger (2016) show that, in performing the Search and Rescue Routine, actors are trained to flexibly select and recombine actions from a familiar repertoire. Similarly, Dönmez et al. (2016) and Sailer et al. (2023) demonstrate that Scrum-based typical action patterns enable actors to reprioritize tasks and adjust plans in response to changing customer needs. Despite these insights, however, the prospective dimension of guiding has been neglected. In particular, we know rather little about how actors use typical action patterns to move forward from within the situations they find themselves in. In the next section, we introduce ways to conceptualize the prospective dimension of guiding in more depth.

Towards considering the prospective dimension of guiding

Feldman and Pentland (2003, p. 105) already allude to the prospective dimension of guiding, noting that actors use typical action patterns 'prospectively, as a guide to what actions ought to be taken.' Feldman (2016, p. 31), however, pointed out that '[t]he anticipation-guiding (Shotter, 2008) quality of the future' has rarely been considered in routines research. Scholars

need to study how actors 'move forward from within current activity' (Feldman et al., 2022, p. 84). This involves considering the flow of time as it is experienced by actors (Blagoev, Hernes, Kunisch, & Schultz, 2024; Feldman et al., 2022), rather than pre-designed temporal orientations that are embedded in typical action patterns (Ritter et al., 2024). In a similar vein, Turner and Rindova (2021, p. 271) argue that 'focusing on temporal orientations in terms of past, present or future alone may be too limiting, as it does not account for time experienced as flow.' Instead, we must consider how the near future is implicated in the present, and, thus, inseparable. Shotter (2008, p. 508) refers to this as 'action guiding anticipations,' through which we 'gain a shaped and vectored sense of where at any one moment in a situation we are, as well as where next in that situation we might go.' This allows scholars to orient towards 'temporal qualities, conditionalities, and directionalities of flows of action' (Baygi, Introna, & Hultin, 2021, p. 424).

As the future is not yet determined, understanding the prospective dimension of routines requires us to make visible the possibilities that guide actions (Feldman & Sengupta, 2020). Conceptually, the notion of *paths* is useful to grasp possibilities (Feuls et al., 2025; Hansson et al., 2023; Karnøe & Garud, 2012). Pentland et al. (2020) use the metaphor of a grassy meadow to illustrate the notion of paths. When people cross the meadow (i.e., perform a routine), they beat down the grass. As more people cross the meadow, they create different paths depending on where they walk. From a bird's-eye perspective, we can observe the emergence of a space of paths on the meadow, some of which have been enacted and others that are possible. Paths have often been considered from an outside perspective (e.g., Goh & Pentland, 2019; Karnøe & Garud, 2012), yet Danner-Schröder and Ostermann (2022) have demonstrated how healthcare professionals in a hospital context form, keep open, enact in parallel, and dissolve actual paths based on situational cues. However, how possible paths—as opposed to actual paths—are orchestrated in situ remains unclear.

Hence, to understand the prospective dimension of guiding, we consider possible paths from within specific situations. This is necessary because we must engage in 'capturing the experience of living forward from within the flow' (Wegener & Lorino, 2020, p. 138) to understand how actors use typical action patterns prospectively. Such an endeavor is consistent with the notion of *wayfinding* (Bouty et al., 2019; Chia & Holt, 2009; Ingold, 2011; Sminia & Corvalán, in press), which refers to a mode of existence in which direction emerges in and through the act of going. Rather than observing situations from the outside, wayfinding stresses that actors are 'reaching out into the unknown and developing an incomplete but practically sufficient comprehension of the situation in order to cope effectively with it' (Chia & Holt, 2009, p. 159). We are interested in the role that typical action patterns play in this regard. Thus, we ask: How do routine participants use typical action patterns to prospectively guide their actions?

Method

Context

We draw on an ethnographic study of Team Alpha, a Scrum software development team. Scrum is widely used in many companies (Schwaber & Sutherland, 2020). It requires a team to define issues (e.g., software bugs, features to be implemented) and allocate a fixed length of time (a Sprint) to addressing them. In Team Alpha, the duration of the Sprint was fixed to two weeks. Additionally, Scrum defines a set of routines (e.g., Planning, Daily Scrum, Review), that are accompanied by a set of roles (Product Owner, Scrum Master, developers) and rules (e.g., the duration of Sprints is fixed), upon which the team can rely as it develops the software.

Team Alpha is part of Technology Innovation Corporation (TIC; a pseudonym), a mechanical engineering company with approximately 1500 employees. TIC develops and builds complex automated machines for other business firms. Team Alpha had six to seven members and was part of the business unit that developed new tools and solutions for all of

TIC's other business units. It developed AlphaSoft, a software tool that was central to the human-machine interface of the machines TIC developed. AlphaSoft enabled operators to carry out tasks as disparate as controlling the doors of the machines that make the cut-out displays used in cell phones, starting and stopping a production process, and changing certain parameters of these procedures. The software, which displayed an interactive interface on computer monitors placed close to the machines, also allowed operators to perform complex functions, such as editing tabulated data or controlling multiple machines simultaneously.

Given the complexity of AlphaSoft, it was often unclear for Team Alpha what the outcomes should look like and what the best way was to develop them. Hence, they often encountered situations where it was unclear how to proceed. At the same time, we observed that Scrum's typical action patterns guided Team Alpha prospectively in moving forward. Hence, Scrum provides a revelatory context to examine the prospective guidance of routines.

Data collection

The first author observed Team Alpha as a non-participant observer over a period of twelve months (two to three days per week). Through this ethnographic approach, we collected four kinds of data: observations, interviews, documents, and digital-trace data. As is typical for ethnographic studies (Dittrich, 2021), observations consisted of approximately 1000 pages of detailed fieldnotes generated by the first author, and 35 interviews. In addition, the first author audiotaped meetings whenever possible, resulting in 122 hours of audio-taped meetings. He collected various documents, such as handbooks, presentations, and emails to complement the observations and interviews, and extracted digital trace data from the Jira database. Jira is a widely used digital tool for organizing and managing work. Jira's functions include creating issues on which actors can comment and which can be supplemented with uploaded documents.

Data analysis

Our analysis was prompted by an empirical puzzle. On one hand, we observed actors' struggles about how to proceed in solving issues—reflected in confusion, emotional reactions, and divergent suggestions. On the other hand, Team Alpha enacted heavily scripted routines (Jarzabkowski et al., 2017) in their daily work. This simultaneous coexistence of coherence and open-endedness drove our inquiry. We followed a discovery-oriented, abductive process (Locke, Feldman, & Golden-Biddle, 2015), allowing insights from the data to guide successive steps.

Step 1: Identifying routines and typical action patterns. While Scrum prescribes routines (Schwaber & Sutherland, 2020), Team Alpha developed more detailed, context-specific typical action patterns in their daily work. Most routines were enacted as regular meetings listed in the team's virtual calendar: Refinement, Planning 1, Planning 2, Sprint, Daily Scrum, and Review. We also added Backlog Grooming and Sprint routines based on fieldwork. Using interviews, fieldnotes, and documents, we identified regularly recurring sequences of actions and expressions of shared understanding through in-vivo coding. We triangulated data sources and aligned our analysis with Feldman's and Pentland's (2003, p. 95) definition of routines as 'repetitive, recognizable patterns of interdependent actions, carried out by multiple actors.'

Step 2: Analyzing the content of organizing. While the typical action patterns described *how* work was organized procedurally, the content of the work—*what* had to be done—was captured in the form of issues (e.g., bugs or new features). Some issues were simple, such as adding texts; others were complex, like enabling users to manipulate tabular files or operate hardware remotely. To test whether there were patterns in *how* these issues were solved, we applied sequence analysis to 481 issues (Mahringer & Pentland, 2021). Of these, 447 sequences were unique—suggesting no clear sequence in resolving issues. Interviews supported this

finding. One developer noted: ‘there are no technical patterns because the issues are so different,’ though he referred to consistent ‘organizational patterns’ in the routines. The Product Owner similarly said, ‘it depends on the content […] there’s no pattern where I’d say, that’s how we solve it.’ Coherence lay in the procedural steps (how), not in the work content (what). Thus, the team did not know in advance which sequence of actions would resolve a particular issue.

Step 3: Unpacking how actors relate procedural and content dimensions. Given Team Alpha’s task was to develop software (i.e., content), we examined how actors oriented to typical action patterns to guide their actions. We constructed detailed narratives from audio recordings, fieldnotes, and documents to capture what actors did and said as they addressed issues in meetings. The resulting 100 pages of narrative enabled close analysis of 30 routine performances—ranging from short interactions to sessions over an hour long, involving both complex and less complex issues.

The first author coded and visualized these narratives, while the co-authors reviewed and challenged interpretations. This iterative process allowed us to understand how actors oriented to their typical action patterns to proceed. For instance, when stuck, the team often advanced to the next familiar step from the routine’s typical action pattern. These patterns did not prescribe what to do, but offered a guiding direction. Inspired by the concept of paths, wayfinding, and by temporality as flow, we conceptualized this phenomenon as *provisional directionality*: a tentative realm of possible paths that actors can enact to proceed. We also identified three mechanisms through which the team used typical action patterns to (re)create provisional directionality.

Findings A: Situations in which Team Alpha struggled to proceed

We show our findings in two parts. In Part A, we show how Team Alpha struggled to proceed with their actions. Specifically, these situations were characterized by either no possible paths

to proceed, too many paths, or fewer paths that pointed in different directions. In Part B, we show how Team Alpha dealt with these situations by (re)creating provisional directionality.

Several factors made work situations challenging for Team Alpha. One aspect was the difficulty of specifying precisely what clients needed. For example, a developer from Team Alpha described how the apparently simple problem of providing a 'wizard'—a functionality in AlphaSoft that enabled clients to configure certain parameters of the parts to be processed—turned out to be more complex than assumed:

> *The users want a wizard, that's the requirement […] Now we start to ask: Should the wizard retrieve parameters in a predefined sequence? Should parameters vary? What about texts? Do you want to go back and forth [in the wizard]? Do you want to abort [the process]? Should it recognize if you enter an invalid value? […] Do you want to insert pictures? Should it be possible to open a second wizard?* (Interview, Developer 7)

Even when the requirements were clear, there could be many possibilities for addressing them. As Developer 7 noted in an interview, 'for each problem there are many possible solutions.' Moreover, the best path to solve an issue was often unclear, as adapting software code could influence other parts of the code in unpredictable ways: 'When you have more complex, new functionalities you discover many new points where you must clarify what is wrong? […] Did new bugs sneak in?' (Interview, Scrum Master).

We now show Team Alpha's struggles to proceed. In this situation, the team defined how to allow the operators of machines to digitally monitor the physical functioning of machine doors. In a Refinement meeting—aimed at adding clarity to those issues that were later addressed in a Sprint—Team Alpha tried to create a button that indicated different door states (i.e., open, closed, locked). As the following excerpt from the meeting shows, the more the team members investigated the issue, the less clear it became how to proceed.

Vignette A1. Developer 2 tried to describe the states that the buttons used to monitor doors should indicate: '[The button can have] two states: everything closed and locked.' Developer 7 added, 'which would be the best possible state.' The Product Owner suggested an alternative state: 'At least one [door] unlocked and at least one [door] open, right?' He began to enact the path of comparing the buttons with other functions in AlphaSoft: 'Like the states we have [in another place].' The Scrum Master then chimed in with a new path of defining color schemes: 'Wait! How do we display [each state]? Only [with an] icon or also [with a different] color?' Developer 8 tried to offer a solution: 'We could display a line above the icon, as we do [in another feature of AlphaSoft], right?' but the Product Owner said, 'then the colors are inconsistent. Or we must find another way [to indicate each status].' Developer 3 confirmed 'that's too many colors,' and the Product Owner added, 'I think it is similar [to other icons]. There, everything is green, closed and locked.'

The paths that the team had created included defining which states the functionality should incorporate, which icons should be used to display those states, whether and how color schemes should be used, and how consistent these aspects were with other functionalities of AlphaSoft. This multitude of possible paths overwhelmed the team, and it became increasingly unclear how to proceed. Developer 5 was visibly confused, asking 'can you dictate [word by word what I should write into the issue description in Jira]?'

The team tried to establish clarity. The Product Owner said, 'um, Ok, at least one [door] unlocked, and all doors closed, or at least one [door] open.' Developer 5 said, 'um, can you repeat that?' and Developer 7 replied, 'open, closed, locked.' This prompted Developer 5 to laugh ironically and say, 'I see we all agree.' Developer 5's impression that the situation was confusing was confirmed by the Scrum Master, who said, 'I don't understand it either.' Team Alpha briefly tried to further define the button, but in vain. They fell silent and then began to laugh as they realized they were stuck.

Analysis A1. As this vignette shows, it had become increasingly unclear which paths were relevant to clarify the door-handling issue and prepare it for the Sprint that usually followed Refinement. This was because Team Alpha had created multiple possible paths to proceed (e.g., defining states, icons, color schemes; aligning with other functions), but it was unclear which of those possible paths to enact. Other times, we observed situations in which there were fewer paths but they pointed in different directions. Our analysis indicates that actors often oriented to the typical action patterns of the Scrum routines to proceed in those situations. But how did they accomplish this? We unravel this in the next section.

Findings B: (Re)creating provisional directionality

In this section, we show how Team Alpha's members oriented to the typical action patterns of Scrum routines to proceed. We identify provisional directionality—a tentative realm of possible paths that actors can enact to proceed—as the key mechanism enabling such orientation. Furthermore, we distinguish three patterning mechanisms through which provisional directionality is (re)created through routines: (1) initiating possible paths, (2) including or excluding possible paths, and (3) reorienting possible paths. Each of the following subsections focuses on one of these mechanisms. We begin each with a vignette illustrating a specific routine performance, followed by its analysis. Where helpful for contextual understanding, we clarify the typical action patterns associated with the routine.

The findings focus on a particular issue that Team Alpha engaged with: How the users of AlphaSoft could use and edit tabular files. This table editor issue was one of the most complex issues encountered by Team Alpha during the fieldwork. The team addressed it across numerous Sprints over several months. The complexity of the issue originated from the clients' unclear articulation of their functional requirements, the requirement that certain functions interacted with deeper software layers, the challenge of meeting the needs of two projects with

a single solution, and the numerous options for implementing the table editor with no clear indication of the optimal choice upfront.

(Re)creating provisional directionality by initiating possible paths

Vignette B1. Team Alpha was informed that two different company projects, Project Basic and Project Complex, would require AlphaSoft to handle tabular files. However, the projects had markedly different requirements. Project Basic required editing simple files, which contained, say, the numeric coordinates of hardware component points targeted by a laser-cutter. Project Complex, in contrast, required thousands of cells and specific functions to be handled. Therefore, the Product Owner created an issue in Jira during the performance of the Backlog Grooming Routine, tasking the team to assess how they could offer a functionality that enabled both projects to handle tabular files. The issue was scheduled for the Refinement Routine, which typically followed the Backlog Grooming Routine and prepared issues for the subsequent Planning 1 Routine.

The core tasks of the Refinement Routine were to (a) assess what needed to be done to resolve each issue, and (b) estimate each issue's complexity. Accordingly, the typical action pattern of the Refinement Routine included three steps that are important for our analysis: (1) the Product Owner presented to the team an issue for refinement; (2) the team clarified (and documented) aspects that needed to be addressed to satisfy the clients' requirements, and which influenced complexity; (3) the developers estimated the complexity of resolving the issue.

After refining several other issues, the Product Owner opened the table editor issue in Jira, which included a plethora of information. The atmosphere instantaneously changed. Team members remembered their past experiences with projects Basic and Complex, yet no one seemed sure how to proceed. Developer 3 muttered, 'oh boy,' while Developer 2 let out a prolonged, uncertain grunt before asking, 'is this for Project Basic?' As was typical for the

Refinement Routine, the Product Owner recalled actions already taken which were relevant to clarifying the current issue:

> *Yes, this is for Project Complex and Project Basic. [...] Here, the question might be, whether we can somehow bring these two requirements together. In Project Complex [the clients use tables with], I don't know, 3000 […] cells and in Project Basic [the clients use] fewer, hopefully. I found some external tools that you could look at [in the Sprint].*

He scrolled down to a comment he had created for the table editor issue in Jira, in which he had listed three external tools that could be integrated into AlphaSoft as table editor functionalities, emphasizing that there were many other tools on the market that they could examine as well:

> *There are countless [tools, that's why I wrote] dot, dot, dot [...] you need to do some research [to solve the issue]. I also see a difference between the requirements of Project Complex and Basic. [...] In Project Complex they [process] I don't know, 3000 cells, and in Project Basic fewer. What do you think?*

The Product Owner's recall of prior actions initiated possible paths (a provisional directionality) that the team could enact to refine the table editor issue in the present situation. This provisional directionality included possible paths, such as clarifying relevant details related to the requirements of projects Basic and Complex (e.g., processing 3000 cells), clarifying information required to assess the complexity of combining the projects in one solution, and discussing the relevance of examining third-party tools.

Developer 3 enacted a path entailed in this provisional directionality when he started to define the implications of handling large tables that included 3000 cells. He wondered whether they also had to make the table editor usable on a touch screen (i.e., touchability):

Developer 3: ‘How about touchability? You just mentioned 3000 cells. If you imagine that, maybe […] the problem is that you have 50 columns, but the columns must be wide enough for the touch function [to work], so that you only see four columns [on your screen], you need [to be able] to scroll horizontally.’

Product Owner: ‘For Project Complex it doesn’t make sense to make [the table editor] touchable, because you’d have to make [the tables] too small to see everything at a glance. This is what the clients on Project Complex require; to see as much as possible at once.’

Developer 3: ‘And if somebody starts scrolling on the touch screen?’

Product Owner: ‘Then he probably pushes it [the focal part of the table] out of the screen.’

Developer 2: ‘Do [the clients in Project Complex] have larger screens [than the screens used in other projects]?’

Product Owner: ‘[…] They usually have larger screens.’

Developer 2: ‘So we have to adapt AlphaSoft anyway.’

The developers realized that the bigger screens in Project Complex required them to adapt the screen resolution of AlphaSoft. Within seconds, the situation had changed from refining the table editor issue to defining the screen resolution for Project Complex in more general terms. The Product Owner, however, felt that the team had deviated from the main topic. He gave the developers a serious look, and again initiated possible paths: ‘as I’ve already mentioned: [... in the Backlog Grooming Routine I realized that] we should integrate both requirements. The small requirement of Project Basic and the mammoth requirement of Project Complex, which includes [...] thousands of cells.’

Once again, this created directionality. Because the Product Owner had noted the differences between the projects, the team began to elaborate on those differences. Developer 3 mentioned that one specific requirement of the clients in Project Complex was to ‘insert buttons and checkpoints,’ which was not required in Project Basic.

Analysis B1. The vignette shows one of the three mechanisms we identified: *initiating possible paths*, which means that actors orient to the typical action pattern of the routine at hand to recall prior actions that suggest a starting point for possible paths relevant to the present situation. The actions being recalled could be more or less distant from present actions, depending on what was most relevant to the current situation. In the vignette, the Product Owner oriented to the typical Refinement pattern, which not only made it clear that he should present the table editor issue, but also that the paths he initiated should enable the team to clarify and document what needed to be done to satisfy the client's requirements and estimate complexity in the present situation. This enabled him to recall prior actions, such as how he had defined the clients' requirements, and how he had identified external tools that might be useful to address those requirements in the prior performance of the Backlog Grooming Routine. While the Product Owner often initiated possible paths, this mechanism was a collective accomplishment. Instance C1 in Table 1, for example, shows another case (from another issue) where the team was able to initiate possible paths even though the Product Owner was absent.

Initiating possible paths was one of the three mechanisms through which the team created *provisional directionality*. In the beginning of the vignette, the developers' utterances (e.g., a prolonged grunt) signaled that the developers lacked a clear idea about how to proceed. In this situation, initiating possible paths created a provisional directionality, which provided a common starting point to refine the table editor issue. This provisional directionality included possible paths such as clarifying relevant details related to the requirements of projects Basic and Complex, clarifying information required to assess the complexity of combining the projects in one solution, and the relevance and scope of examining third-party tools. When the Product Owner initiated possible paths, he did not know what the team would ask or clarify, nor did he ask specific questions or propose specific paths that might constrain the emergence

of new insights. Rather, he created a directionality that was coherent enough to ensure that the team knew how they could proceed.

The vignette also shows how the team *enacted* two different paths incorporated in this provisional directionality. Note that while provisional directionality is a realm of possible paths, actors can (usually) only enact one of these paths at a time. In the first instance in the vignette, Developer 3 started to elaborate the relevance of implementing touch functionality. Had the Product Owner not mentioned that one of the projects required editing 3000 cells when he initiated possible paths, Developer 3 might not have enacted this path—as indicated by his statement, 'How about touchability? You just mentioned 3000 cells.' It would also be hard to believe that the Product Owner already knew that Developer 3 would enact this path when he initiated possible paths. In the second instance in the vignette, the developers enacted another path: they defined the different requirements of the clients in projects Basic and Complex.

The directionality that actors created was *provisional* because it was impossible for team members to predict a priori where enacting a path would lead, and which paths might be relevant some moments later. In the vignette, the Product Owner created a provisional directionality that incorporated possible paths related to client requirements. However, the developers quickly drifted to the path of defining screen resolution in Project Complex, which was not immediately relevant for clarifying the issue. Hence, directionality blurred as the team members took further actions.

The language that team members used in the vignette signals the provisional nature of actions. The Product Owner, for instance, emphasized that there were many other third-party tools that could be assessed (implied by 'dot, dot, dot'), while his statement that 'you need to do some research' stressed that actions were by no means prescribed. Moreover, Developer 3 used phrases like 'if you imagine that, maybe,' which shows that he was not referring to a prescribed course of actions.

(Re)creating provisional directionality by including or excluding possible paths

We now introduce the second mechanism through which actors (re)created provisional directionality: *including or excluding possible paths*. Vignette B2 illustrates how the Refinement performance introduced in Vignette B1 continued to evolve.

Vignette B2. Team Alpha was refining the table editor issue, which involved differentiating between projects Basic and Complex. While elaborating on project differences, Developer 3 enacted a path related to possible technical solutions, but concluded that the requirements of the two projects were too different to be met by a single solution. He argued that 'in Project Basic you only need to edit a CSV file [i.e., a simple file format used in spreadsheets], but the other [project] is, um, [more complex]. I think we have to deal with this separately. [The projects] don't have much in common.' Developer 2 agreed: 'Yes, I fear so.'

Given that the purpose of the issue was to identify a solution that could cover the requirements of both projects, the developers' conclusion that these requirements were incompatible and that a single solution was not feasible was itself a form of resolution. The Product Owner, however, felt that the path of achieving a practical conclusion for the issue should be pursued later in the Sprint. What was currently required was for the team to clarify the details of the issue and estimate its complexity, as indicated by the typical Refinement pattern. Hence, paths pointed in different directions. Thus, the Product Owner noted that meeting the needs of both projects in practical terms would have to be accomplished later, when the team came to resolve the issue in the Sprint: 'This [path] is the question you have to answer [in the Sprint].' Developer 2 glanced at the Product Owner and nodded: 'So this [path] is what we should do when we resolve this issue [in the Sprint]?' The Product Owner confirmed this statement.

By orienting to the typical Sprint pattern, the Product Owner had excluded the path of coming to a conclusion for the table editor issue from the present situation. This led the team

to enact other paths that were relevant for the situation. Developer 3 asked, ‘a question is what [functions the table editor] requires beyond simply entering and displaying values?’ He pointed out that a counting function could be included in the table editor.

Analysis B2. *Including or excluding possible paths* is the second mechanism through which actors (re)created provisional directionality. By this mechanism, we mean that actors signify whether a specific path should or should not be enacted in a particular situation by connecting it with the typical action pattern of the present or another routine. In the vignette, the developers pursued the path of conclusion (agreeing that the projects were incompatible). The Product Owner, however, felt that this path was at odds with the current Refinement performance, as solving an issue should be carried out later in the Sprint. Note that Refinement is typically followed by Planning 1 and 2, which in turn are followed by the Sprint. He therefore excluded the path of conclusion from the situation at hand by remarking that the developers had to enact it later, in the Sprint. Here, he was orienting to another typical action pattern (i.e., the Sprint) to exclude a possible path, but we also observed cases in which actors oriented to the typical action pattern of the present routine to include or exclude paths (see instances C2 and C3 in Table 1).

In excluding a path, the Product Owner had (re)created provisional directionality. Once the team realized this, it could focus on a consolidated realm of possible paths. In the vignette, Developer 3 *enacted a path* contained in the consolidated directionality when he elaborated which functionality the table editor required, which led him to define the possibility of a counting function. The counting function was relevant for the provisional directionality because the functions the table editor had to include influenced the complexity of the issue, and estimating complexity was a step in the typical Refinement pattern.

(Re)creating provisional directionality by reorienting possible paths

In this section, we introduce *reorienting possible paths*, the third patterning mechanism through which provisional directionality was (re)created. This patterning mechanism was often enacted when routine participants got stuck.

Vignette B3. After Team Alpha had added the table editor issue to the upcoming Sprint (Planning 1), they met for Planning 2. The core task of the Planning 2 Routine was to assess how to resolve the issues planned for the upcoming Sprint from a technical perspective. The following steps of the typical Planning 2 pattern are relevant to our analysis: (1) the developers select an issue from the Sprint backlog in Jira, (2) they clarify how they would technically resolve the issue during the Sprint, and (3) if necessary, subdivide the issue, and document it in Jira. Then they proceed with the next issue, repeating the procedure until they have addressed every issue planned for the forthcoming Sprint.

In the Planning 2 meeting, the developers were enacting paths they had to consider when addressing the table editor issue. Developer 4, for instance, mentioned that he knew various third-party tools to implement the table editor. Developer 3 brought up that 'the [graphical] design [of the table editor also] has to be adapted to AlphaSoft's style.' Developer 4 said that it was difficult to understand the design limitations of third-party tools, asking 'how much should we analyze each [of the tools]?' Developer 4 then enacted another path by saying that there were substantial differences between projects Basic and Complex: '[Project Basic] is very simple, the user can only enter values [...] and the other one is for Project Complex. They want to edit processual variables [...] and they should be able to see everything at a glance.' Developer 3 enacted yet another path, asking whether existing code components should be used to implement this complex table editor, which prompted Developer 4 to further examine the specific types of code required. Over the next few minutes, the developers came up with further

paths, such as defining output formats, how to connect the table editor to deeper software layers, and defining requirements in terms of copy-paste and counting functions.

Despite the experience of the developers, the multiplicity of paths caused confusion. Developer 4, in a desperate tone, summarized the paths that the team had brought up in rapid succession: ‘but this flexibility, writing true here and ‘bubu’ there and 1, 2, 3, and things like that, and then is it nationalization, is it a dot or comma, and stuff like that, um, um. I don’t know how well described is that stuff, um?’ The Scrum Master and Developer 7 replied together with ‘not at all,’ raising a laugh from the team. As a result of the developers’ actions, it was not clear how to proceed.

After some moments of silence, Developer 8 oriented to the typical Planning 2 pattern by suggesting moving on to the step that typically followed next: splitting the issue into smaller parts and documenting these in Jira. He said, ‘Shall we talk about the two table editors [for projects Basic and Complex] to start with, so that everybody [understands the differences]?’ He then added a relevant comment in Jira, and started to list what each project required of the table editor. Moving on to the next step guided the team to discuss what functionalities Project Basic required the table editor to provide. Developer 7, for instance, asked whether the table editor functionality required scrolling, which his colleagues confirmed. Then, they documented that the clients should be able to define headers and the number of columns, as well as the option of loading and saving external files.

Analysis B3. The vignette shows a situation in which there were many possible paths, but it was unclear how to proceed. The possible paths included definition of relevant third-party tools, the graphical design, types of code required, output formats, how the table editor would link to deeper software layers, and functional requirements in terms of copy-pasting and counting. The comments of the Scrum Master and Developer 7 that all these things were not

yet defined, and the ensuing laughter of the team, signaled uncertainty. The subsequent silence reveals that it was unclear which paths the team could enact.

Developer 8, therefore, used the typical Planning 2 pattern to *reorient possible paths*. This mechanism means that actors move to the next step in the typical action pattern of the present routine, which orients actions to a different set of possible paths. In the vignette, Developer 8 suggested that they could move to the step that typically followed the preliminary discussion of how to resolve the issue from a technical point of view; namely, splitting the issue into smaller parts. This helped the team to focus on paths that defined the functionality of the table editor for Project Basic. This focus was possible because the team had split the issue into two subproblems and knew that the next step was to document the requirements for Project Basic.

Table 1 presents additional empirical evidence from the table editor issue and another issue that Team Alpha handled, i.e., the door-handling functionality. The table depicts variations in the mechanisms we identified. Specifically, example C1 illustrates how *initiating possible paths* is not attributable to a single individual but emerges as a collective accomplishment. Examples C2 and C3 show how *including or excluding possible paths* can occur with reference to the present routine's typical action pattern.

Empirical evidence	Analysis
C1) Example illustrating how *initiating possible paths* represents a collective accomplishment	
Refinement (door-handling functionality): The developers opened the door-handling issue. However, they were unable to recognize which previous actions would help them clarify the issue and assess its complexity. Usually, the Product Owner would initiate possible paths but he was not present at that meeting. Developer 5 tried to define the subject of the issue: 'Um, so, you want to have this [functionality in AlphaSoft], um—but it already exists [nested in the software]?' He scrolled through the issue in Jira but noted that the information provided was not helpful. Developer 2 shrugged. Developer 8 now glanced at Developer 7, as he vaguely remembered a previous action that might be relevant to this issue, and asked: 'Didn't you have a related issue with [a client]?' Developer 7 replied, 'Oh yes, that's the bug [linked to this issue in Jira].' Developer 8 asked what the code for the door-handling functionality nested in the software looked like. Developer 5 opened a visualization of the required function attached to the issue in Jira. Simultaneously, Developer 7 started to recall relevant details, explaining that 'the function should have multiple door circuits.' As the situation became clearer, Developer 7 mentioned a certain kind of code that would be required to implement these door circuits. This prompted Developer 8 to define the variable types through which the door icons showed whether machine doors were open or closed. Building on this, Developers 2 and 7 started to discuss how the icons could be connected to deeper layers of the software to know the current status of the machine doors.	***Situation***: Because the Product Owner could not attend the meeting, the team struggled to recall previous actions as usual, and thus was not immediately able to initiate possible paths. ***Initiating possible paths:*** As the developers read the description in Jira and brought in the experiences of different individuals, they eventually managed to collectively initiate possible paths. The resulting provisional directionality included paths such as defining the specific functionality required (i.e., each icon represented a machine door, and multiple icons were organized in door circuits), and how AlphaSoft could gather information about the physical states (e.g., open versus closed) of machine doors. These paths were important to get an impression of what needed to be done to solve the issue, which influenced its complexity; a relevant matter in the Refinement Routine. ***Consequence:*** The developers enacted a path related to code and the types of variables needed to implement the door circuits.
C2) Example illustrating how actors *include a possible path* by connecting it with the present routine's pattern	
Planning 2 (table editor): The team defined at length the specific technical details of the table editor, iterating back and forth between Project Basic and Complex. Developer 2, however, said, 'but [I thought] this issue is just about the table editor for Project Basic?' He referred to the prior Planning 1 performance, in which the team had decided that they should figure out the table editor for Project Basic in the first step, and postpone aspects related to Project Complex to a subsequent Sprint. The Scrum Master explained that it was important to 'demarcate the requirements of both projects [to complete the Planning 2 performance]. My impression was that this morning [in the Planning 1 performance] nobody had a plan, right?' Developer 2 jokingly quoted from the Planning 1 performance: 'we will clarify it this noon [in the Planning 2 performance].' The Scrum Master laughed, 'and that's what we are currently doing [as typical for this routine].' Hence, the team proceeded with demarcating the requirements for both projects.	***Situation:*** It was not clear whether the developers had to pursue paths related to both projects or only to Project Basic. Paths related to Project Complex were somehow at odds with how they had defined the issue in the Planning 1 Routine, in which they had put an explicit focus on Project Basic. ***Including possible path:*** The Scrum Master included the path of demarcating both projects by orienting to the typical Planning 2 pattern. In this pattern, the developers clarified how to resolve the issue from a technical point of view, which in this case required understanding the differences of the two projects. These differences would allow them to better understand what had to be done to resolve the issue for Project Basic. Because it was important for the team to have clarity about issues when they started the Sprint, not having a 'plan' in the Planning 2 performance was problematic and had to be dealt with. ***Consequence:*** The team further enacted the path of demarcating the requirements of both projects.
C3) Example illustrating how actors *exclude a possible path* by connecting it with the present routine's pattern	
Refinement (door-handling functionality): As the team was defining the door-handling functionality, a mockup that graphically showed the door-handling function was presented. However, as the mockup also showed other aspects of AlphaSoft, the team started to enact a path related to 'something else, but still related to the mockup' (Developer 2)—a boxlike feature in AlphaSoft, which was not relevant for the door-handling functionality. The Product Owner noted, 'I admit that this mockup is not optimal. Now we discuss everything else, but not [the door-handling functionality].' The team laughed, and focused on examining the design of buttons included in the door-handling functionality.	***Situation:*** As the team discussed the mockup to refine the door-handling functionality, they started to enact the path of defining the boxlike feature in AlphaSoft. ***Excluding possible path:*** The Product Owner reminded the team that refining door-handling was the current task. This excluded the path of further defining the boxlike feature which, although included in AlphaSoft, was not relevant for the problem of enabling the clients to digitally handle doors. ***Consequence:*** Actors enacted another path related to defining the design of buttons, which was involved in the current provisional directionality.

Table 1. Additional evidence for (re)creating provisional directionality.

Discussion

Provisional directionality as a new concept for routine dynamics

We empirically examined how routine participants orient to typical action patterns to prospectively guide their actions. This inquiry led us to develop the concept of provisional directionality—a tentative realm of possible paths that actors can enact to proceed—along with the mechanisms through which it is (re)created. Figure 1 highlights three key characteristics of provisional directionality, which jointly underscore its conceptual novelty: (1) its situated and collective accomplishment, (2) its dynamic evolution, and (3) its orchestration of possibilities.

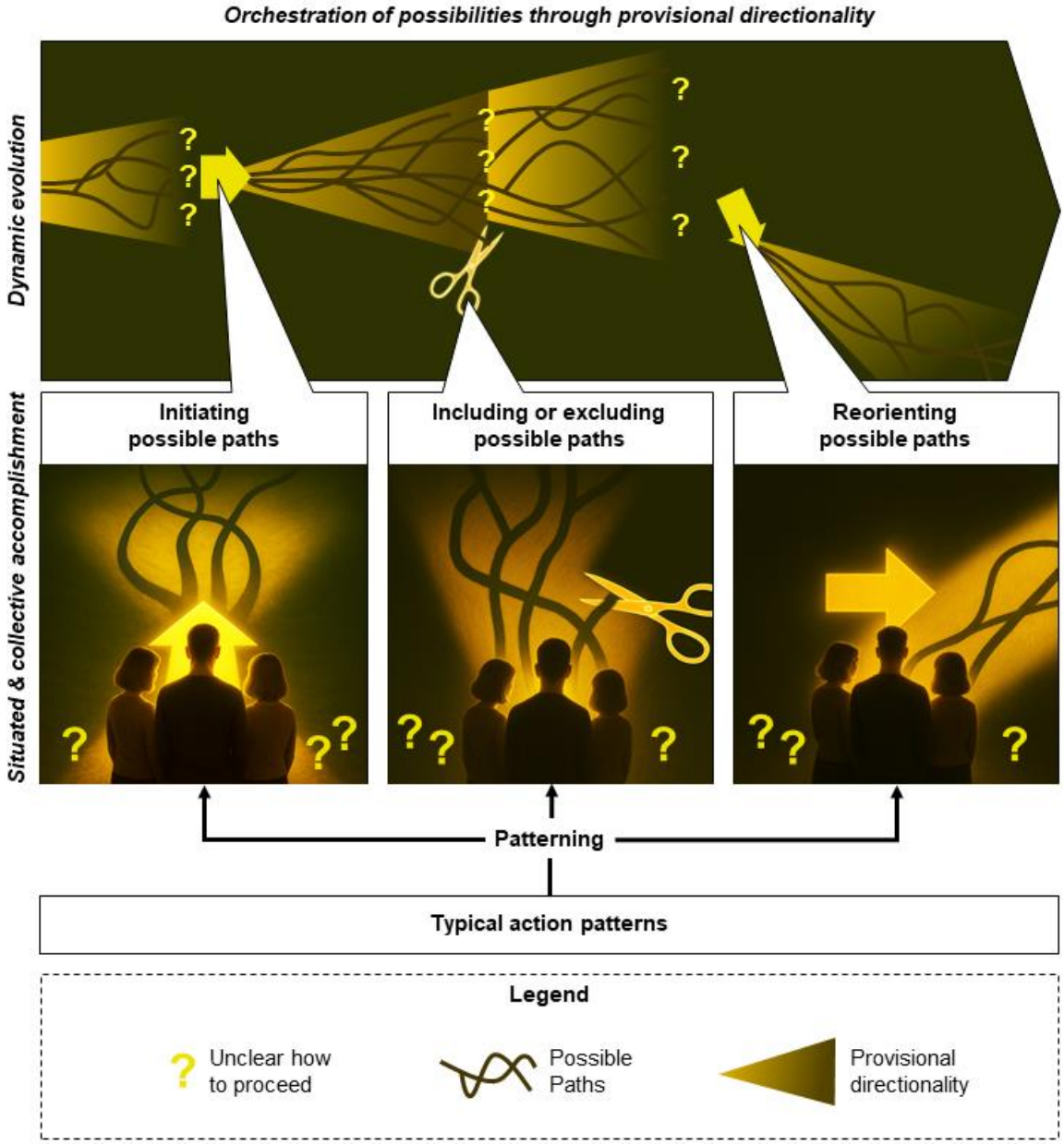

Figure 1. Process model of provisional directionality.

The bottom and middle parts of Figure 1 show that provisional directionality is a *situated and collective accomplishment*, not an inherent property of the situation nor an abstract feature of typical action patterns. This becomes evident in how actors orient to typical action patterns to (re)create provisional directionality in situ. These acts reflect 'the impact of patterns on actions' (Feldman & Pentland, 2022, p. 850), and thus represent instances of patterning.

The middle part of Figure 1 shows three different mechanisms of (re)creating provisional directionality in situ. First, initiating possible paths means that actors orient to the typical action pattern of the routine at hand to recall prior actions that suggest a starting point for possible paths relevant to the present situation. Second, including or excluding possible paths means that actors signify whether a specific path should or should not be enacted in a particular situation by connecting it with the typical action pattern of the present or another routine. Third, reorienting possible paths means that actors move on to the next step in the typical action pattern of the present routine, which orients actions to a different set of possible paths.

The upper part of Figure 1 shows how provisional directionality *dynamically evolves* over time. It illustrates how actors iterate between situations where possible paths are lacking (indicated by the question marks) and those where a provisional directionality exists. The term 'provisional' underscores this continuous and open-ended shaping of the way forward. In this vein, provisional directionality frames routine performances as a mode of wayfinding (Bouty et al., 2019; Chia & Holt, 2009; Ingold, 2011; Sminia & Corvalán, in press). Rather than pursuing fixed end states, actors navigate into an unfolding future. Depending on the situation, they create and recreate provisional directionality, enabling them to go with the flow.

Overall, Figure 1 highlights that provisional directionality centers on the *orchestration of possibilities* (Feldman & Sengupta, 2020; Pentland et al., 2020). Both the upper and middle part of the figure illustrate provisional directionality as cones that illuminate a range of possible paths actors could take to proceed. Within these cones, different possible paths come into view.

By (re)creating provisional directionality, actors render possibilities intelligible and actionable in situations in which it is unclear how to proceed, which is indicated by the question marks in the figure.

When and how often routine participants (re)create provisional directionality depends on how often they encounter situations where it is unclear how to proceed. We observed provisional directionality in a setting where such situations arose frequently—a hallmark of generative contexts like product and service innovation, software development, and emergent strategizing, where neither outcomes nor suitable paths are known in advance. In more stable and repetitive settings, by contrast, the way forward is usually clearer. Yet, most real-world cases fall somewhere between these extremes. Moreover, because we conceptualized provisional directionality as a collective accomplishment, it might be less important in guiding individual action. In contexts where coordination is mediated by interfaces rather than real-time interaction, provisional directionality may also be obstructed or absent. In addition, Team Alpha was very proficient with the typical action patterns, but we would expect that provisional directionality is more difficult to (re)create when people have divergent understandings of typical action patterns.

Theoretical contributions to routine dynamics research

Provisional directionality as the prospective dimension of patterning in routines. Although previous research on routines has demonstrated that patterning involves the impact of typical action patterns on actions (Danner-Schröder & Geiger, 2016; Feldman & Pentland, 2022; LeBaron et al., 2016), it has remained unclear how these patterns guide actions prospectively. This study contributes to the routine dynamics literature by introducing the concept of provisional directionality as the prospective dimension of patterning in routines. Unpacking this prospective dimension is crucial, as it enables us to theorize and conceptualize how routine participants experience guiding from within (Wegener & Lorino, 2020).

Provisional directionality builds theory specifying the argument of Feldman and Pentland (2003, p. 105) that typical action patterns influence routine performances 'prospectively, as a guide to what actions ought to be taken.' Our findings show that typical action patterns do not prescribe specific paths, but instead allow actors to create a realm of possible paths—a provisional directionality. Guidance, in this sense, is not about what *ought* to be done but about what *could* be done. While some paths are excluded through patterning, which means that they ought not to be performed, many paths remain provisionally open.

This extension of the patterning concept is important because routines have long been understood as constraining or prescribing actions (Feldman et al., 2021). Although routine dynamics research has challenged this view, the concept of provisional directionality pushes further by theorizing and conceptualizing how routines guide subsequent actions without collapsing into prescription. It clarifies how actors orient to typical action patterns to prospectively act in a way that retains both coherence and open-endedness.

Unpacking provisional directionality as the prospective dimension of patterning also extends the notion of paths. Scholars often examine how paths develop over longer periods of time from an outside perspective (Goh & Pentland, 2019; Karnøe & Garud, 2012). The metaphor of the grassy meadow (Pentland et al., 2020), for example, considers the form and development of the meadow from a bird's-eye perspective. In contrast, our insights about provisional directionality can be interpreted as the view that actors have at each intersection as they experience and shape the unfolding of the meadow from within. Thus, provisional directionality shows how patterning orchestrates paths from within. This approach allows us to better understand how the structure of the grassy meadow emerges through a series of situated actions. Moreover, we not only show how people enact actual paths (Danner-Schröder & Ostermann, 2022), but also introduce provisional directionality as a conceptual tool for theorizing possible paths.

Provisional directionality enacts temporality as flow. Temporality is a key aspect of routine dynamics (Mahringer et al., 2024; Turner & Rindova, 2018, 2021). Prior research has often focused on distinct temporal orientations to the past, present, or future that participants adopt (Howard-Grenville, 2005) or that are designed into typical action patterns (Ritter et al., 2024). Rather than viewing temporality as discrete orientations, however, provisional directionality conceptualizes and theorizes temporality as a continuous flow (Baygi et al., 2021; Blagoev et al., 2024; Skade, 2025; Turner & Rindova, 2021; Xu & Carlile, 2024), where past actions color present enactments of the near future. It highlights how routine participants enact the near future in the present, by orienting to typical action patterns to establish direction. In doing so, they actively shape how the future unfolds in real time. Provisional directionality thereby advances a strong process ontology on temporality in routine dynamics. In this view, the near future is not a predefined horizon to be integrated or demarcated, but a provisional construct that unfolds dynamically through situated acts of patterning.

The three patterning mechanisms we identified add nuance to how routine participants experience and enact temporality as flow. When initiating possible paths, actors draw on past occurrences to shape the present in ways that enable near-future actions. Including or excluding possible paths shape how the present extends into the unfolding future by directing attention and effort to some paths but not others. In reorienting possible paths, they let go of current paths and identify a new provisional directionality to move forward. Together, initiating, including or excluding, and reorienting possible paths are the ways through which actors make future possibilities actionable in the present.

Unpacking scripted routines when it is unclear how to proceed. Routine dynamics research has explored how routines guide actions, either by prescribing sequences of actions (Danner-Schröder & Geiger, 2016; LeBaron et al., 2016) or by fostering flexible recombination and adjustment (Danner-Schröder & Geiger, 2016; Dönmez et al., 2016; Pentland & Rueter,

1994; Sailer et al., 2023; Turner & Rindova, 2012). The concept of provisional directionality diverges from both of these ways by revealing how scripted routines (Jarzabkowski et al., 2017) enable actors to move forward when it is unclear how to proceed.

First, provisional directionality differs from prescribing sequence, because it does not constrain actors to one specific path that ought to be performed. Instead, it offers a bounded realm of possible paths that allows for open-ended adjustment while retaining coherence. This is particularly important in generative contexts, where prescribing a path would prematurely narrow the scope for creativity or learning.

Second, provisional directionality goes beyond the view that guiding means flexible recombination and adjustment. The notion of selecting from a repertoire helps explain how actors know which actions fit a particular situation. However, it does not account for how they orchestrate different possible paths in situ, and it does not explain how actors draw on the typical action patterns characterizing scripted routines to generate directions for moving forward (Jarzabkowski et al., 2017). Provisional directionality helps to appreciate how heavily scripted routines guide action in coherent yet open-ended ways at the same time.

Avenues for future research

In this section, we outline avenues for refining the concept of provisional directionality, and we discuss how routines research can inform work in generative contexts.

Conceptual refinements of provisional directionality. We see different ways to further refine and extend the notion of provisional directionality. First, research could examine power and conflict (Feldman & Pentland, 2003; Salvato & Rerup, 2018) in relation to provisional directionality. Our findings show that the mechanisms of initiating, including or excluding, and reorienting possible paths enable actors to influence how they move forward. While initiating and reorienting are softer in tone, the including or excluding mechanism is more directive, focusing on specific paths. These can be understood as forms of enacted power, but more

research is required to understand this aspect: When is provisional directionality accepted or challenged? What causes possible paths to be acceptable in some situations (signaling a broader directionality) or contested in others (signaling a narrower directionality)? How does the legitimacy of those enacting provisional directionality influence whether it is accepted, adapted, or resisted?

Second, provisional directionality could help to better understand the dynamics of routine ecologies and clusters (Dönmez et al., 2016; Sailer et al., 2023). Provisional directionality may open paths that point to other routines, and thus enact connections among routines (Turner & Rindova, 2018). Under which conditions does provisional directionality involve other routines? Can different kinds of interfaces between routines or routine clusters (Mahringer & Danner-Schröder, in press) hinder or enable such connections?

Third, research could broaden the temporal scope. We have focused on how provisional directionality makes the next steps intelligible (i.e., near future), but it might be interesting to also link series of provisional directionality enactments to distant future outcomes (Blagoev et al., 2024; Xu & Carlile, 2024). This would allow scholars to link provisional directionality to societal concerns such as digitalization and sustainability. For instance, how does the directionality created by actors influence whether organizations become sustainable or digitized? Actors may or may not see possible paths to sustainability or digitization (Feuls et al., 2025), and even if they see such paths, they may still be experienced as unfeasible. Why are these paths contested and how can we enable actors to see other possibilities?

Routines in product and service innovation. It is well established that routines support innovation (Deken et al., 2016; Deken & Sele, 2021; Mengis et al., 2018; Salvato & Rerup, 2018). Provisional directionality explains how routines can contribute to innovation: It enables coherence, allowing the actions of different actors to interlock in generative ways; at the same time, it remains open-ended, ensuring that creative exploration is not prematurely constrained.

As actors continuously (re)create provisional directionality, they enact paths that can culminate in new products or services over time. Such outcomes are not the result of following a single predefined path, but of many instances of situated orchestration.

Our findings show that heavily scripted routines can support innovating, not by determining the content of the work (the what), but by regulating the process through which actors interact (the how), which in turn influences the content as it is enacted. Through provisional directionality, typical action patterns enable actors to move forward collectively, balancing the coherence needed for coordination with the open-endedness required for innovation. This balance would not be possible if paths were fully prescribed by typical action patterns, nor would innovation emerge if actors had no orientation. Provisional directionality thus occupies a generative middle ground: It orients actions without fully constraining possibilities.

Building on this, we propose that provisional directionality can inform research on routines and innovation: When does provisional directionality lead to novel outcomes rather than dead ends? How is provisional directionality initiated and maintained in innovation projects that unfold across organizational boundaries? What happens when multiple innovation teams generate divergent provisional directionalities?

Routines in agile software development. Provisional directionality can also inform research on routines in agile software development contexts (Baham & Hirschheim, 2022; Benlian et al., in press; Mahringer & Danner-Schröder, in press; Ritter et al., 2024). Our study underscores the importance of understanding how typical action patterns (e.g., Scrum routines) guide actions prospectively. What matters is not only the existence of typical action patterns, but how these are used in situ to create direction.

Our research encourages scholars to attend more closely to how agile methods guide action in particular situations. This perspective invites a shift from evaluating agility in terms of

adherence to certain routines toward assessing how routines are used to shape possible paths in situ. Future research might explore: How does the enactment of provisional directionality vary across routines based on different agile frameworks (e.g., Scrum, SAFe, Kanban)? What happens when developers have incongruent understandings of typical action patterns? How do distributed or remote agile teams create provisional directionality?

Routines in emergent strategizing. Research has proposed that strategies can emerge through the patterned consistencies of everyday practices, such as organizational routines (Feldman, 2025; Seidl, Grossmann-Hensel, & Jarzabkowski, 2021). Daymond, Meisiek, and Knight (2024), for instance, show that customer-centric strategies emerge through routines such as product design, marketing, and customer feedback. Research has proposed wayfinding as a concept to better account for emergent strategizing (Bouty et al., 2019; Chia & Holt, 2009; Ingold, 2011; Sminia & Corvalán, in press). Here, strategies emerge purposively from people coping with practical contingencies in everyday routines.

While the wayfinding concept is promising, it has rarely been employed empirically (Bouty et al., 2019). Our empirical inquiry suggests that provisional directionality can extend the wayfinding concept by adding a prospective dimension. While it is theoretically clear that wayfinding entails a prospective dimension—that is, movement into the yet-unknown—it has been less clear how it enables actors to move forward when there are different possibilities. Provisional directionality emphasizes this aspect, revealing how actors can dwell forward. Future research can thus examine: How does the orchestration of possibilities over time constitute, maintain, and shape emergent strategies? How do competing provisional directionalities across organizational units influence the coherence of emergent strategies? How can provisional directionality be deliberately cultivated as a capability for organizations facing persistent environmental dynamism?

Conclusion

This paper introduces provisional directionality as a central way through which typical action patterns prospectively guide actions, along with the mechanisms by which such directionality is continuously (re)created. We show that provisional directionality is essential for routine-based patterning, but other coordination mechanisms such as objects, rules, norms, bodily movements, emotions, and rhetoric may also play a role. We encourage future research to apply and further develop the notion of provisional directionality to explore how people find their way in a world that is open-ended and rife with possibilities.

Acknowledgements

We thank the members of TIC, who enabled us to observe provisional directionality in action. We are also deeply grateful to Senior Editor Paul Spee and the three anonymous reviewers for their invaluable guidance throughout the review process. In addition, we thank Anja Danner-Schröder, Martha Feldman, Daniel Gäckle, Vern Glaser, Susan Hilbolling, David M. Herold, Jennifer Howard-Grenville, Burcu Küçükkeles, Waldemar Kremser, Kuo-Ching Mei, Sunny Mosangzi Xu, Elvira Periac, Kurt Rachlitz, Claus Rerup, Martin Rost, Carlo Salvato, Kathrin Sele, David Seidl, Laura Schmiedle, Jacky Swan, and Philipp Tuertscher for their thoughtful feedback on earlier versions of this paper. In line with Sage's AI policy, we used OpenAI's ChatGPT-4 to correct spelling and grammatical errors, and to make editorial improvements to the wording of the manuscript. Additionally, the three-dimensional elements in the visual process model were created with the support of OpenAI's DALL·E 3. We carefully reviewed all AI-assisted revisions and assume full responsibility for the final content.

References

Baham, C., & Hirschheim, R. (2022). Issues, challenges, and a proposed theoretical core of agile software development research. *Information Systems Journal, 32*(1), 103–129. https://doi.org/10.1111/isj.12336

Baygi, R. M., Introna, L. D., & Hultin, L. (2021). Everything flows: Studying continuous socio-technological transformation in a fluid and dynamic digital world. *MIS Quarterly*, *45*(1), 423-452. https://doi.org/10.25300/MISQ/2021/15887

Benlian, A., Pinski, M., & Adam, M. (in press). Team-enacted use versus developer-needed use of agile practices: How perceptual (in-)congruence and team feedback-seeking shape developer well-being. *Information Systems Research*. https://doi.org/10.1287/isre.2023.0402

Birnholtz, J. P., Cohen, M. D., & Hoch, S. V. (2007). Organizational character: On the regeneration of Camp Poplar Grove. *Organization Science*, *18*(2), 315-332. https://doi.org/10.1287/orsc.1070.0248

Blagoev, B., Hernes, T., Kunisch, S., & Schultz, M. (2024). Time as a research lens: A conceptual review and research agenda. *Journal of Management*, *50*(6), 2152-2196. https://doi.org/10.1177/01492063231215032

Bouty, I., Gomez, M.-L., & Chia, R. (2019). Strategy emergence as wayfinding. *M@n@gement*, *22*(3), 438-465.

Chia, R., & Holt, R. (2009). *Strategy without design: The silent efficacy of indirect action*. Cambridge University Press.

Danner-Schröder, A., & Geiger, D. (2016). Unravelling the motor of patterning work: Toward an understanding of the microlevel dynamics of standardization and flexibility. *Organization Science*, *27*(3), 633-658. https://doi.org/10.1287/orsc.2016.1055

Danner-Schröder, A., & Ostermann, S. M. (2022). Towards a processual understanding of task complexity: Constructing task complexity in practice. *Organization Studies*, *43*(3), 437-463. https://doi.org/10.1177/0170840620941314

Daymond, J., Meisiek, S., & Knight, E. (2024). Into the customers' shoes: Multimodal practices for customer-centric strategizing. *Organization Studies*, *45*(11), 1579-1609. https://doi.org/10.1177/01708406241273792

Deken, F., Carlile, P. R., Berends, H., & Lauche, K. (2016). Generating novelty through interdependent routines: A process model of routine work. *Organization Science*, *27*(3), 659-677. https://doi.org/10.1287/orsc.2016.1051

Deken, F., & Sele, K. (2021). Innovation work and routine dynamics. In M. S. Feldman, B. T. Pentland, L. D'Adderio, K. Dittrich, C. Rerup, & D. Seidl (Eds.), *Cambridge handbook of routine dynamics* (pp. 288-300). Cambridge University Press. https://doi.org/10.1017/9781108993340.025

Dittrich, K. (2021). Ethnography and routine dynamics. In M. S. Feldman, B. T. Pentland, L. D'Adderio, K. Dittrich, C. Rerup, & D. Seidl (Eds.), *Cambridge handbook of routine dynamics* (pp. 103-129). Cambridge University Press. https://doi.org/10.1017/9781108993340.011

Dönmez, D., Grote, G., & Brusoni, S. (2016). Routine interdependencies as a source of stability and flexibility. A study of agile software development teams. *Information and Organization*, *26*(3), 63-83. https://doi.org/10.1016/j.infoandorg.2016.07.001

Feldman, M. S. (2016). Routines as process: Past, present, and future. In J. Howard-Grenville, C. Rerup, A. Langley, & H. Tsoukas (Eds.), *Organizational routines. How they are created, maintained, and changed* (pp. 23–46). Oxford University Press. https://doi.org/10.1093/acprof:oso/9780198759485.003.0002

Feldman, M. S. (2025). Routine dynamics and connections to strategy as practice. In D. Golsorkhi, L. Rouleau, D. Seidl, & E. Vaara (Eds.), *Cambridge handbook of strategy as practice* (pp. 367–380). Cambridge University Press. https://doi.org/10.1017/9781009216067.022

Feldman, M. S., & Pentland, B. T. (2003). Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly*, *48*(1), 94–118. https://doi.org/10.2307/3556620

Feldman, M. S., & Pentland, B. T. (2022). Routine dynamics: Toward a critical conversation. *Strategic Organization*, *20*(4), 846–859. https://doi.org/10.1177/14761270221130876

Feldman, M. S., Pentland, B. T., D'Adderio, L., Dittrich, K., Rerup, C., & Seidl, D. (2021). What is routine dynamics? In M. S. Feldman, B. T. Pentland, L. D'Adderio, K. Dittrich, C. Rerup, & D. Seidl (Eds.), *Cambridge handbook of routine dynamics* (pp. 1–18). Cambridge University Press. https://doi.org/10.1017/9781108993340.002

Feldman, M. S., Pentland, B. T., D'Adderio, L., & Lazaric, N. (2016). Beyond routines as things: Introduction to the special issue on routine dynamics. *Organization Science*, *27*(3), 505–513. https://doi.org/10.1287/orsc.2016.1070

Feldman, M. S., & Sengupta, P. (2020). Enacting the logic of possibility in organizations and management. *Perspectives on Public Management and Governance*, *3*(2), 95–107. https://doi.org/10.1093/ppmgov/gvz033

Feldman, M. S., Worline, M., Baker, N., & Lowerson Bredow, V. (2022). Continuity as patterning: A process perspective on continuity. *Strategic Organization*, *20*(1), 80–109. https://doi.org/10.1177/14761270211046878

Feuls, M., Hernes, T., & Schultz, M. (2025). Putting distant futures into action: How actors sustain a course of action toward distant-future goals through path enactment. *Academy of Management Journal*, *68*(2), 297–325. https://doi.org/10.5465/amj.2022.0257

Geiger, D., Danner-Schröder, A., & Kremser, W. (2021). Getting ahead of time—performing temporal boundaries to coordinate routines under temporal uncertainty. *Administrative Science Quarterly*, *66*(1), 220–264. https://doi.org/10.1177/0001839220941010

Goh, K., & Pentland, B. T. (2019). From actions to paths to patterning: Toward a dynamic theory of patterning in routines. *Academy of Management Journal*, *62*(6), 1901–1929. https://doi.org/10.5465/amj.2018.0042

Hansson, M., Hærem, T., & Pentland, B. T. (2023). The effect of repertoire, routinization and enacted complexity: Explaining task performance through patterns of action. *Organization Studies*, *44*(3), 473–496. https://doi.org/10.1177/01708406211069438

Howard-Grenville, J. (2005). The persistence of flexible organizational routines: The role of agency and organizational context. *Organization Science*, *16*(6), 618–636. https://doi.org/10.1287/orsc.1050.0150

Ingold, T. (2011). *Being alive: Essays on movement, knowledge and description*. Routledge.

Jarzabkowski, P., Lê, J., & Spee, P. (2017). Taking a strong process approach to analyzing qualitative process data. In A. Langley & H. Tsoukas (Eds.), *The sage handbook of process organization studies* (pp. 237–253). Sage. https://doi.org/10.4135/9781473957954.n15

Jarzabkowski, P., Lê, J. K., Seidl, D., & Vaara, E. (2025). Two decades of revolutionizing strategy research: How strategy-as-practice illuminates the complex, dynamic and consequential ways of doing strategy. *Academy of Management Collections*, *4*(2), 55–76. https://doi.org/10.5465/amc.2023.0005

Karnøe, P., & Garud, R. (2012). Path creation: Co-creation of heterogeneous resources in the emergence of the Danish wind turbine cluster. *European Planning Studies*, *20*(5), 733–752. https://doi.org/10.1080/09654313.2012.667923

Kremser, W., & Blagoev, B. (2021). The dynamics of prioritizing: How actors temporally pattern complex role–routine ecologies. *Administrative Science Quarterly*, *66*(2), 339–379. https://doi.org/10.1177/0001839220948483

LeBaron, C., Christianson, M. K., Garrett, L., & Ilan, R. (2016). Coordinating flexible performance during everyday work: An ethnomethodological study of handoff routines. *Organization Science*, *27*(3), 514–534. https://doi.org/10.1287/orsc.2015.1043

Locke, K., Feldman, M. S., & Golden-Biddle, K. (2015). Discovery, validation, and live coding. In K. D. Elsbach & R. Kramer (Eds.), *Handbook of qualitative organizational research* (pp. 371–379). Routledge.

Mahringer, C. A., & Danner-Schröder, A. (in press). Autonomous, yet interdependent: Designing interfaces across routine clusters. *Academy of Management Journal*. https://doi.org/10.5465/amj.2022.0853

Mahringer, C. A., & Pentland, B. T. (2021). Sequence analysis in routine dynamics. In M. S. Feldman, B. T. Pentland, L. D'Adderio, K. Dittrich, C. Rerup, & D. Seidl (Eds.), *Cambridge handbook of routine dynamics* (pp. 172–183). Cambridge University Press. https://doi.org/10.1017/9781108993340.015

Mahringer, C. A., Pentland, B. T., Renzl, B., Sele, K., & Spee, P. (2024). Routine dynamics: Organizing in a world in flux. *Research in the Sociology of Organizations*, *88*, 1–15. https://doi.org/10.1108/S0733-558X20240000088001

Mengis, J., Nicolini, D., & Swan, J. (2018). Integrating knowledge in the face of epistemic uncertainty: Dialogically drawing distinctions. *Management Learning*, *49*(5), 595–612. https://doi.org/10.1177/1350507618797216

Pentland, B. T., Mahringer, C. A., Dittrich, K., Feldman, M. S., & Ryan Wolf, J. (2020). Process multiplicity and process dynamics: Weaving the space of possible paths. *Organization Theory*, *1*(3), 1–21. https://doi.org/10.1177/2631787720963138

Pentland, B. T., & Rueter, H. H. (1994). Organizational routines as grammars of action. *Administrative Science Quarterly*, *39*(3), 484–510. https://doi.org/10.2307/2393300

Ritter, F., Danner-Schröder, A., & Müller-Seitz, G. (2024). Agile routines enabling efficiency and flexibility: Demarcating and integrating temporal orientations. *Research in the Sociology of Organizations*, *88*, 151–177. https://doi.org/10.1108/S0733-558X20240000088008

Sailer, P., Loscher, G. J., & Kaiser, S. (2023). Coordinated interdependence: How patterning governs flexibility in a routine cluster. *Journal of Management Studies*, *61*(5), 1884–1915. https://doi.org/10.1111/joms.12960

Salvato, C., & Rerup, C. (2018). Routine regulation: Balancing conflicting goals in organizational routines. *Administrative Science Quarterly*, *63*(1), 170–209. https://doi.org/10.1177/0001839217707738

Schwaber, K., & Sutherland, J. (2020). *The Scrum guide. The definitive guide to Scrum: The rules of the game*. https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Seidl, D., Grossmann-Hensel, B., & Jarzabkowski, P. (2021). Strategy as practice and routine dynamics. In M. S. Feldman, B. T. Pentland, L. D'Adderio, K. Dittrich, C. Rerup, & D. Seidl (Eds.), *Cambridge handbook of routine dynamics* (pp. 481–500). Cambridge University Press. https://doi.org/10.1017/9781108993340.040

Sele, K., Danner-Schröder, A., & Mahringer, C. A. (2025). Embodied connection work: The role of the lived body in routine recreation in extreme contexts. *Journal of Management Studies*, *62*(3), 1300–1329. https://doi.org/10.1111/joms.13113

Shotter, J. (2008). Dialogism and polyphony in organizing theorizing in organization studies: Action guiding anticipations and the continuous creation of novelty. *Organization Studies*, *29*(4), 501–524. https://doi.org/10.1177/0170840608088701

Skade, L. (2025). The future looks bright: Toward a dialectical understanding of the future in the strategy-making process. *Organization Theory*, *6*(3), 1–24. https://doi.org/10.1177/26317877251377779

Sminia, H., & Corvalán, J. (in press). Strategic management as wayfinding: The doing of strategy work. In A. Zuback & S. S. Kirkpatrick (Eds.), *The Palgrave handbook of strategy, change and transformational project leadership*. Palgrave Macmillan.

Tsoukas, H. (2017). Don't simplify, complexify: From disjunctive to conjunctive theorizing in organization and management studies. *Journal of Management Studies*, *54*(2), 132–153. https://doi.org/10.1111/joms.12219

Turner, S. F., & Rindova, V. (2012). A balancing act: How organizations pursue consistency in routine functioning in the face of ongoing change. *Organization Science*, *23*(1), 24–46. https://doi.org/10.1287/orsc.1110.0653

Turner, S. F., & Rindova, V. P. (2018). Watching the clock: Action timing, patterning, and routine performance. *Academy of Management Journal*, *61*(4), 1253–1280. https://doi.org/10.5465/amj.2015.0947

Turner, S. F., & Rindova, V. P. (2021). Time, temporality and history in routine dynamics. In M. S. Feldman, B. T. Pentland, L. D'Adderio, K. Dittrich, C. Rerup, & D. Seidl (Eds.), *Cambridge handbook of routine dynamics* (pp. 266–276). Cambridge University Press. https://doi.org/10.1017/9781108993340.023

Wegener, F. E., & Lorino, P. (2020). Capturing the experience of living forward from within the flow: Fusing "withness" approach and Pragmatist inquiry. In J. Reinecke, R. Suddaby, A. Langley, & H. Tsoukas (Eds.), *Time, temporality, and history in process organization studies* (pp. 138–168). Oxford University Press. https://doi.org/10.1093/oso/9780198870715.003.0009

Xu, S. M., & Carlile, P. R. (2024). Agency, action, and time: A relational approach to routine dynamics in a world in flux. In *Routine dynamics: Organizing in a world in flux* (Vol. 88, pp. 245-269). Emerald Publishing Limited. https://doi.org/10.1108/S0733-558X202488

Author biographies

Christian A. Mahringer is a postdoctoral researcher at the University of Stuttgart, Germany and a project leader at the WIN-Kolleg of the Heidelberg Academy of Sciences and Humanities. His research adopts a practice-theoretical perspective to examine how organizational actors coordinate and navigate strategic change in response to emerging technologies and grand challenges. His work has been published in academic journals, including the Academy of Management Journal, Journal of Management Studies, Organization Theory, and Strategic Organization.

Katharina Dittrich is Associate Professor of Organisation Studies in the Organisation and Work Group at Warwick Business School, UK. She received her PhD in Organisation Theory from the University of Zurich, Switzerland and has been a visiting research fellow at the Said Business School in Oxford, UK, and the University of California at Irvine, USA. Her research

interests include climate change, organizational change, organizational routines/routine dynamics and strategy, with an emphasis on practice-theoretical approaches and qualitative research methods. Her worked has been published in Organization Science, Academy of Management Journal, and Organization Theory. She is member of the editorial board of the Journal of Management Studies and the Journal of Organizational Ethnography and a Fellow of the Higher Education Academy.

Birgit Renzl is the Chair of Management and Organization at the University of Stuttgart, Germany. She received her doctoral degree from the University of Innsbruck, Austria and has been visiting research fellow at the University of Strathclyde in Glasgow, Scotland/UK, the University of St. Gallen, Switzerland, and the KIN Center for Digital Innovation at the VU Amsterdam, the Netherlands. Her current research focuses on organizing and strategic change processes, with particular focus on emerging technologies and organizational routines. She is committed to the scientific community and equally dedicated to fostering the dialogue between research and practice. Her work has been published in Management Learning, Omega, The International Journal of Human Resource Management, Journal of Economic Psychology, and Strategic Organization.