

# **Liberating Pair Programming Research from the Oppressive Driver/Observer Regime**

Lutz Prechelt  
Freie Universität Berlin, Institut für Informatik

- Stephan Salinger, Franz Zieris, Lutz Prechelt:  
*"Liberating Pair Programming Research from the Oppressive Driver/Observer Regime"*,  
International Conference on Software Engineering,  
New Ideas and Emerging Results Track (NIER),  
May 2013, San Francisco, to appear.

## Definition: Pair Programming (PP)

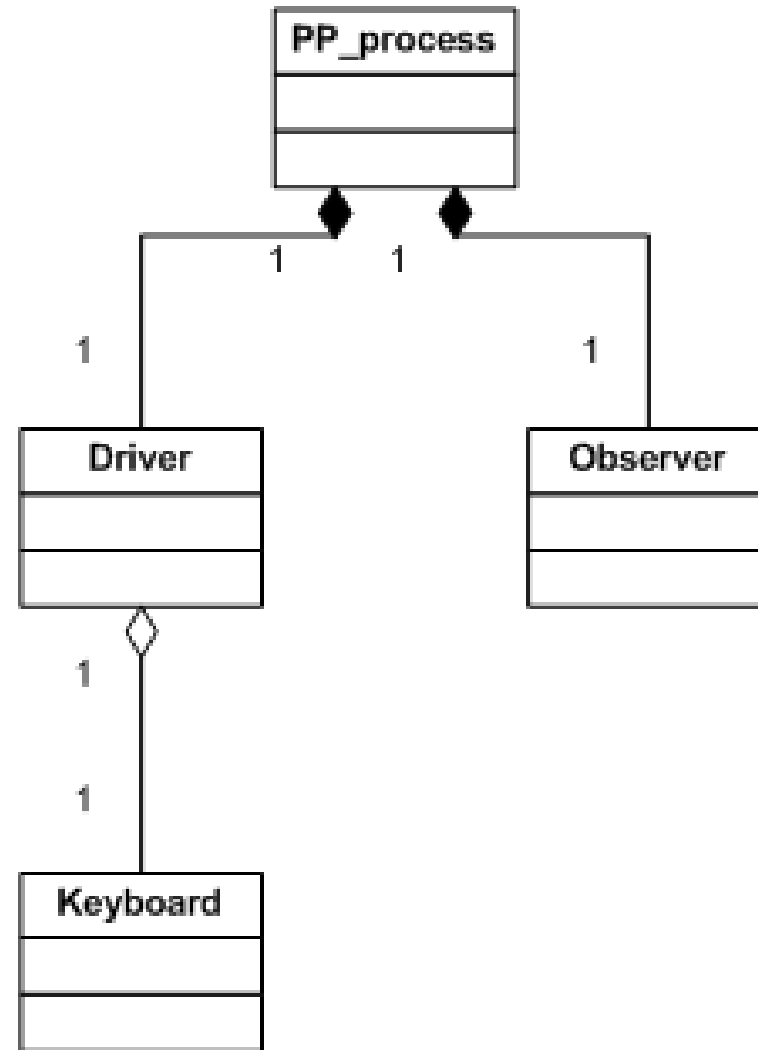
From [WilKesCun00]

"In pair-programming,

- two programmers jointly produce one artifact (design, algorithm, code, etc.).
- The two programmers are like a coherent, intelligent organism **working with one mind**, responsible for every aspect of this artifact.
- One partner is the '**driver**' and has control of the pencil/mouse/keyboard and is writing the design or code.
- **The other person continuously and actively observes** the work of the driver -- watching for defects, thinking of alternatives, looking up resources, and considering strategic implications of the work at hand.
- **The roles of driver and observer** are deliberately switched between the pair periodically.
- Both are equal, active participants in the process at all times and wholly share the ownership of the work products whether they be a morning's effort or an entire project."

# Roles model of this definition

- Driver + Observer constitute the PP process
- Driver is recognized simply from "having the keyboard"



- There is a contradiction between "working as one mind" and the different responsibilities ascribed to the roles
- Prior research found that the partners are not typically working at *different* levels of abstraction
  - Sallyann Bryant, P. Romero, B. du Boulay:  
*"Pair Programming and the Mysterious Role of the Navigator"*.  
International Journal of Human-Computer Studies (2008)
- However, an alternative roles model has not yet been proposed

## Why is this of interest?

- PP advocates claim many advantages from using PP:
  - High productivity
  - Better resulting design
  - Fewer defects
  - Strong mutual learning
  - etc.
- But empirical results are either lacking or show wide variation
- We do not know how to do PP 'right'
- We do not even know what it means to do PP at all
- Understanding PP roles may help with both

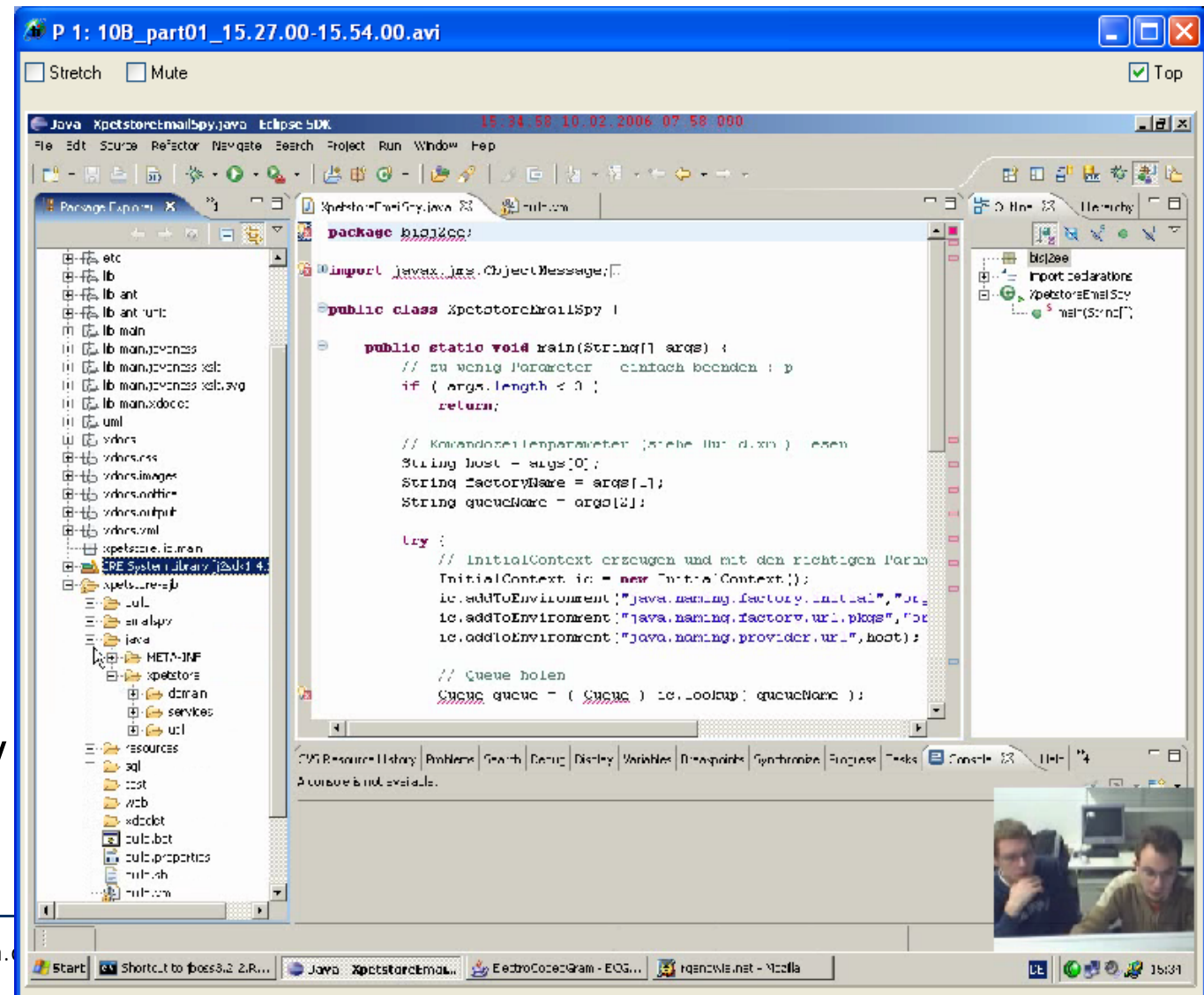
# AG SE research approach

## Goal:

- Characterize the PP process conceptually
  - important mechanisms, typical event episodes, styles, ...
  - Describe helpful behaviors as process patterns
  - Describe hampering behaviors as process anti-patterns
- Approach:
  - Bottom-up empirical analysis:
    - Use audio/video recordings of PP sessions
    - Conceptualize locally
    - Find structure
  - Not a short single study, but a long-term research program
  - Main research method: Grounded Theory Methodology

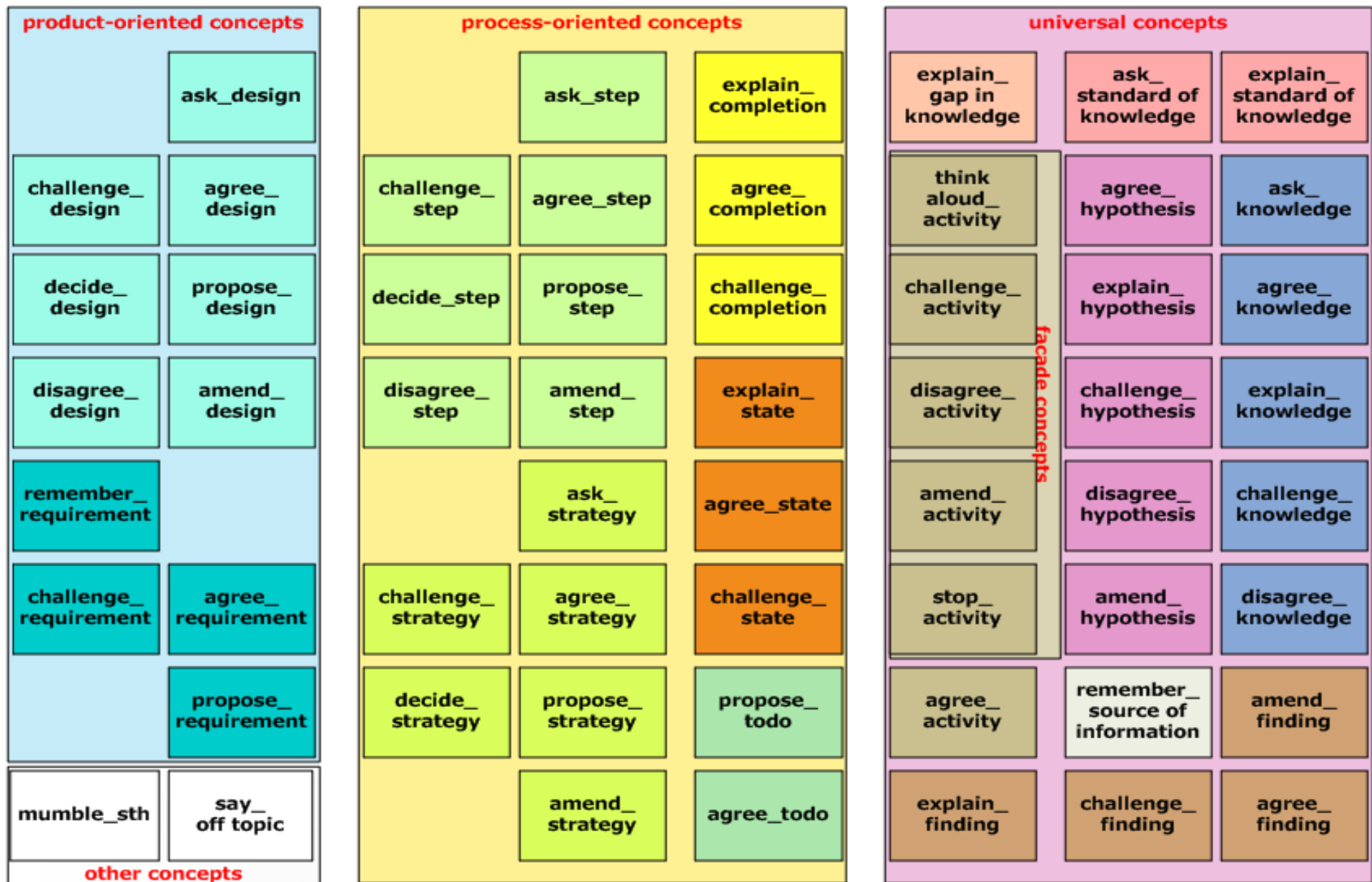
# Initial insights

- Such data is *shockingly* rich
- Should focus on verbal interaction
- Many many utterances appear ambiguous
- Many many actions are ambiguous/multilayered
- 1st goal: Must get clarity for basic dialog steps



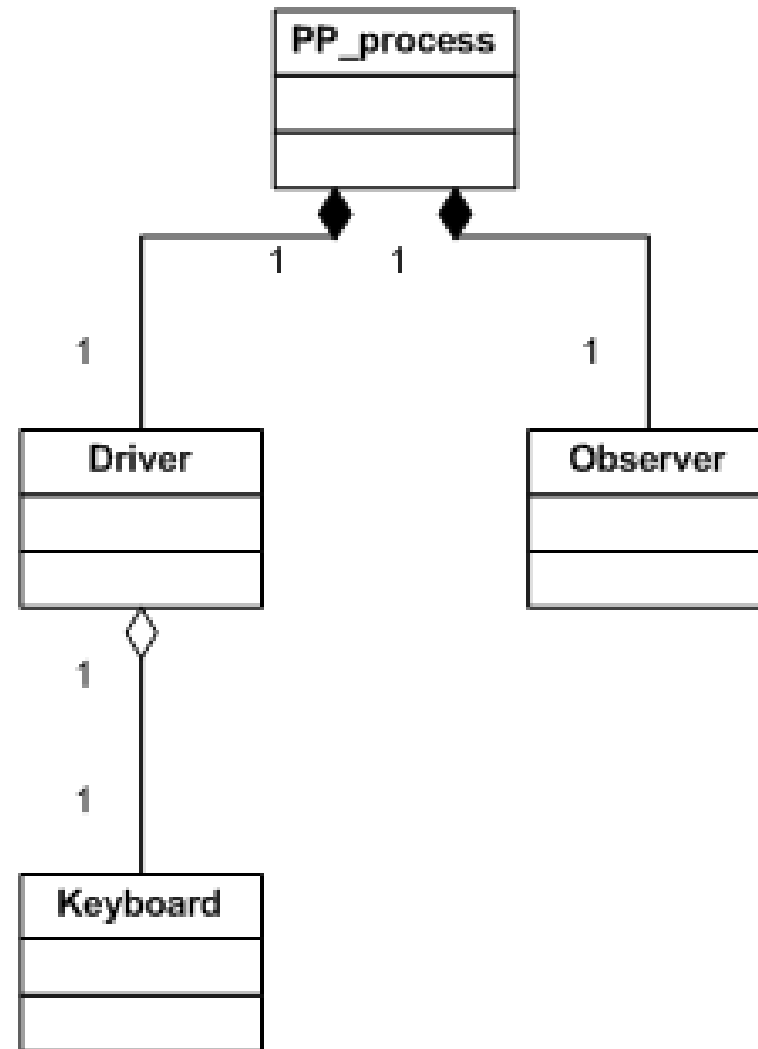


# Base Concept Set: HHI concepts

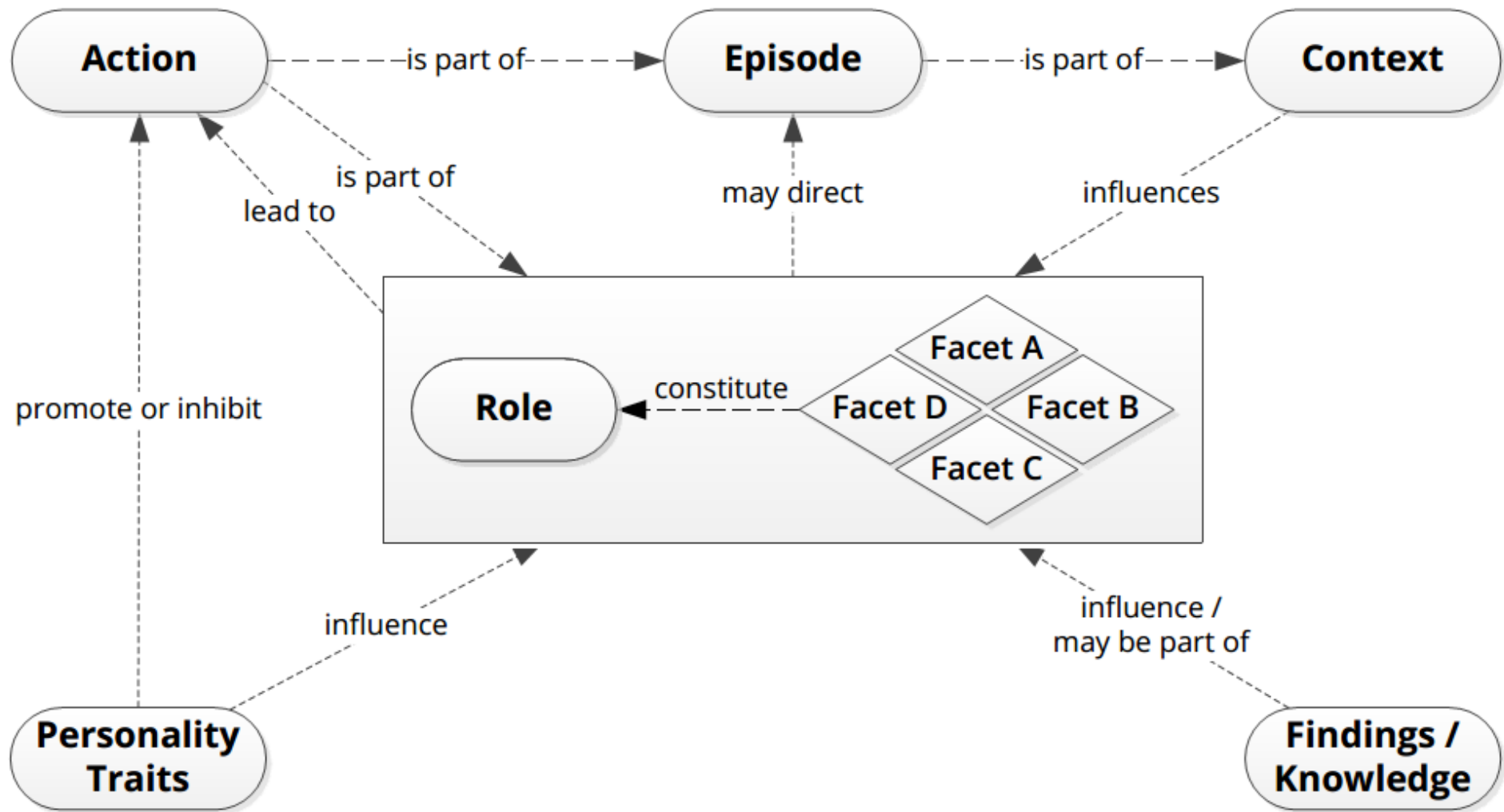


# Again: Roles model of the traditional PP definition

- Driver + Observer constitute the PP process
- Driver is recognized simply from "having the keyboard"



# New, open-ended roles meta-model



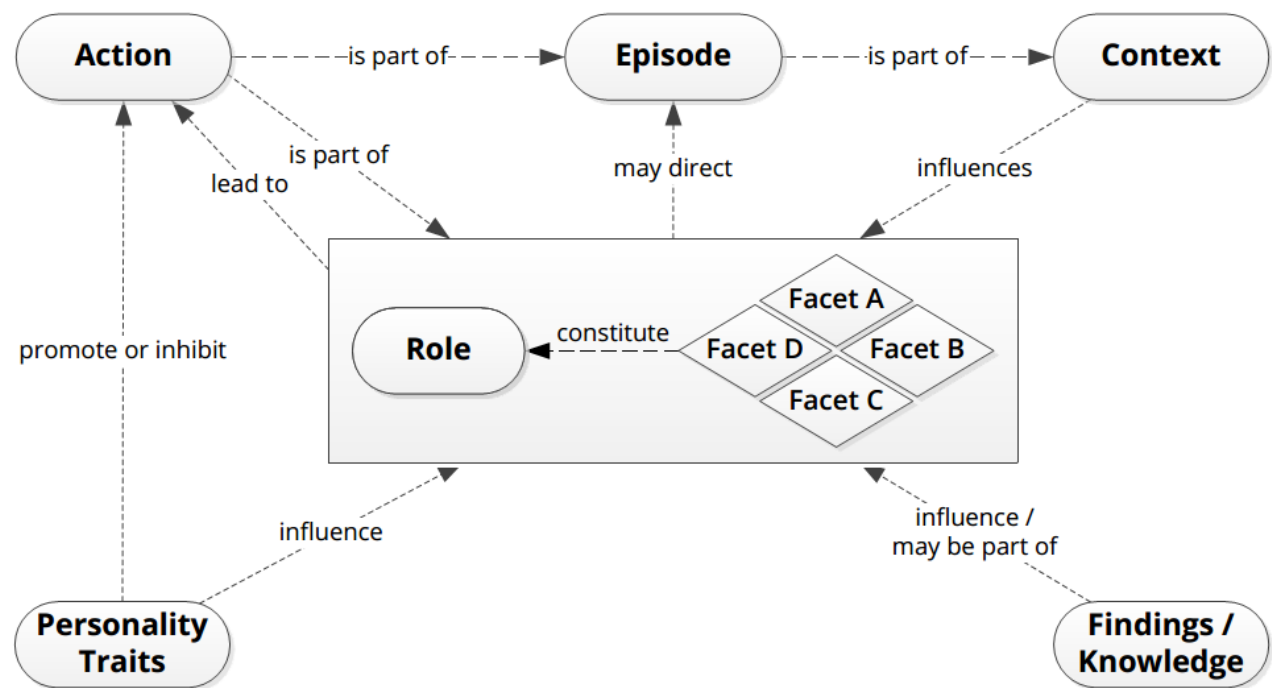
# Finding 1: A Role consists of Facets

(We have not identified many roles yet, so we focus on the meta-model and only use roles as examples)

- A **Role** is described via several characteristic **Facets**
- e.g. the role **watchman** has three Facets:
  - **recognizing hazards**: looking for dangers, e.g. when the current set of files is not up-to-date or when not finishing the task could hamper the release process
  - **setting priorities**: Insisting that something be done now or early enough to avoid negative or ensure positive consequences.
  - **shifting**: Triggering a switch from one context of consideration (e.g. the PP session) into another (e.g. the overall software process)

# Recognizing role-fillings from Facets and Actions

- The Facets as such are missions/intentions/responsibilities and are thus not directly observable, but **Actions** resulting from them are
  - e.g. looking for hazards is unobservable but warning of a hazard is observable
- Sometimes a single Action and Facet is enough to diagnose a role-filling
- Sometimes repeated Actions or multiple Facets are required



## Finding 2: Roles require skill

---

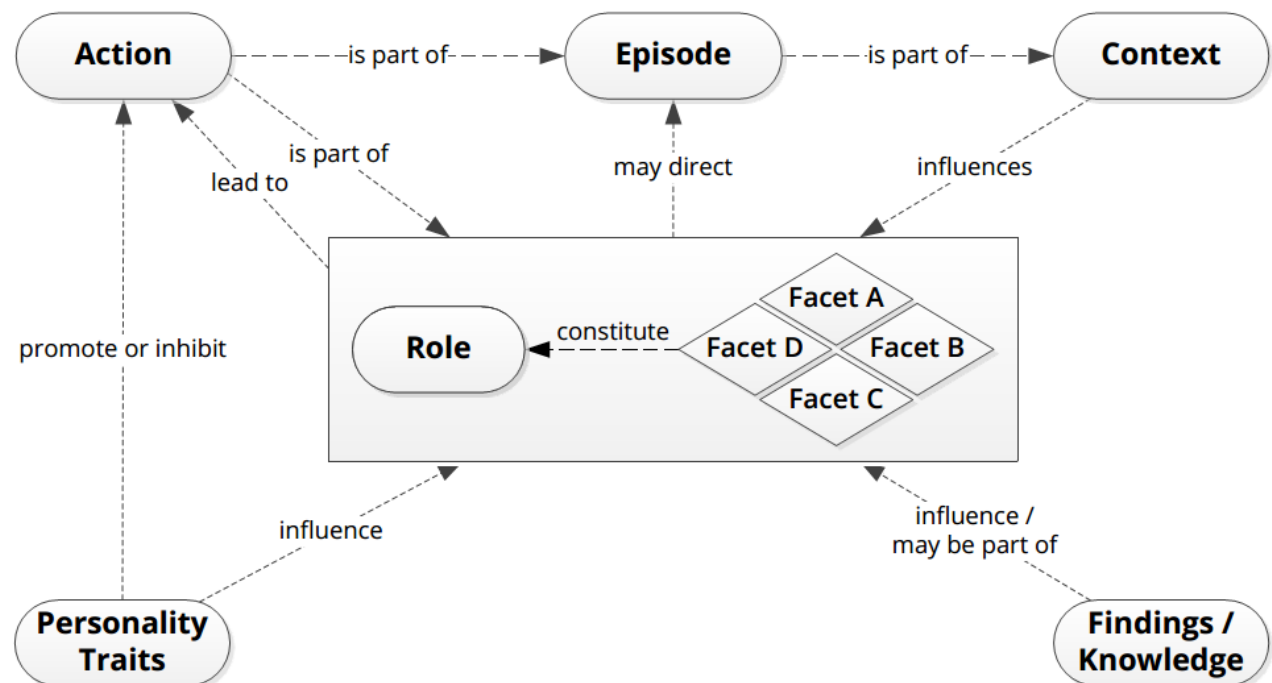
- All three Facets of the watchman Role involve learnable skills
  - searching for hazards
  - setting priorities
  - shifting

Therefore,

- recognizing and understanding the role (in research) may lead to process patterns;
- understanding the role (as a practitioner) may improve the PP behavior

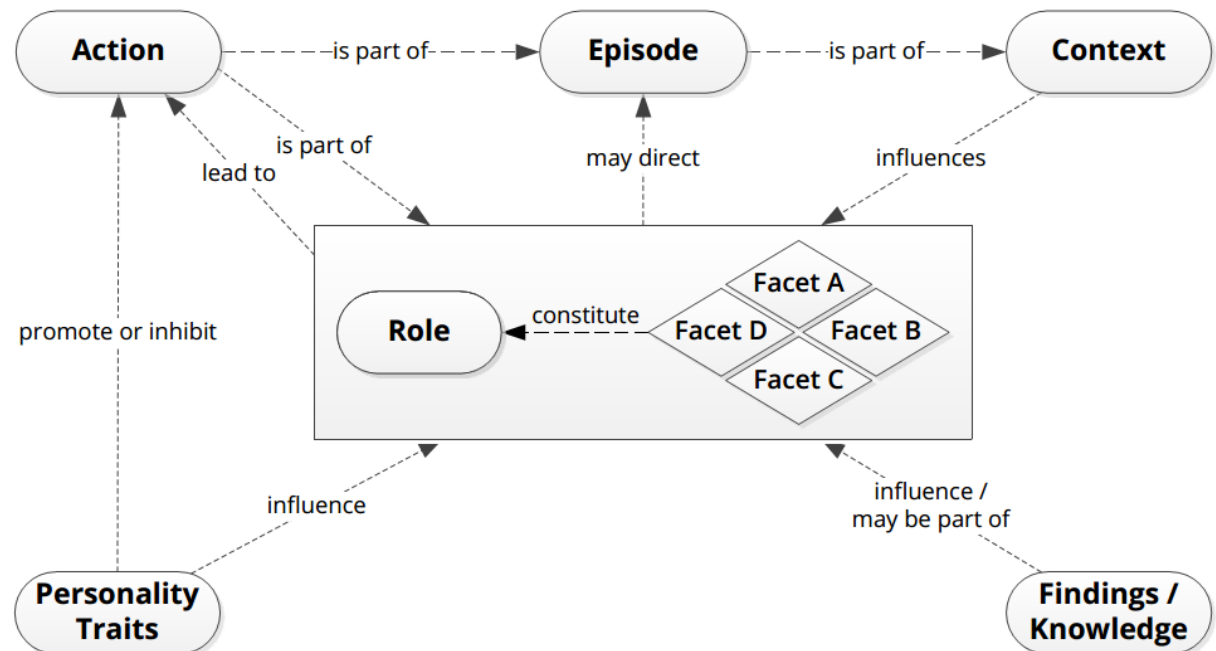
# Finding 3: Role-fillings depend on Context

- Role **task expert** has two facets:
  - pass on task knowledge explicitly
  - turn task knowledge into proposals
- Which of these will appear depends on the current **Context**, e.g.
  - within a knowledge transfer **Episode**, only Facet 1 will appear
  - in a session, where knowledge transfer is not a goal, Facet 1 will hardly ever appear
- (Seeing one Facet is almost always enough to diagnose a task expert)



# Finding 4: Role likelihood may depend on personality traits

- The **spokesperson** Role means being a representative for a certain concern. It has three Facets:
  - **opening a dialog** about the concern.
  - **carrying forward the dialog**, often in such a way that expert knowledge is not required
  - **rounding off the dialog**: Not accepting an end of the dialog without a resolution of the concern.
- The Role is typically assumed by the more assertive member of a pair
  - being assertive is a personality trait, not a Facet





## Finding 5: Some roles have counterparts

---

- If member A is task expert in Facet-1 (knowledge-passing) mode, member B is usually **mentee**
- **mentee** need not occur in pure Facet-2 (proposing) mode
- Some combinations of roles are even mandatory:
  - Member A can be **guide** only if Member B acts as **robot**

## Finding 6: Role-filling can be gradual, intermittent, and concurrent

- A Role assumption may start and end gradually over a stretch of time.
- Having a Role does not mean acting out that Role at every moment.
- A pair member may have more than one Role at the same time.
- The full version of the meta model has additional classes such as **Role-Filling** and **Facet-Execution**

# Finding 7: There may be no canonical Roles catalog

- Before we found *task expert*, we had a Role *updater*
  - much more specialized.
  - We are happy with task expert: updater is gone for good and task expert will likely stay
- However, before we found *watchman*, we had a Role ***foreign minister***
  - who had a subset of the *recognizing hazards* Facet only.
  - But even the more general watchman does not feel final.
- The final solution may be a set of *overlapping* roles
  - from which a pair can select those that fit best to their task, circumstances, and work style.

- The driver/observer roles model is simplistic and unrealistic
  - It is also not constructive (does not provide advice)
- There are far more than two roles in the PP process
- Roles are such complex constructs that their definition is arbitrary to some degree
  - Hence, there may be no fixed true-for-all catalog of roles
- Roles talk about the responsibilities being assumed

**Thank you!**