

Softwaretechnik

WiSe 09/10

Übungsblatt 7

Aufgabe 1

Gegeben sei das Use Case-Modell und das statische Modell für das Wetterstationssystem (vgl. Übung 5 und 6). Es soll ein dynamisches Modell für das System entwickelt werden. Dazu sind folgende Schritte durchzuführen:

1. Modellierung der Interaktionen zwischen den Objekten für die einzelnen Anwendungsfälle.
2. Entwicklung von Zustands- und/oder Aktivitätsdiagrammen.

Sequenzdiagramm

- Zeigt Interaktion zwischen **mehreren** Objekten
- Hebt **zeitliche** Reihenfolge hervor

Kommunikationsdiagramm

- Zeigt Interaktion zwischen **mehreren** Objekten
- Hebt **strukturelle** Beziehungen hervor

Hinweis: Sequenz- und Kommunikationsdiagramme stellen im Wesentlichen dieselben Informationen dar.

Input:

- Use-Case-Beschreibungen (Szenarien)
- Statisches Modell

Vorgehensweise

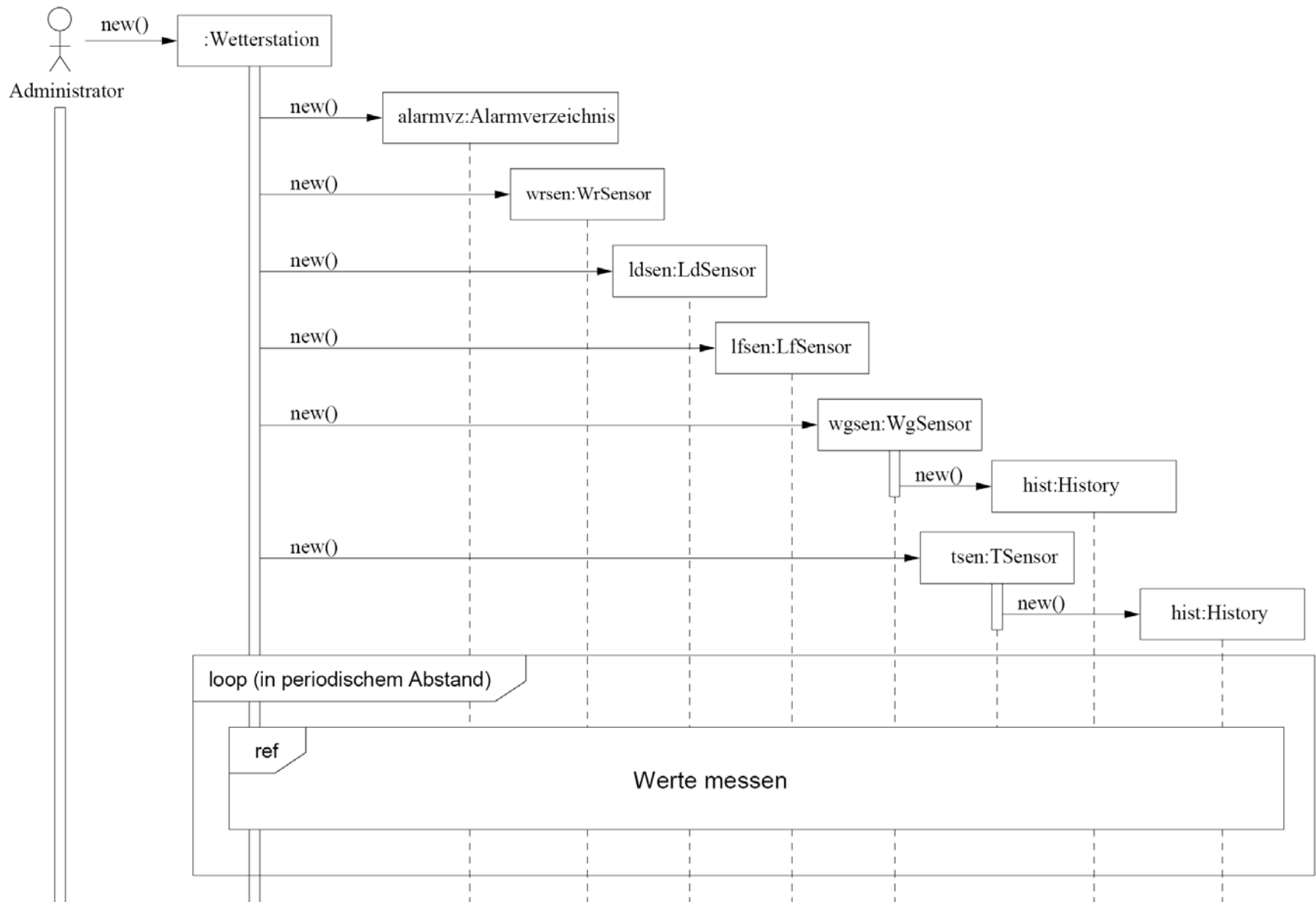
1. Identifizierte Nachrichten, die innerhalb eines Anwendungsfalls ausgetauscht werden und die Objekte, die die Nachrichten senden und empfangen.
2. Konstruiere Interaktionsdiagramme für jeden Anwendungsfall

Hinweis: Pro Anwendungsfall werden i.A. mehrere Interaktionsdiagramme erstellt (für jedes Szenario eines).

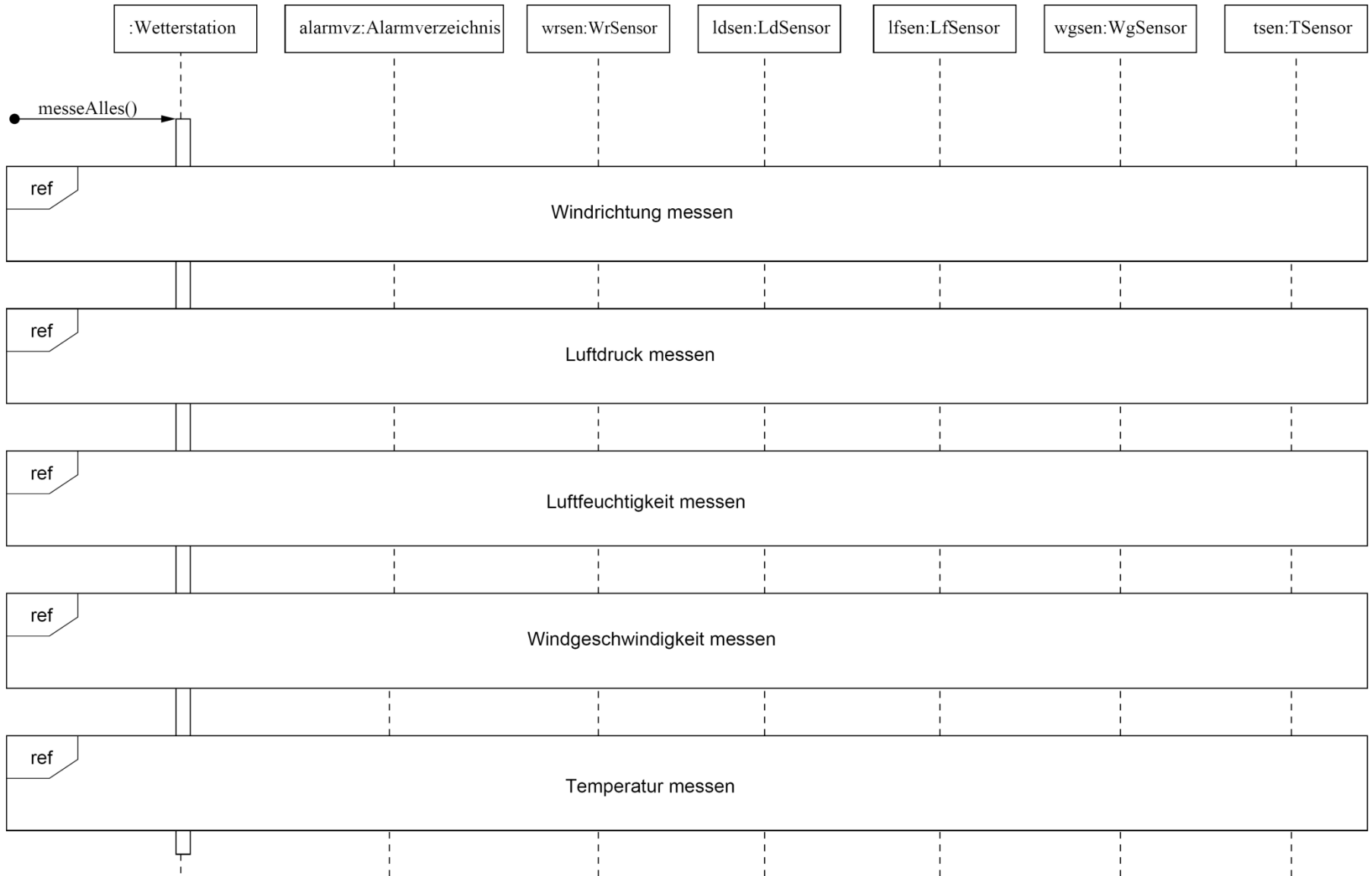
Sequenzdiagramme

- Allgemein
 - Wetterstation starten
 - Werte messen
- Sensoren:
 - Windrichtung messen
 - Luftdruck messen (Luftfeuchtigkeit analog)
 - Windgeschwindigkeit messen (Temperatur analog)
 - **Sturmwarnung bearbeiten** (Glatteiswarnung analog)

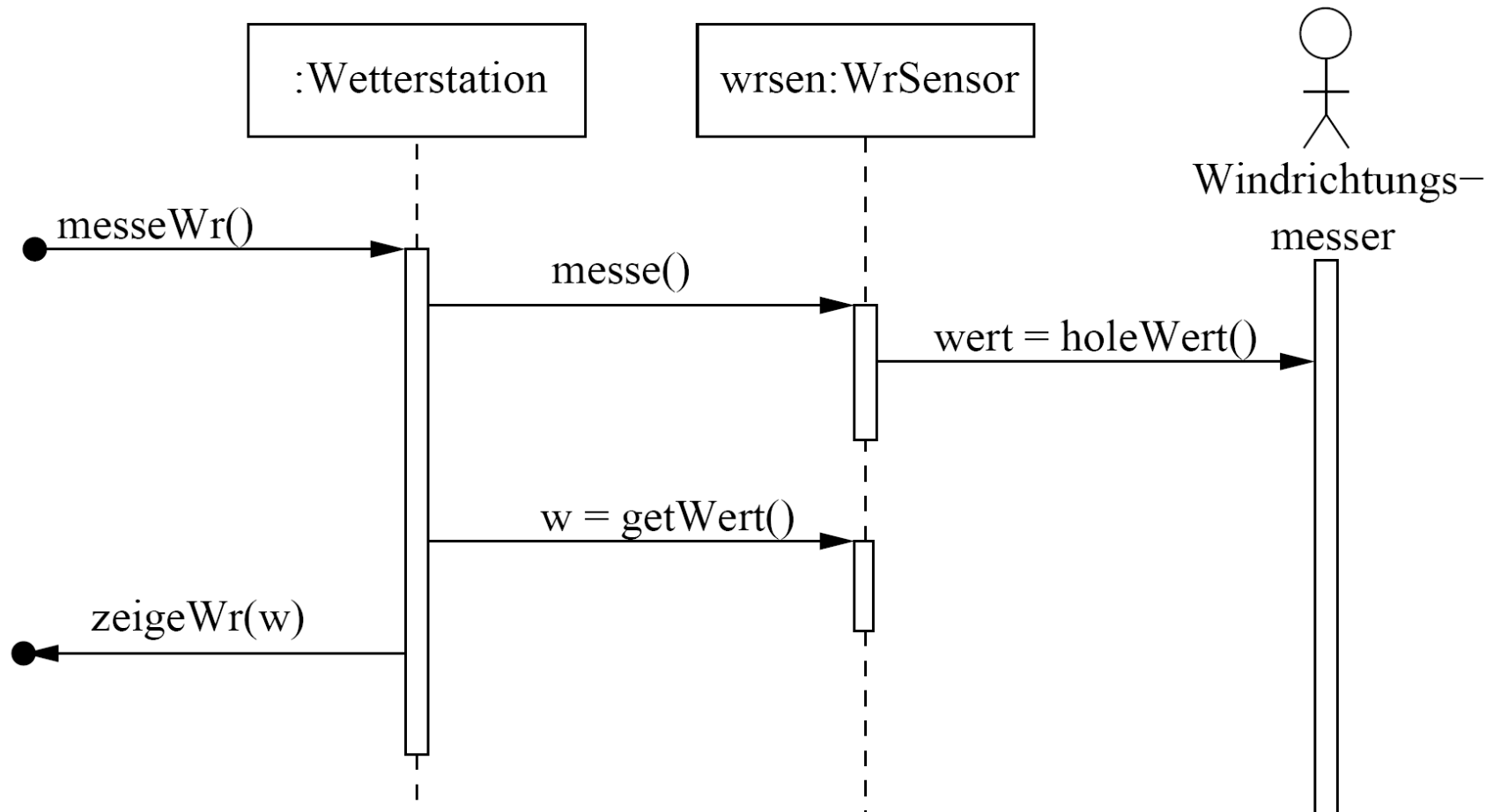
Sequenzdiagramm „Wetterstation starten“



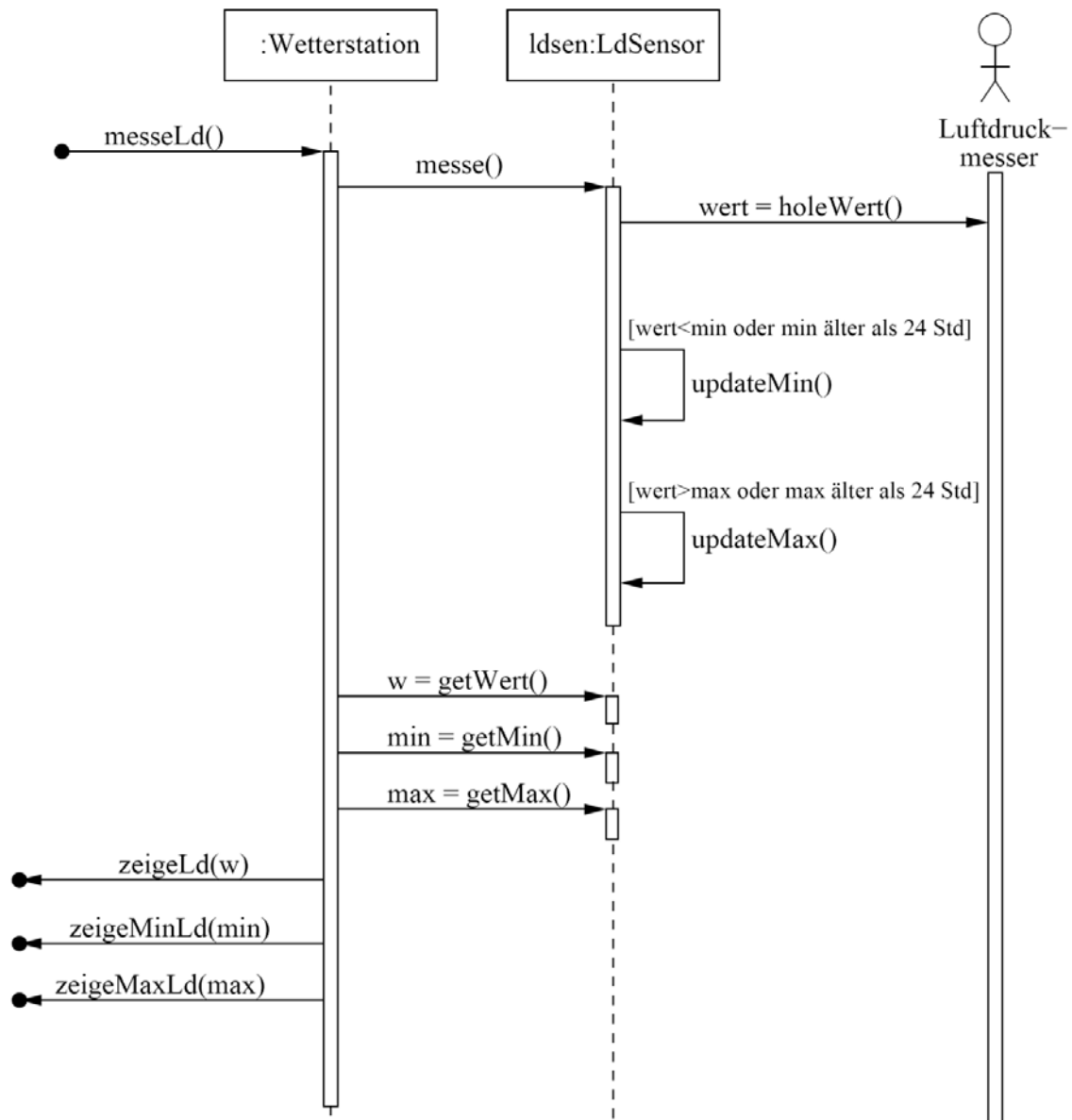
Sequenzdiagramm „Werte messen“



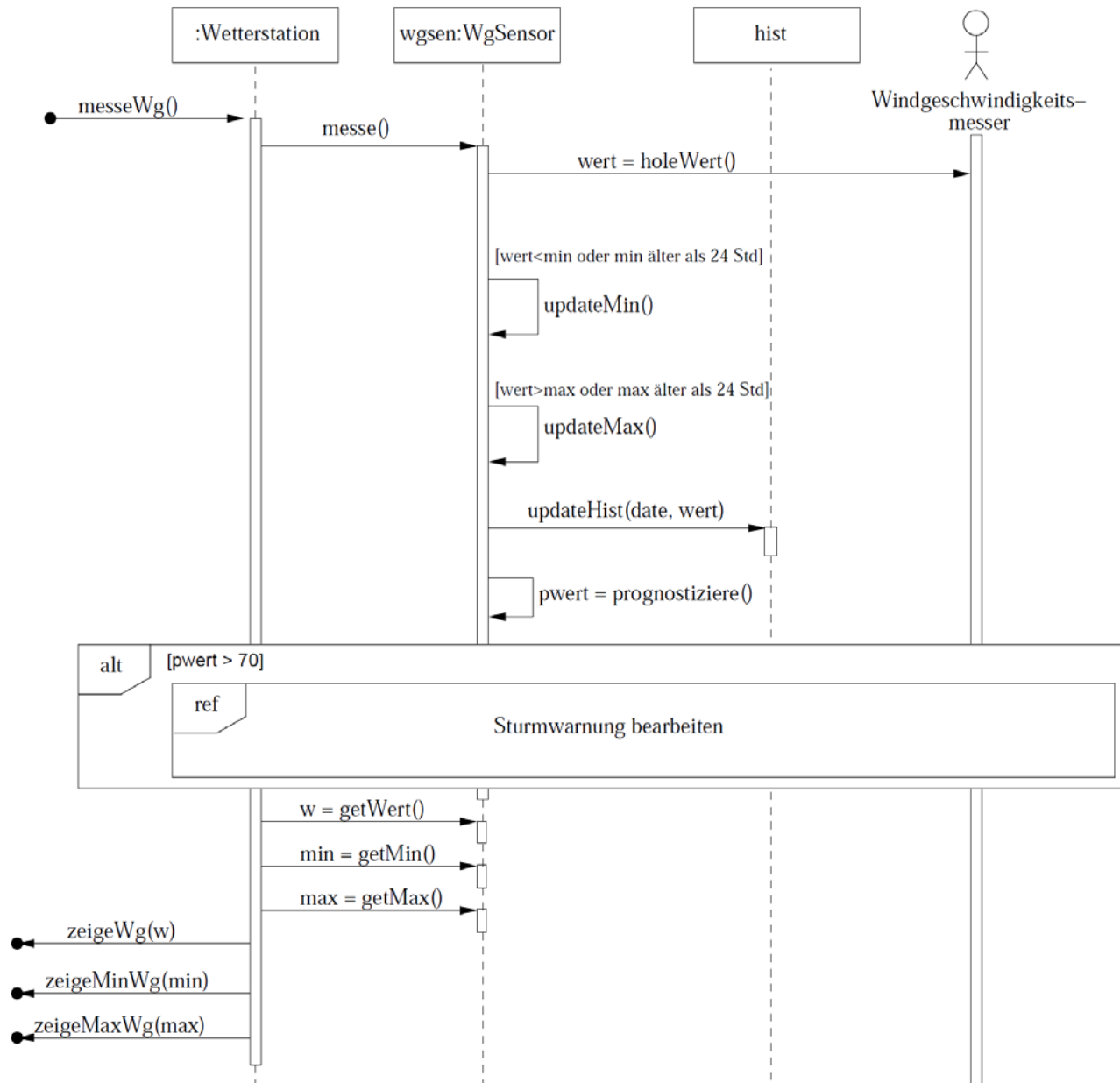
Sequenzdiagramm „Windrichtung messen“



Sequenzdiagramm „Luftdruck messen“

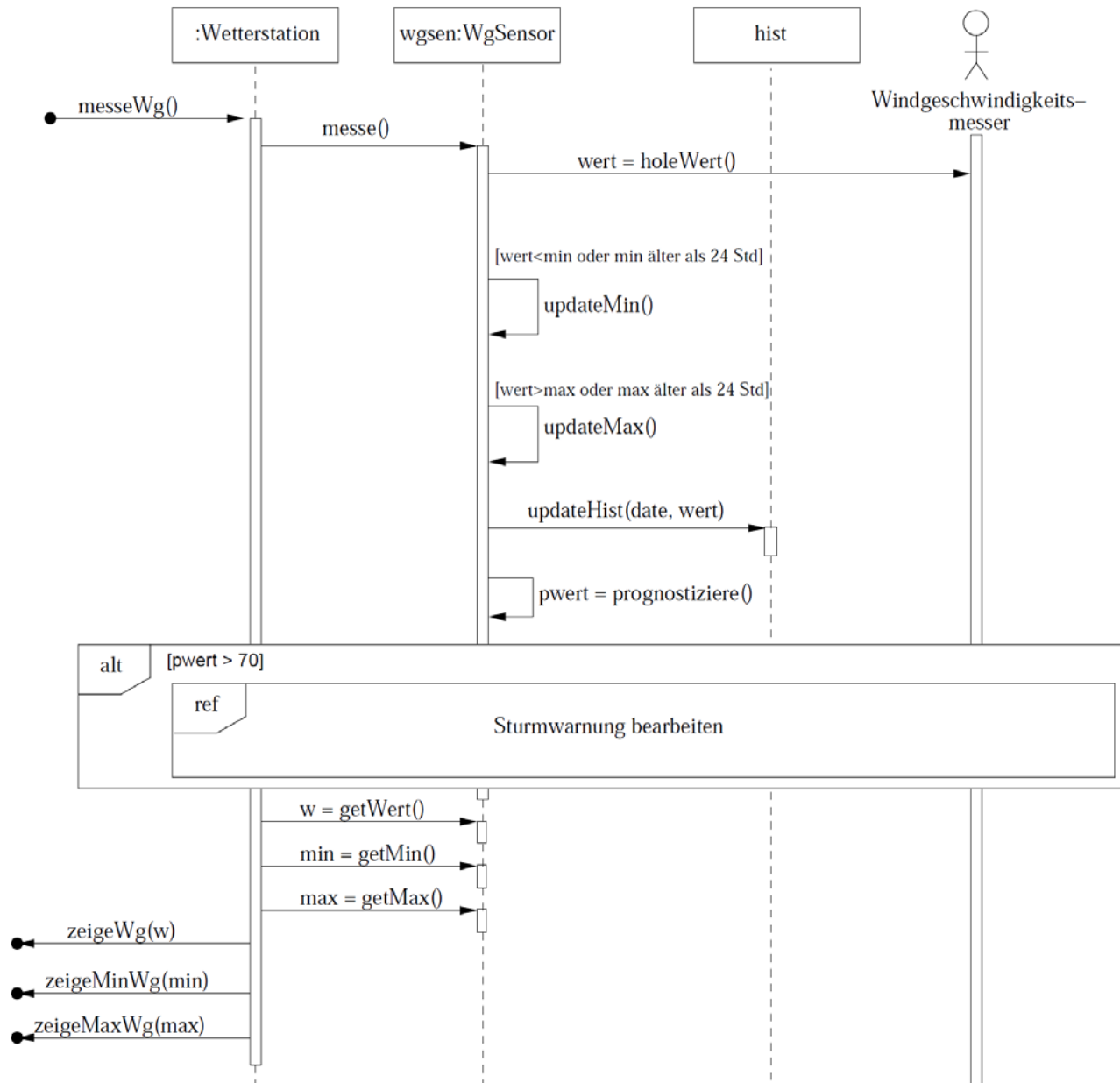


Sequenzdiagramm „Windgeschwindigkeit messen“



- “Sturmwarnung bearbeiten” fehlt noch
- Input
 - [SD von Windgeschwindigkeit messen](#)
 - [Beschreibung des Use Cases](#)
 - [Klassendiagramm](#)
- Hands-On: Erzeugung des Sequenzdiagramms

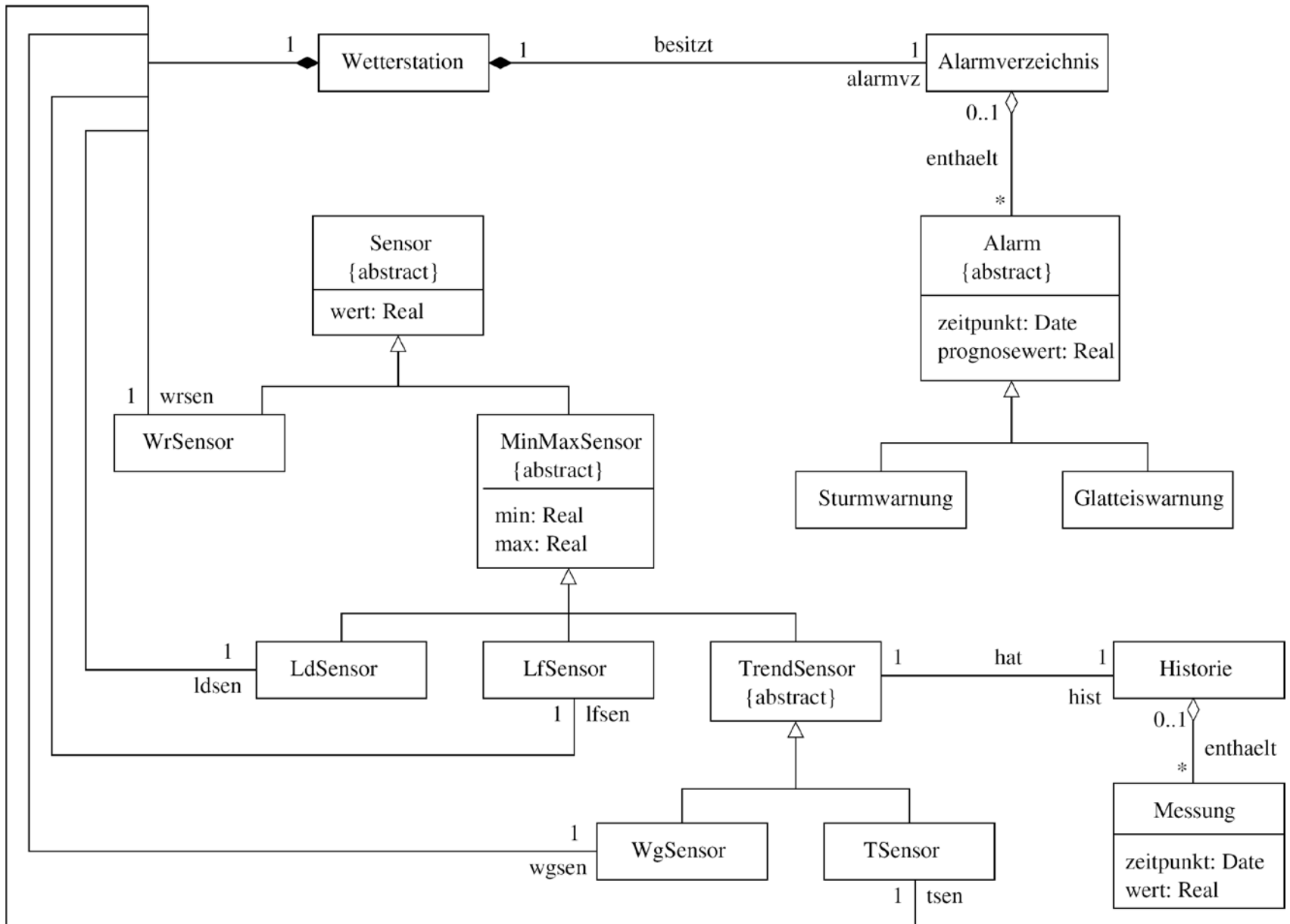
Sequenzdiagramm „Windgeschwindigkeit messen“



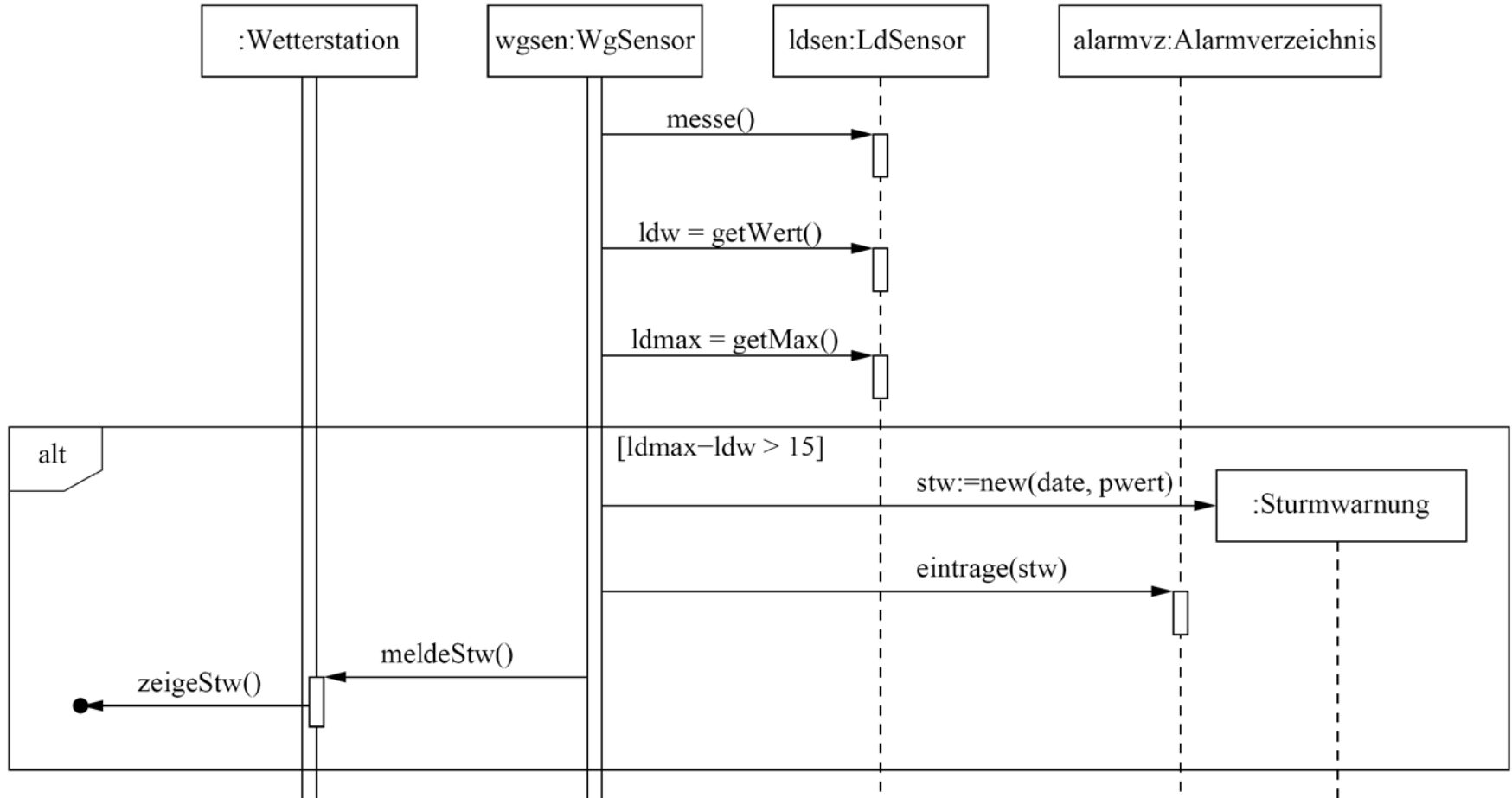
Use Case Sturmwarnung

1. Der Windgeschwindigkeitssensor beauftragt den Luftdrucksensor eine Messung durchzuführen.
2. Der Windgeschwindigkeitssensor holt vom Luftdrucksensor den aktuellen Luftdruck und den Maximalwert der letzten 24 Stunden.
3. Falls der Luftdruck um mehr als 15 Millibar unter den Maximalwert der letzten 24 Stunden gefallen ist, wird eine Sturmwarnung erzeugt.
4. Eine erzeugte Sturmwarnung wird mit den zugehörigen Daten (Zeitpunkt, Prognosewert) in das Alarmverzeichnis eingetragen.
5. Eine erzeugte Sturmwarnung wird vom Windgeschwindigkeitssensor an die Wetterstation gemeldet.
6. Die Wetterstation zeigt eine erzeugte Sturmwarnung an.

Klassendiagramm nach Überarbeitung



Sequenzdiagramm „Sturmwarnung bearbeiten“



Aufgabe 1

Gegeben sei das Use Case-Modell und das statische Modell für das Wetterstationssystem (vgl. Übung 5 und 6). Es soll ein dynamisches Modell für das System entwickelt werden. Dazu sind folgende Schritte durchzuführen:

1. Modellierung der Interaktionen zwischen den Objekten für die einzelnen Anwendungsfälle.
2. Entwicklung von Zustands- und/oder Aktivitätsdiagrammen.

Ausgangspunkt:

- Menge von **Sequenzdiagrammen**

Ziel:

- Je ein **Zustandsdiagramm** für jede *Klasse* mit „interessantem Verhalten“ (Lebenszyklus)
- **Aktivitätsdiagramme** zur Beschreibung der (nichttrivialen) Abläufe von *Operationen*

Hinweis: Ein Zustands- und mehrere Aktivitätsdiagramme erfassen zusammen das vollständige Verhalten eines Objekts einer Klasse über viele Szenarien hinweg.

Zustand:

- Es gibt mind. ein Ereignis, das in Abhängigkeit vom Objektzustand unterschiedliche Reaktionen auslösen kann
- Mind. ein Ereignis wird in bestimmten Zuständen ignoriert.
 - Beispiel: ...?

Aktivität:

- Eine Operation taucht in mehreren Interaktionsdiagrammen auf
 - Erstes Beispiel: Wetterstation: *messeAlles()* ...aber sehr trivial!
 - Anderes Beispiel: WG-Sensor: *messe()* ...bietet etwas mehr!

Algorithmus für Zustandsdiagramme

Für jede Klasse: Bearbeite alle SDs und:

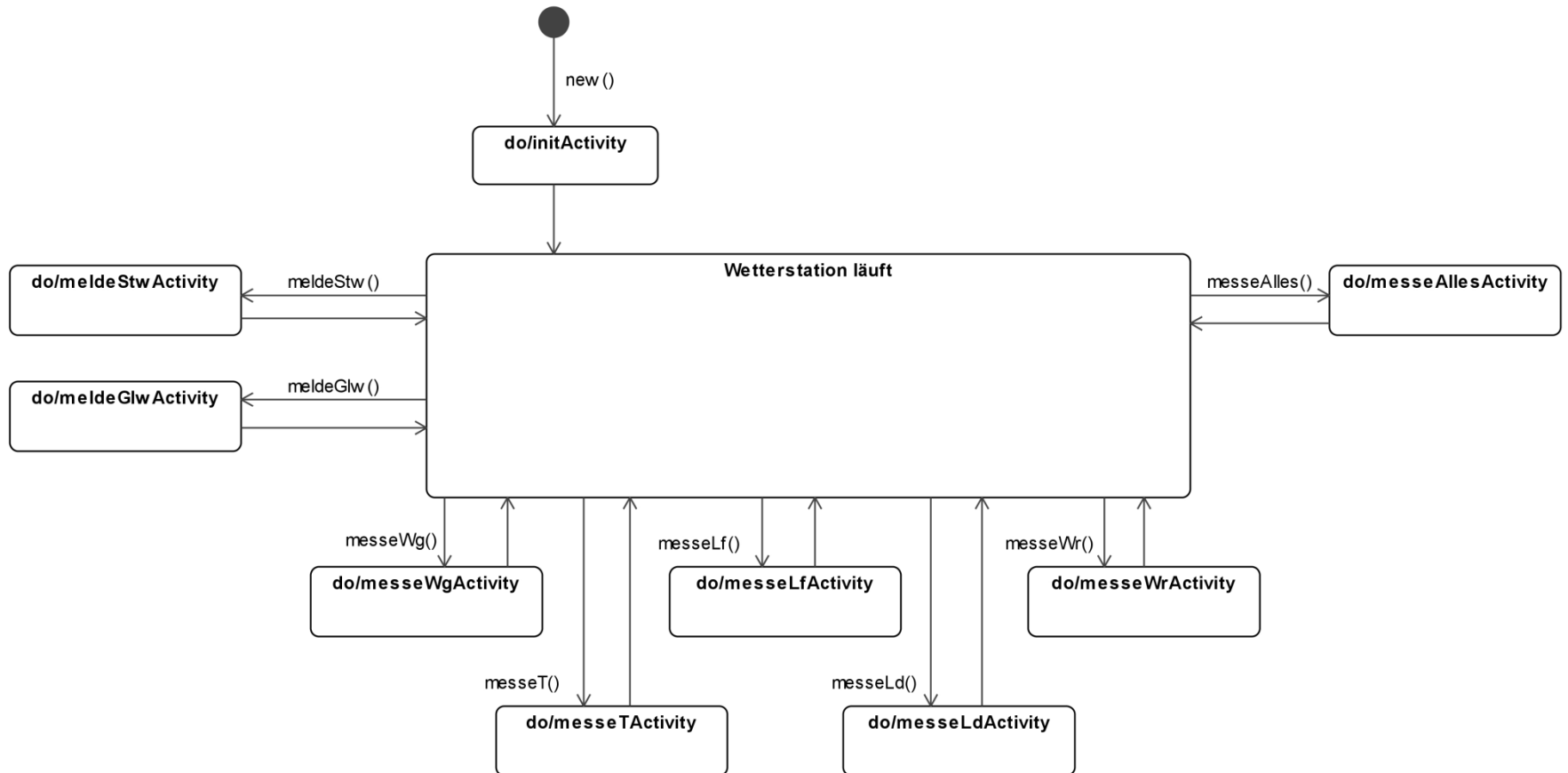
- Bilde der Lebenslinie des Objekts folgend Kette von Zuständen und Transitionen
 1. Nicht-Aktive Phasen => Stabile Zustände
 2. Aktive Phasen => Aktivitätszustände
 3. Eintreffende Ereignisse => Transition von stabilem Zust. zu Aktivitätszustand
 4. Beendigung aktiver Phase => Completion Event von aktivem zu stabilem Zustand
- Bilde Zyklen, falls nötig
- Konstruiere Aktivitätsdiagramme für lokale Operationen

Zuletzt

- Verfeinerung durch Bedingungen / weiteren Informationen

- Klasse **Wetterstation** kommt in jedem Sequenzdiagramm vor
- Konstruktion eines Zustandsdiagramm gemäß Algorithmus der Vorlesung
- **Hands-On:** Erzeugung des Zustandsdiagramms
 - [Siehe Sequenzdiagramme](#)

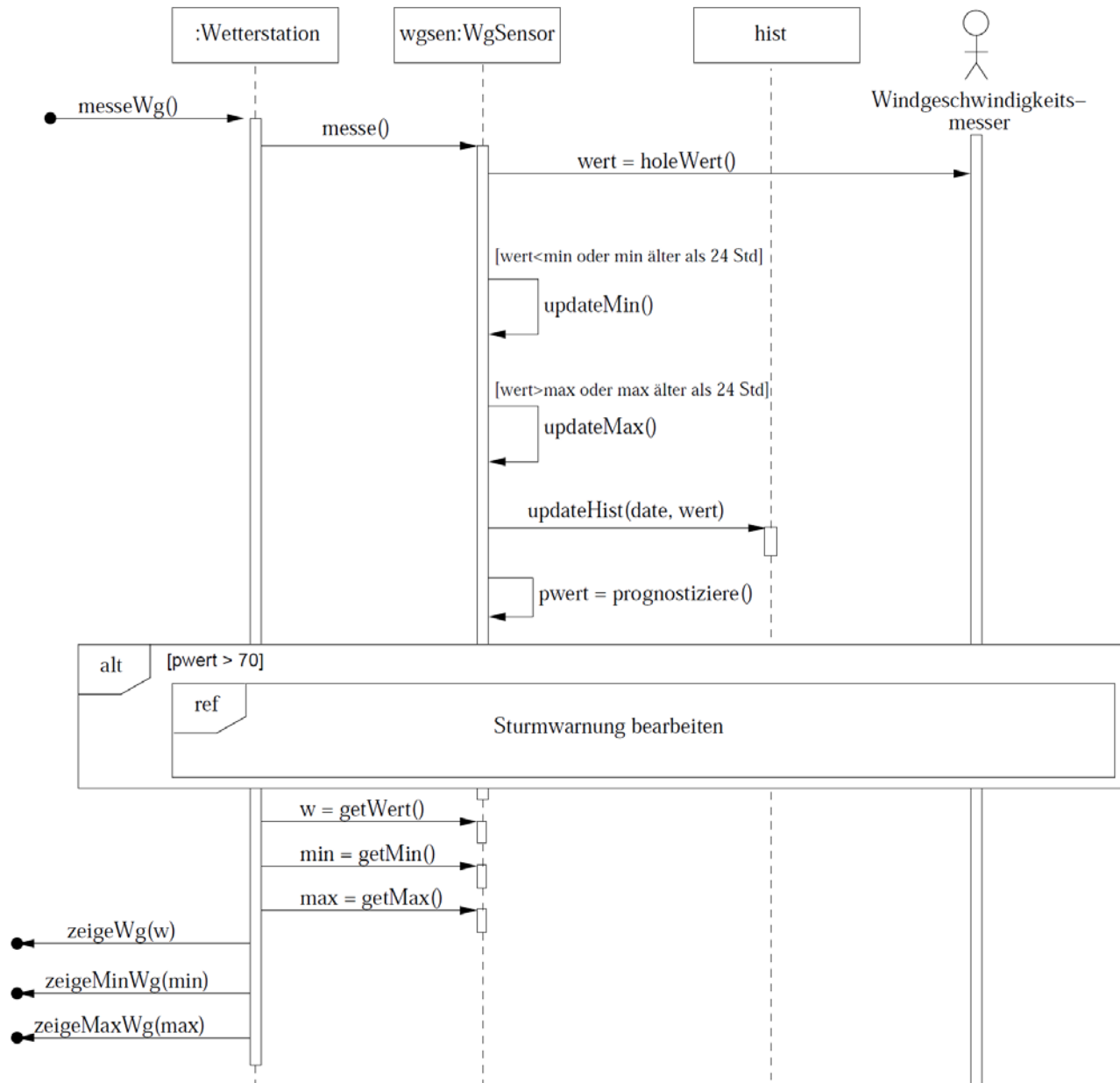
Zustandsdiagramm für „Wetterstation“



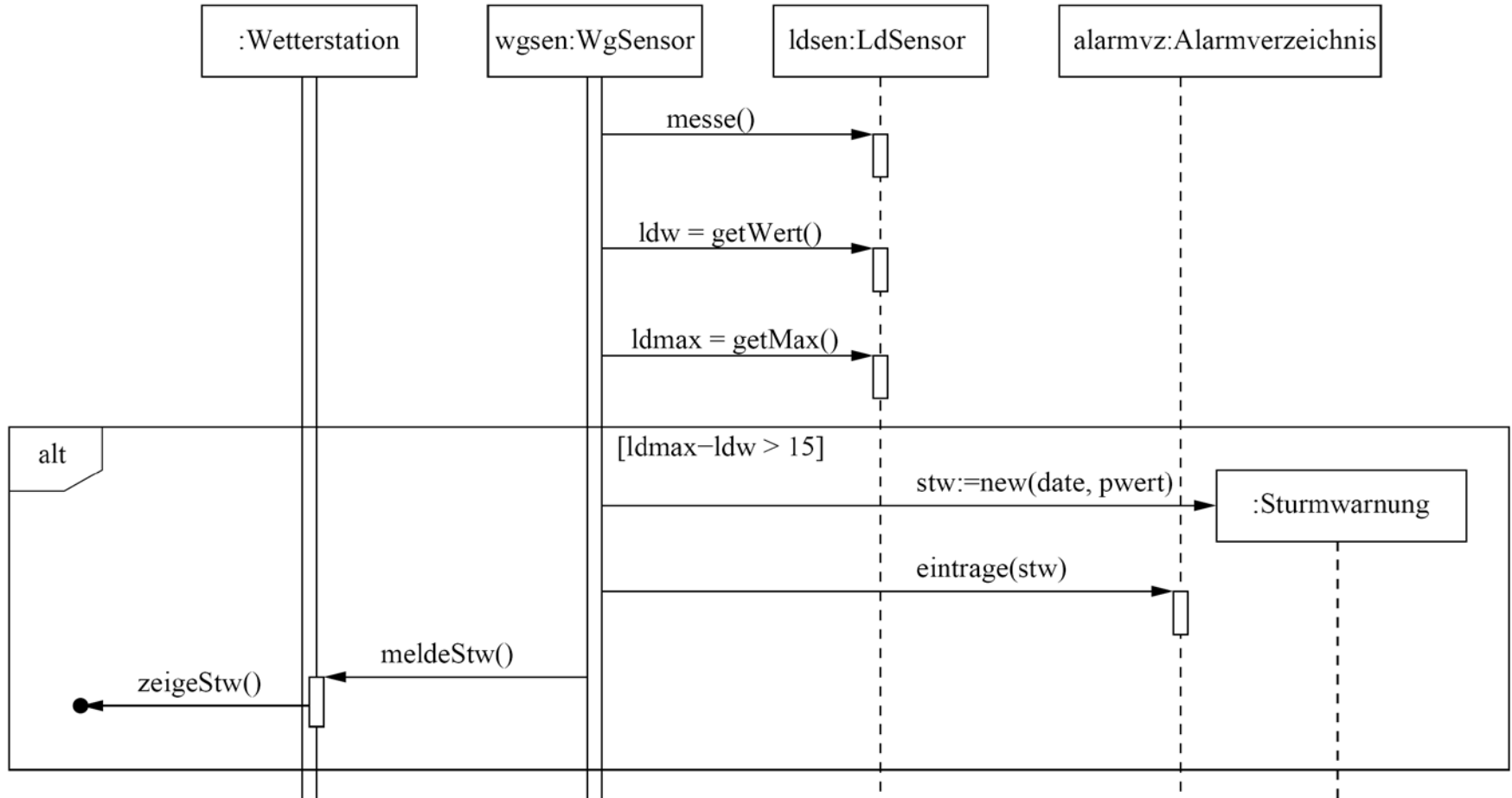
- Aktivitätsdiagramme beschreiben das Verhalten **innerhalb einer Operation**
- Interessant für Operationen mit komplexem Verhalten, z.B. Verzeigungen, Parallelität, etc.
- **Vorgehen:**
 - Analyse der Sequenzdiagramme, in welchen die gewählte **Operation** modelliert ist
 - Transformation der lokalen Operationsaufrufe in Aktivitäten und Aktivitätsübergänge

- `messe()` des **Windgeschwindigkeits-Sensors** ist interessant, da in zwei Sequenzdiagrammen vorhanden:
 - Windgeschwindigkeit messen (nächste Folie)
 - Sturmwarnung bearbeiten (übernächste Folie)
- Hands-On: Erzeugung eines Aktivitätsdiagramms

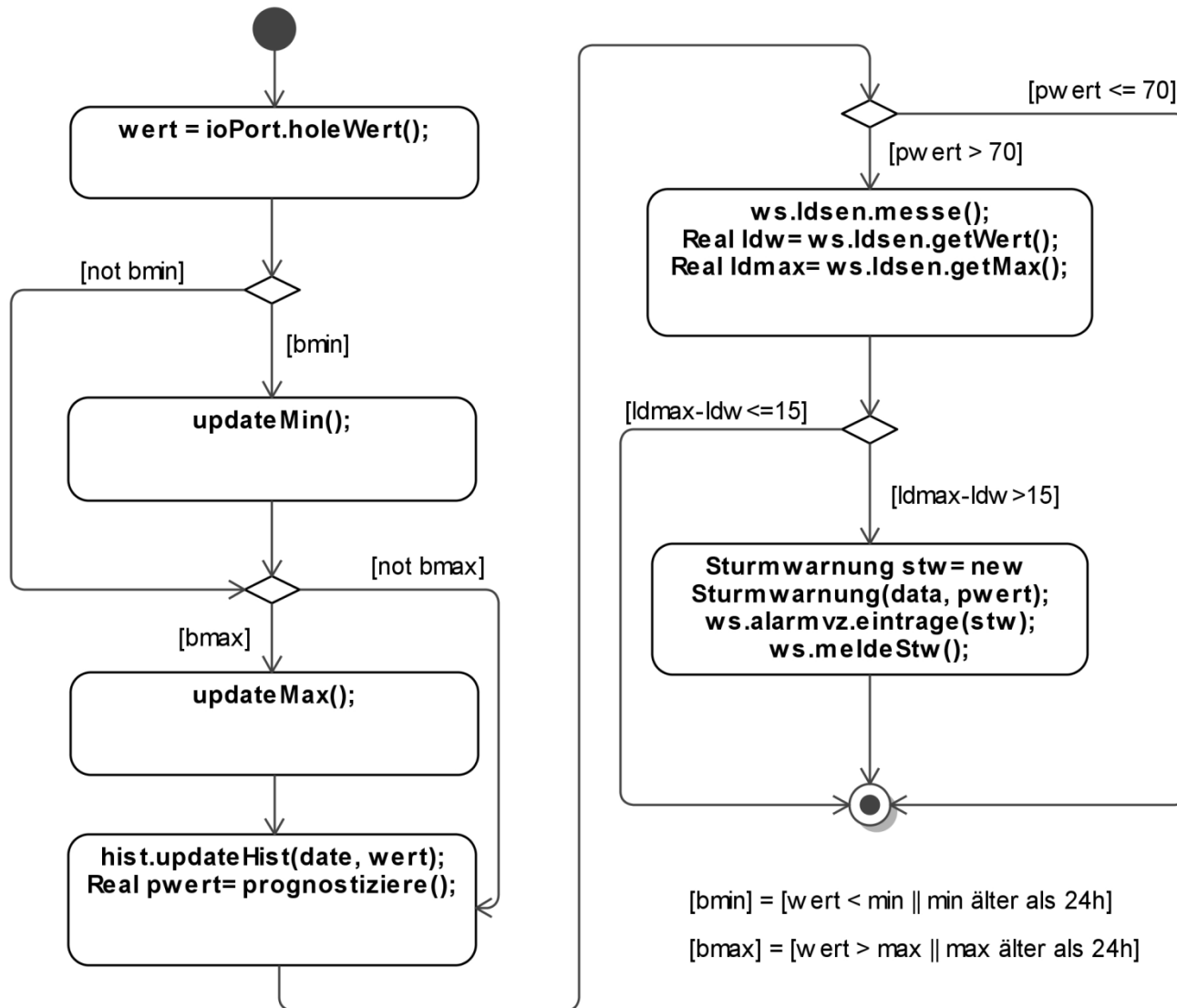
Sequenzdiagramm „Windgeschwindigkeit messen“



Sequenzdiagramm „Sturmwarnung bearbeiten“



Aktivitätsdiagramm messe() des WG-Sensors



Übung 7

ENDE

messeAlles() der Wetterstation

- Ruft sich selbst mehrfach auf
- Sehr triviales Aktivitätsdiagramm ohne weitere Verzweigungen

