

Tamagotchi-Spezifikation mit Statemate

Wolfgang Wagenbichler
Bertram Schilling
Michael Frey

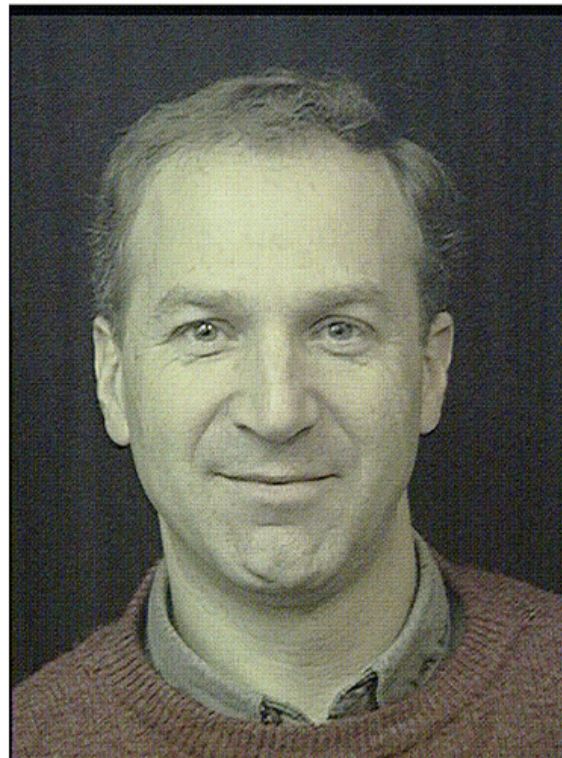
Übersicht

- Statecharts
- Notation / Methode / Tool
- Modellierung im Team
- Ausschnitt der Spezifikation
- Erfahrungen
- Fazit



Statecharts (1)

- für Spezifikation und Design großer reaktiver Systeme entwickelt (D. Harel, 1983)



Statecharts (2)

- basierend auf Zustandsautomaten
 - Super- und Substates
 - Nebenläufige Zustandsautomaten
 - Zustände mit Gedächtnis
 - Events mit Bedingungen
 - hybride Zustandsautomaten

Überblick Notation / Methode

- Drei verschiedene Sichten
 - 1. structural view (Module-Chart) **Wie ?**
 - 2. functional view (Activity-Chart) **Was ?**
 - 3. behavioral view (State-Chart) **Wann ?**
- Vorgehensweise: Top-Down
- Richtlinien zur Modellierung in Handbüchern nicht dokumentiert

Überblick Tool: Statemate Magnum

- Grafische Editoren für verschiedene Charts
- textbasiertes Data-Dictionary
- besondere Features:
 - Simulation
 - Konsistenz-Check
 - Panel Graphic Editor
 - Requirements-Table
 - Kodegenerierung
 - Dokumentator

Modellierung im Team (1)

Vorgehensweise:

- In Teamarbeit:
 - Identifizierung von Einzelfunktionalitäten (spielen, füttern, etc.)
 - Festlegung des Informationsfluß auf oberster Ebene
 - Aufteilung in einzelne Aktivitäten
- In Einzelarbeit:
 - Modellierung der einzelnen Aktivitäten
 - Simulation

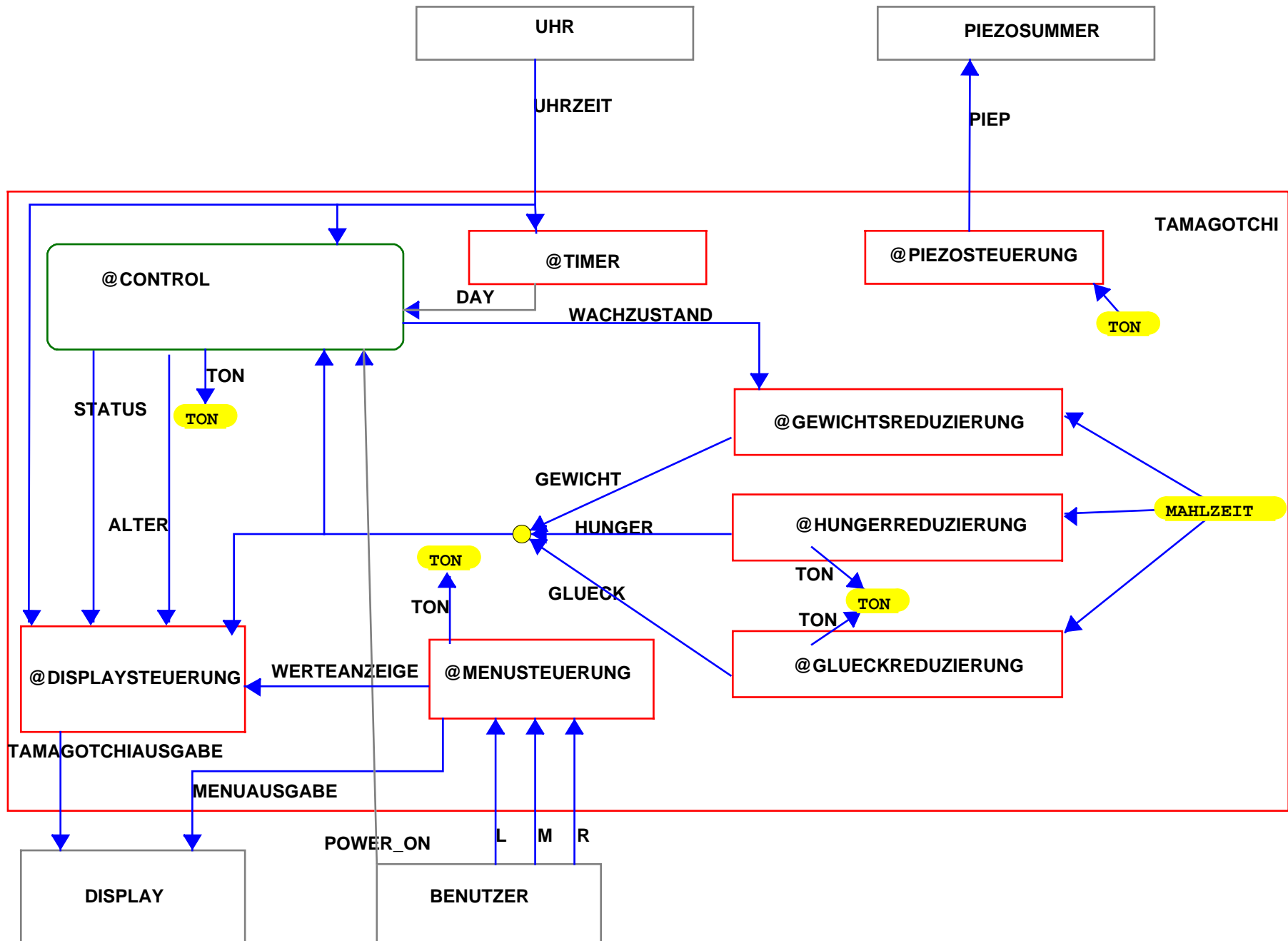
Modellierung im Team (2)

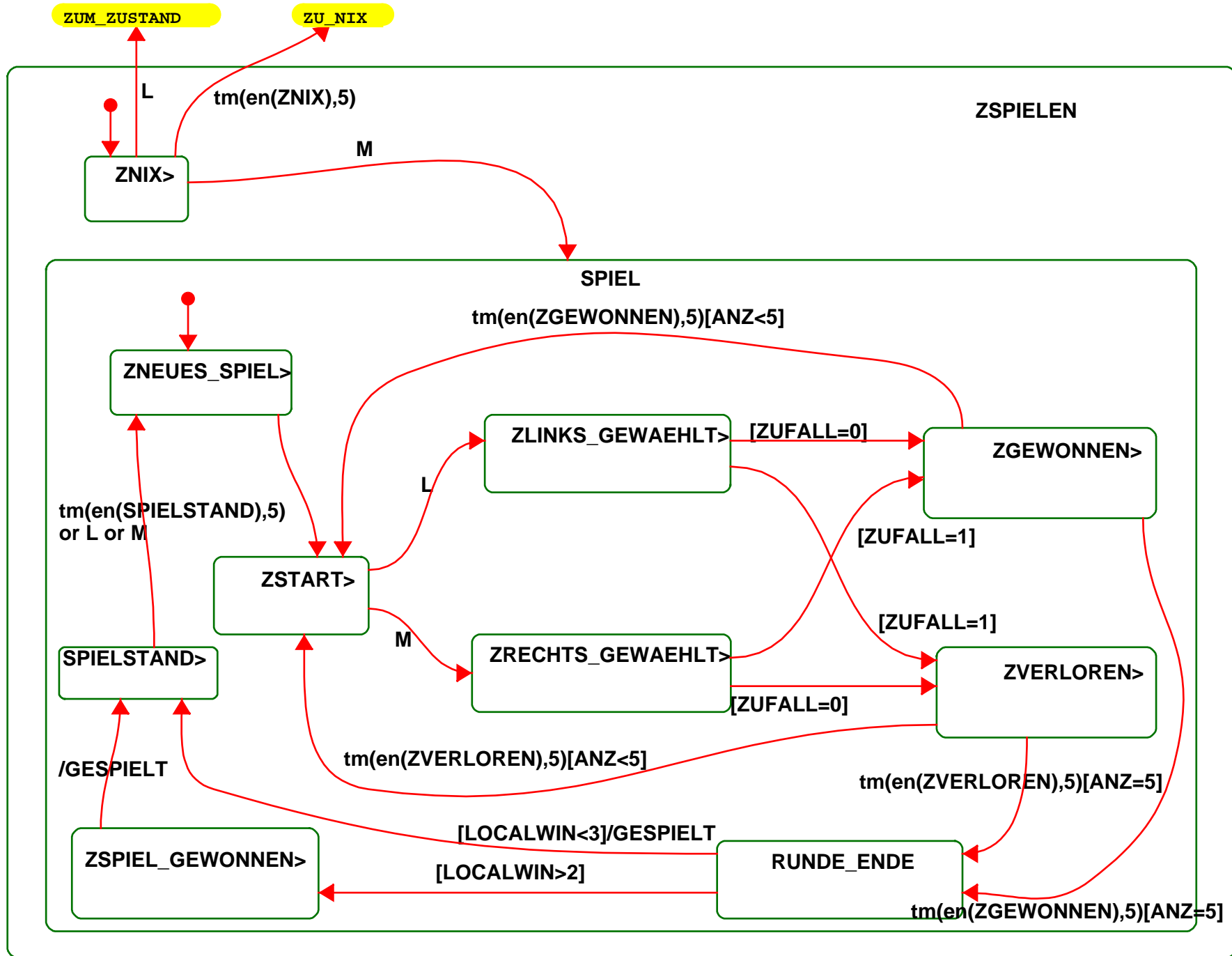
Toolunterstützung:

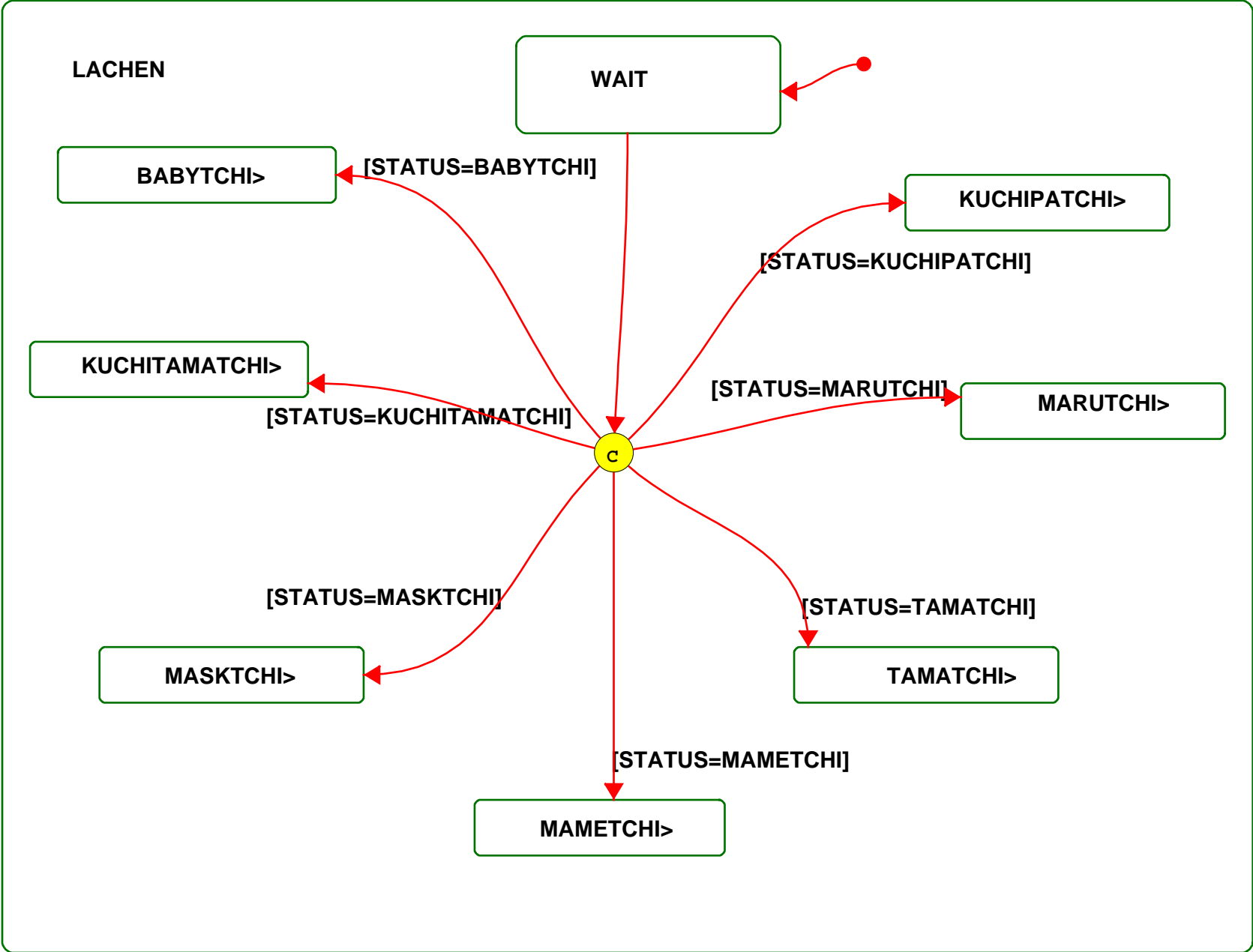
- Integrierte Versionsverwaltung
- Projektmanagementsystem

Größe der Spezifikation:

- sehr detaillierte Modellierung (Subcharts)
- vollständige Modellierung aller Tamagotchifunktionalitäten
- ca. 30 Charts + Data Dictionary Einträge







Data-Dictionary

ZNEUES_SPIEL

Defined in chart: ZSPIELEN

Static reactions:

```
entering/ANZ:=0;LOCALWIN:=0;  
      WERTEANZEIGE:=SPIELEN;  
      fs!(GEWONNEN);
```

ZSTART

Defined in chart: ZSPIELEN

Static reactions:

```
entering/WERTEANZEIGE:=SPIELEN;TICK;;  
tm(TICK,1)/TON;TICK
```

ZLINKS_GEWAEHLT

Defined in chart: ZSPIELEN

Static reactions:

```
entering/ZUFALL:=RAND_IUNIFORM(0,1);
```

ZGEWONNEN

Defined in chart: ZSPIELEN

Static reactions:

```
entering/ANZ:=ANZ+1;  
      LOCALWIN:=LOCALWIN+1;  
      WERTEANZEIGE:=LACHEN;
```

ZSPIEL_GEWONNEN

Defined in chart: ZSPIELEN

Static reactions:

```
entering/tr!(GEWONNEN);
```

Erfahrungen (1)

Einarbeitung

- SafetyInjection-Beispiel sehr einfach
→ Grundvorgehensweise von Statemate nicht erfaßt
- Aufwand ca. 30 Stunden

Erlernbarkeit

- Statecharts sind intuitiv
- Umfangreiche Online-Dokumentation

Erfahrungen (2)

Modellierung des Tamagotchi

- Statecharts gut zur Modellierung geeignet
- Keine Einschränkungen der Tamagotchifunktionalität durch das Tool
- Aufwand ca. 100 Stunden für Modellierung + Simulation

Anfängerprobleme

- Hierarchiebildung von Zuständen
- Bedienung des Tools

Erfahrungen (3)

Vergleich zu reviewten Spezifikation

	Statemate	Octopus
Verständlichkeit	<ul style="list-style-type: none"> + formal definiert + leicht verständlich - sehr detailliert + Systemverhalten nachvollziehbar 	<ul style="list-style-type: none"> - zu informell - viele verschiedene Diagrammtypen + Detaillierungsgrad frei wählbar - Systemverhalten teilweise nicht spezifiziert
Verfolgbarkeit von Anforderungen	<ul style="list-style-type: none"> + Toolunterstützung bei Verfolgbarkeitsinformationen 	<ul style="list-style-type: none"> - manuell
Überprüfbarkeit der Spezifikation	<ul style="list-style-type: none"> + Toolunterstützung (Konsistenzcheck) 	<ul style="list-style-type: none"> - verschiedene Sichten - besonders Toolunterstützung fehlt

Fazit

zur Teamarbeit:

- auf der obersten Ebene schlecht realisierbar
 - Informationsfluß festlegen
 - Verwendete Events informell beschreiben
- danach gut parallelisierbar

Stärken und Schwächen des Tools:

- + Simulation
- + unterstützt Teamarbeit
- lange Einarbeitung nötig