



UNICAM
UNIVERSITÀ DI CAMERINO



X. Specifications (IV)

Outline

▼ Z Notation



Zed main characteristics

- ▼ Formal notation that combine different features of the formalisms already presented
- ▼ Provide a simple way to **compose/decompose** a system into smaller pieces called schema
- ▼ Describe both static and dynamic aspects of the system
- ▼ Through **refinement steps** could bring from specifications to implementation



▼ Schema describe static:

- State the system can occupy
- Invariant relationship that are maintained as the system moves from states to state

▼ and dynamic aspects:

- Operations that are possible
- Relationship between their inputs and outputs
- The changes of state that happen



The Birthday Book

▼ We want to develop a book that allow us to store anniversaries and than alert us when the date is coming

▼ [*NAME* , *DATE*] relevant sets for the specification

▼ *BirthdayBook*

known : P NAME

Birthday : NAME \rightarrow DATE

known = dom *birthday*

The Birthday Book

▼ A possible state:

Known = {John, Mike, Susan}

birthday = {John --> 25-Mar,
Mike --> 20-Dec,
Susan --> 20-Dec }

▼ Any person can have only one birthday (it is a function)

▼ Add an operation

▼ *AddBirthday*

Δ *BirthdayBook*

name? : NAME

date? : DATE

name \notin *known*

birthday' = *birthday* \cup {*name?* --> *date?*}

The Birthday Book

▼ We expect that:

$$Known' = Known \cup \{name?\}$$

in fact it can be proven from the specification

▼ *FindBirthday*

\exists *BirthdayBook*

name? : NAME

date! : DATE

name \in *known*

date! = *birthday* (*name?*)



Strengthening the Spec – Schema Calculus

▼ We expect that:

▼ *Success*

result! : REPORT (REPORT ::= ok | already_known | not_known)

result! = ok

▼ Combining the spec: AddBirthday \wedge Success

▼ *AlreadyKnown*

\exists BirthdayBook

name? : NAME

result! : REPORT

name? \in known

result! = *already_known*

▼ $RAddBirthday = (AddBirthday \wedge Success) \vee AlreadyKnown$

Discussione Progetti

Documento dei Requisiti

- ▼ 1. **Introduzione** (perché è necessario sviluppare il sistema - e quali sono i problemi che vogliono essere risolti tramite l'introduzione del sistema nell'organizzazione)
- ▼ 2. **Glossario** (descrizione dei termini usati, degli attori e dei concetti introdotti nel documento)
- ▼ 3. **User Requirements** (descrizione dell'uso del sistema da parte del cliente tramite Use cases, e descrizione di possibili requisiti non funzionali)
- ▼ 4. **Architettura del Sistema**
- ▼ 5. **System Requirements** (descrizione più dettagliata tramite casi d'uso e scenari che definiranno come il sistema interagirà anche con gli altri sistemi - vedi uso di diagrammi di sequenza)
- **Modelli che descrivono il Sistema**
- ▼ 6. Eventuali appendici



Element in a Use case definition

- ▼ Preface
- ▼ Attori e lista degli obiettivi
- ▼ Precondizioni e postcondizioni
- ▼ Principale scenario di successo
- ▼ Estensioni – (si compone di condizione e gestione)
- ▼ Requisiti speciali
- ▼ Technologie

