C340 Concurrency: Modelling Concurrency in FSP

Wolfgang Emmerich

Mark Levene

What do we have to model?

- Relative or absolute speed?
  - Neither!
- Concurrency or parallelism?
  - Interleaved model of concurrency!
- Relative order of actions?
  - Arbitrary interleaving!
- We use an asynchronous model of execution!

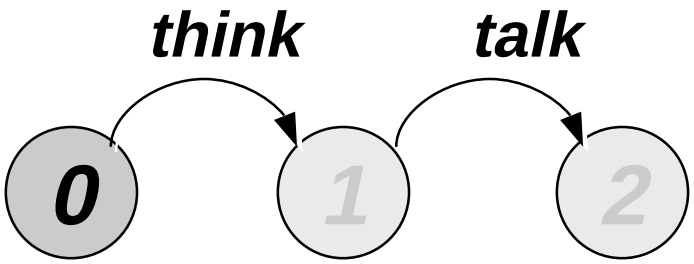
FSP: Parallel Composition

- *If* `P` *and* `Q` *are processes then* `(P||Q)` *denotes the parallel execution of* `P` *and* `Q`
- `||` *is used to model parallel composition of processes*
- *Names of concurrent processes are preceeded by* `||`
- *Example:*

```
CONVERSE =(think->talk->STOP).
ITCH =(scratch->STOP).
||CONVERSE_ITCH =(ITCH || CONVERSE).
```

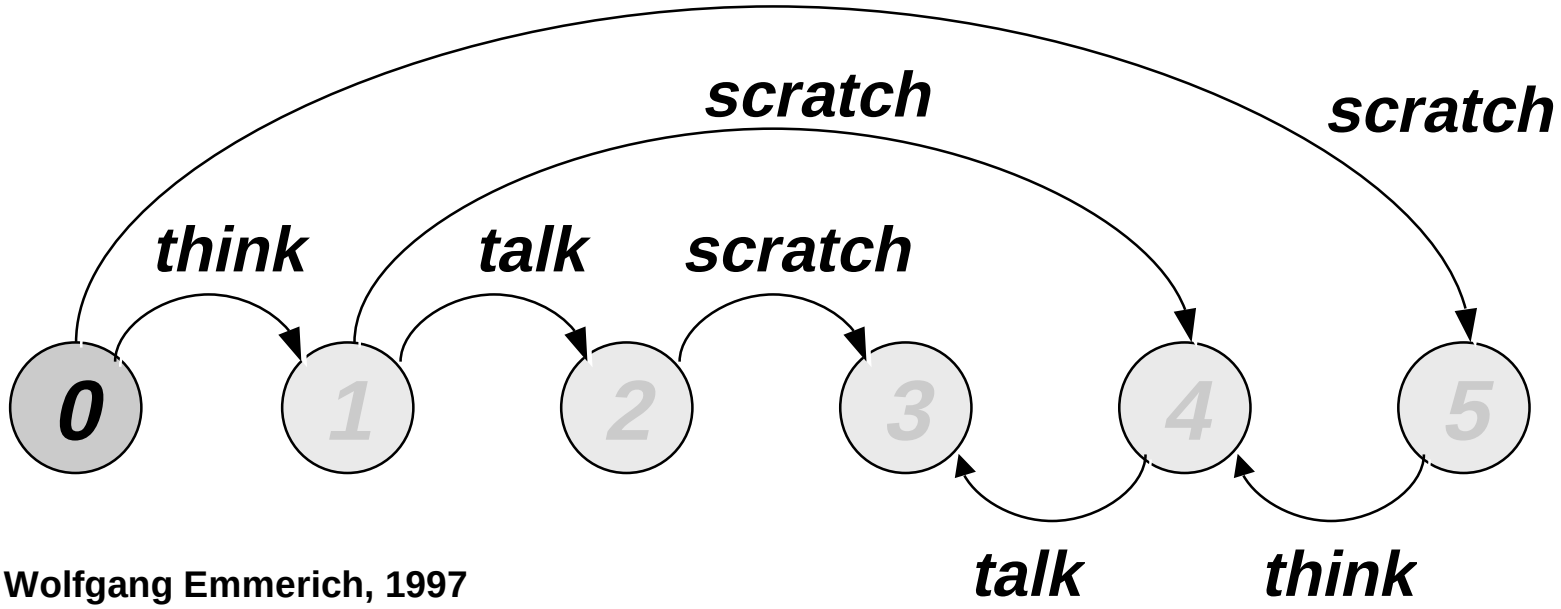
Equivalent LTSs

```
CONVERSE =(think->talk->STOP).
```




```
ITCH=(scratch->STOP).
```


```
||CONVERSE_ITCH =(ITCH || CONVERSE).
```



Properties of Parallel Composition

- *Parallel composition operator has two important algebraic properties*
- *Commutativeness*
  - `(P||Q)=(Q||P)`
  - *ordering is not important!*
- *Associativeness*
  - `((P||Q)||R)=(P||(Q||R))=(P||Q||R)`
  - *brackets can be omitted!*

FSP: Process Interactions

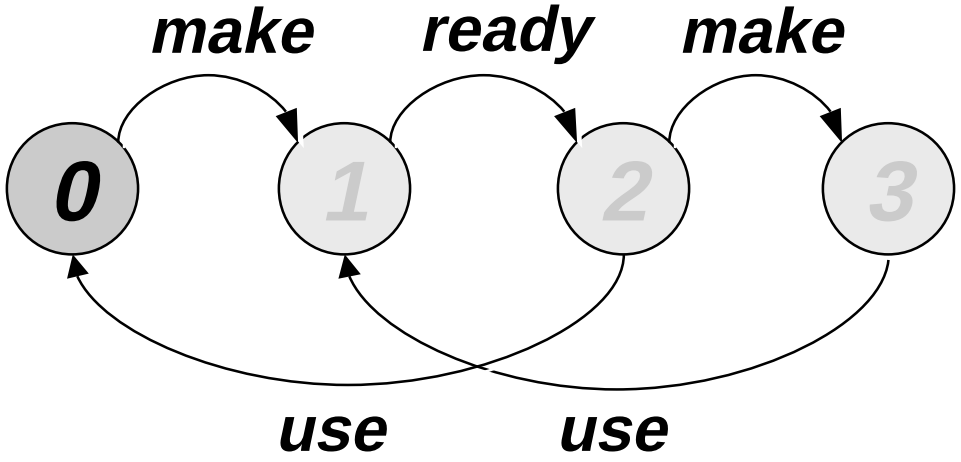
- *Concurrent processes that share actions <u>interact</u> with each other*
- *Used to model synchronisation*
- *Example:*

```
MAKER = (make->ready->MAKER).
USER  = (ready->use->USER).
||MAKER_USER = (MAKER || USER).
```

- *Product has to be ready before it can be used.*

Equivalent LTS

```
MAKER = (make->ready->MAKER).
USER  = (ready->use->USER).
||MAKER_USER = (MAKER || USER).
```

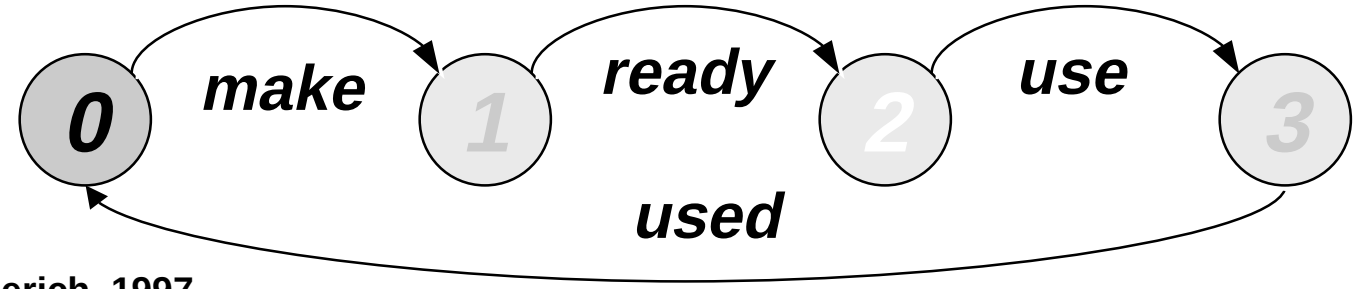


Handshake

- *An action that is acknowledged by another action is referred to as handshake*
- *Widely used to structure process interactions*
- *Example:*

```
MAKERv2=(make->ready->used->MAKERv2).
USERv2 =(ready->use->used ->USERv2).
||MAKER_USERv2 = (MAKERv2 || USERv2).
```

- *LTS:*



FSP: Process Labelling

- *The process label* `a:P` *prefixes each label in the alphabet of* `P` *with* `a`
- *Avoids name clashes in different instantiations of processes and enables reuse.*
- *Example:*

```
SWITCH = (on->off->SWITCH).
||TWOSWITCH=(a:SWITCH||b:SWITCH).
```

- *Alphabet of* `||TWOSWITCH`*:*

```
{a.on, a.off, b.on, b.off}
```

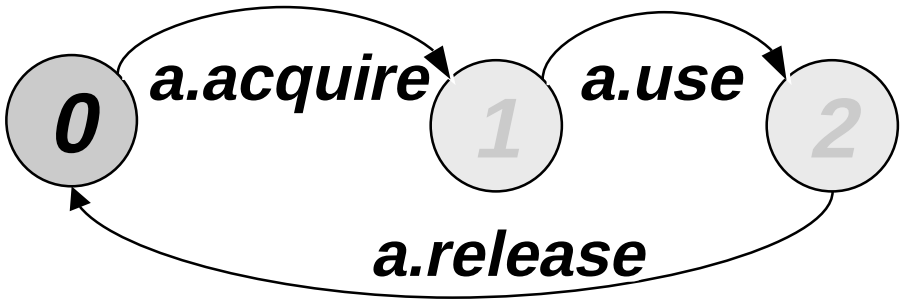
FSP: Process Labelling (cont'd).

- *The process label* `{a1,..,ax}::P` *replaces every label* `n` *in the alphabet of* `P` *with label* `a1.n,…,ax.n`.
- *Example:*

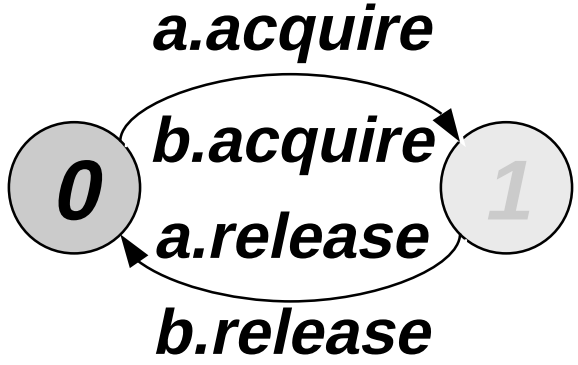
```
RESOURCE=(acquire->release->RESOURCE).
USER = (acquire->use->release->USER).
||RESOURCE_SHARE =
 (a:USER || b:USER || {a,b}::RESOURCE).
```

Equivalent LTSs

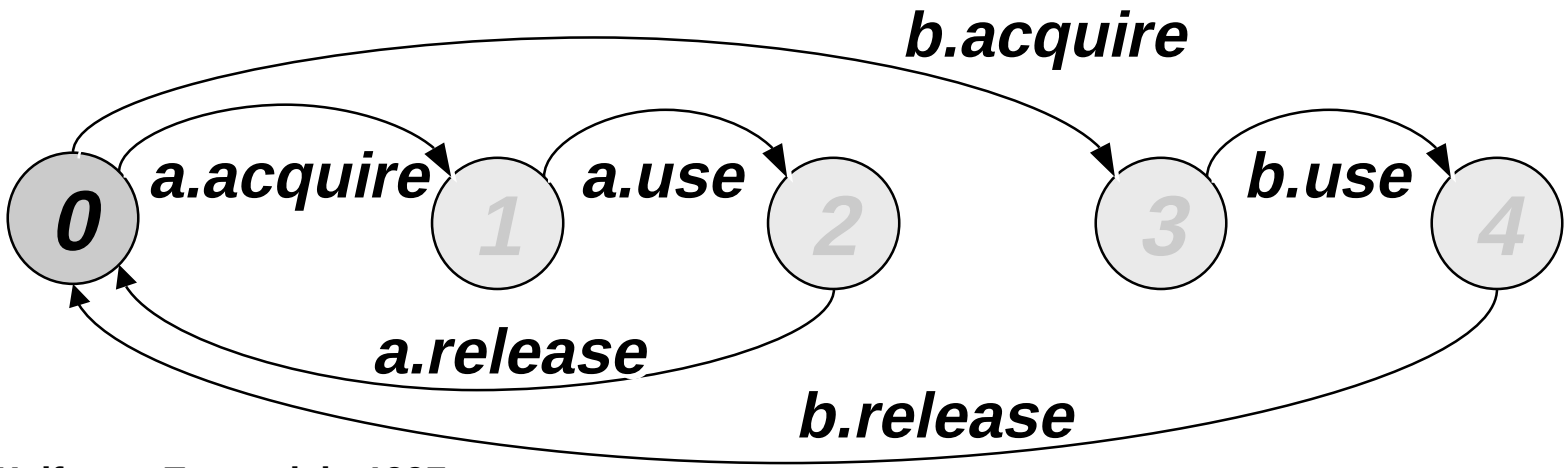
a:USER



{a,b}::RESOURCE



a:USER || b:USER || {a,b}::RESOURCE



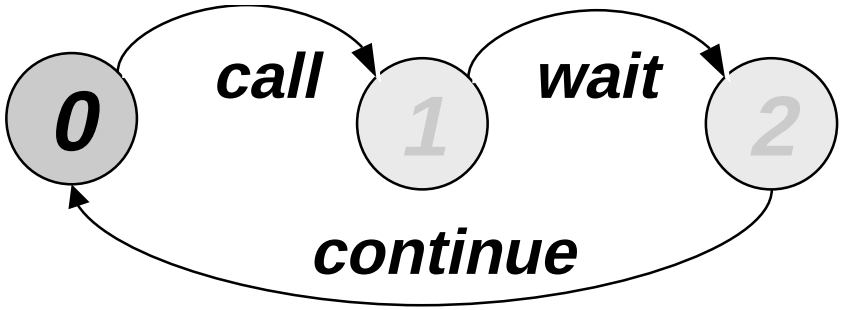
FSP: Relabelling

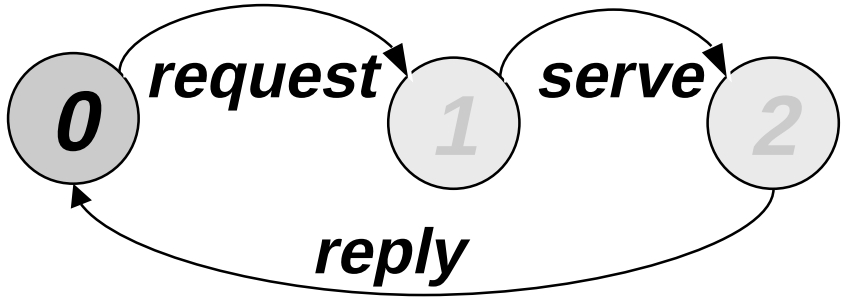
- ***Relabelling functions change names of action labels. The relabelling function is:*** **`/{new1/old1,… newn/oldn}.`**
- ***Used to synchronise actions with different names in composite processes.***
- ***Example:***

```
CLIENT=(call->wait->continue->CLIENT).
SERVER=(request->serve->reply->SERVER).
||CLIENT_SERVER = (CLIENT || SERVER)
   /{call/request, reply/wait}.
```

Equivalent LTSs

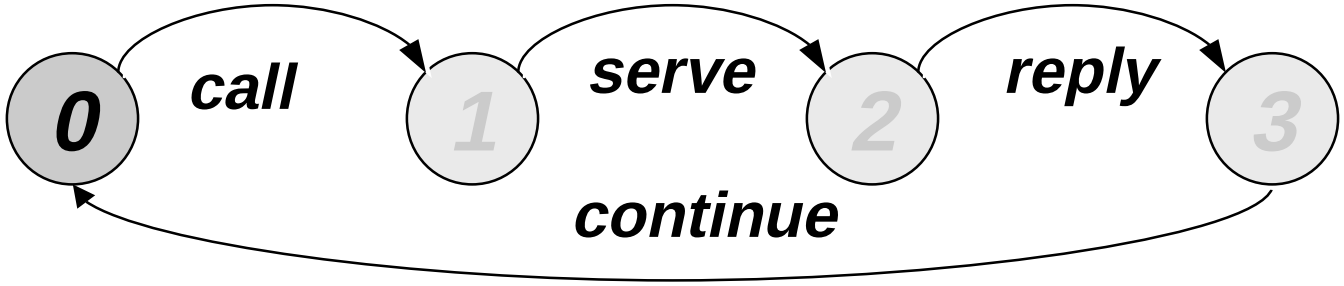
```
CLIENT=(call->wait->
continue->CLIENT).
```






```
SERVER=(request->serve
->reply->SERVER).
```

```
||CLIENT_SERVER = (CLIENT || SERVER)
          /{call/request, reply/wait}.
```



FSP: Hiding

- *The hiding operator* **/{a1..ax}** *removes action labels* **a1..ax** *from alphabet of* **P** *and hides them*
- *Hidden actions are labelled* **tau**
- *Hidden actions in different processes are not shared*
- *Example:*

```
USER=(acquire->use->release->
        USER)\{use}.
```

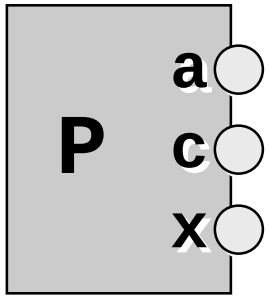
FSP: Interfaces

- *The interface operator* `@{a1..ax}` *hides all actions in the alphabet of* `P` *that do not occur in the set* `a1..ax`*.*
- *Complementary to hiding*
- *Like hiding used to reduce complexity of resulting LTS.*
- *Example:*

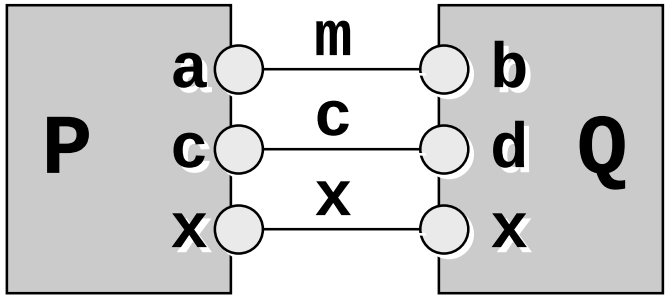
```
USER = (acquire->use->release->
          USER)@{acquire,release}.
```

FSP: Structure Diagrams

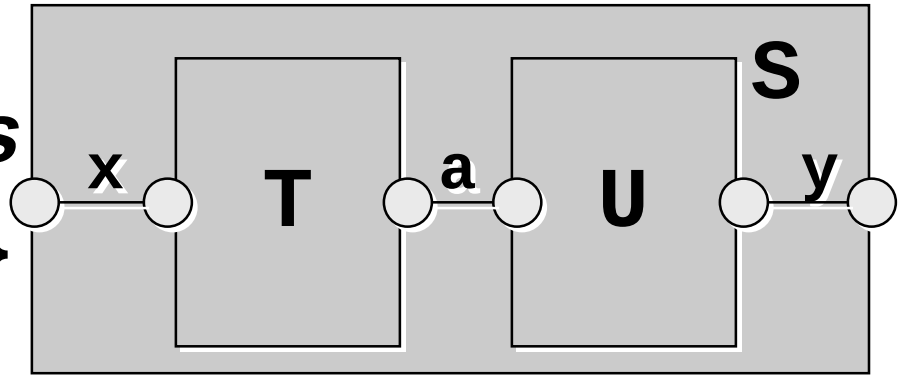
Process P with alphabet `{a,c,x}`



Parallel Composition `(P||Q)/{m/a,m/b,c/d}`



Composite Process `||S = (T||U)@{x,y}`



Summary

- *Parallel Composition*
- *Process Interactions*
- *Process Labelling*
- *Process Relabelling*
- *Hiding / Interfaces*
- *Structure Diagrams*
- *Next session: Tutorial on FSP modelling*
- *Solve Exercises 3 and 4 of tutorial sheet*