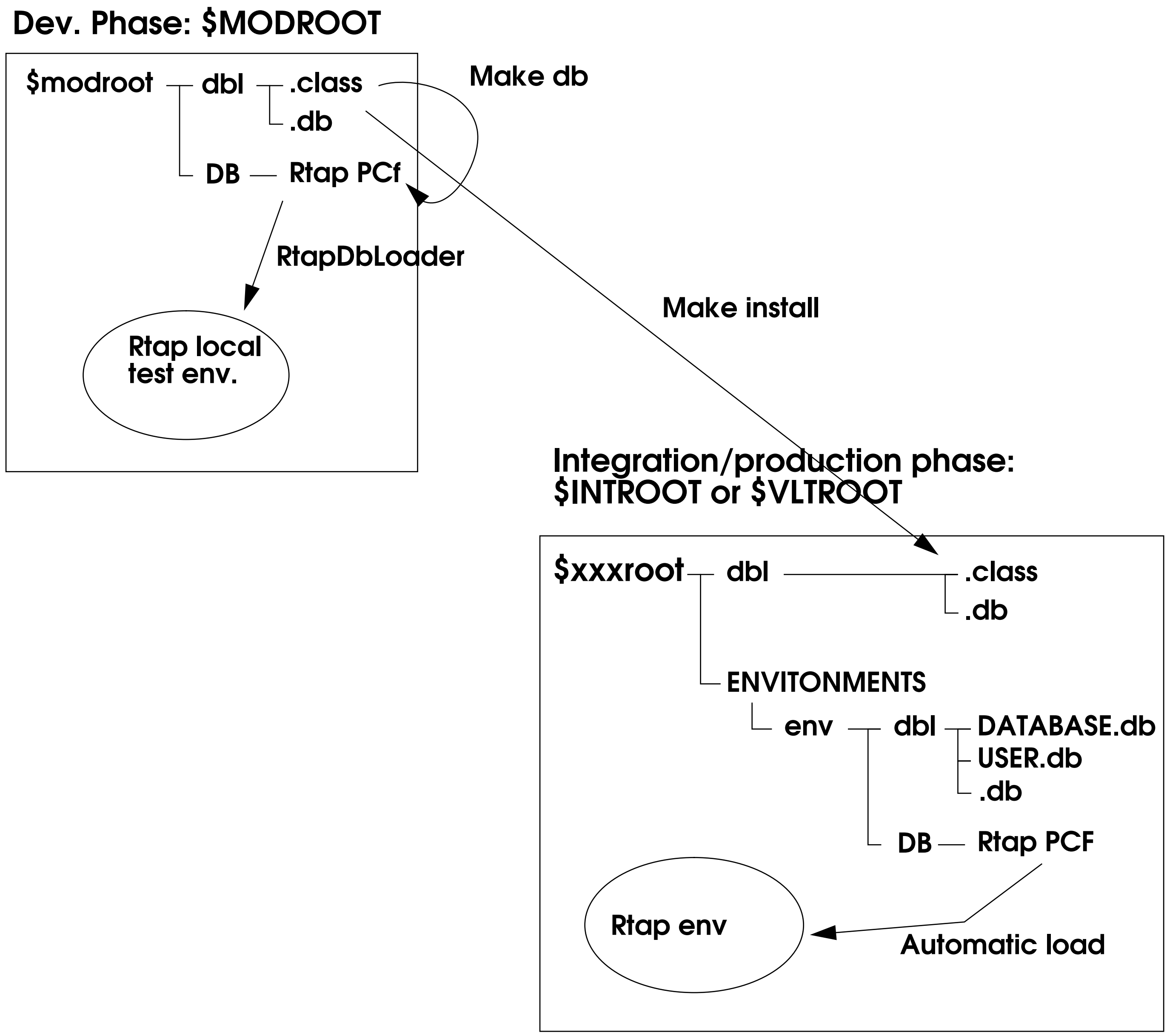
CCS Database Loader Tutorial

G.Chiozzi
03/04/95

Development environment



Makefiles

Targets:

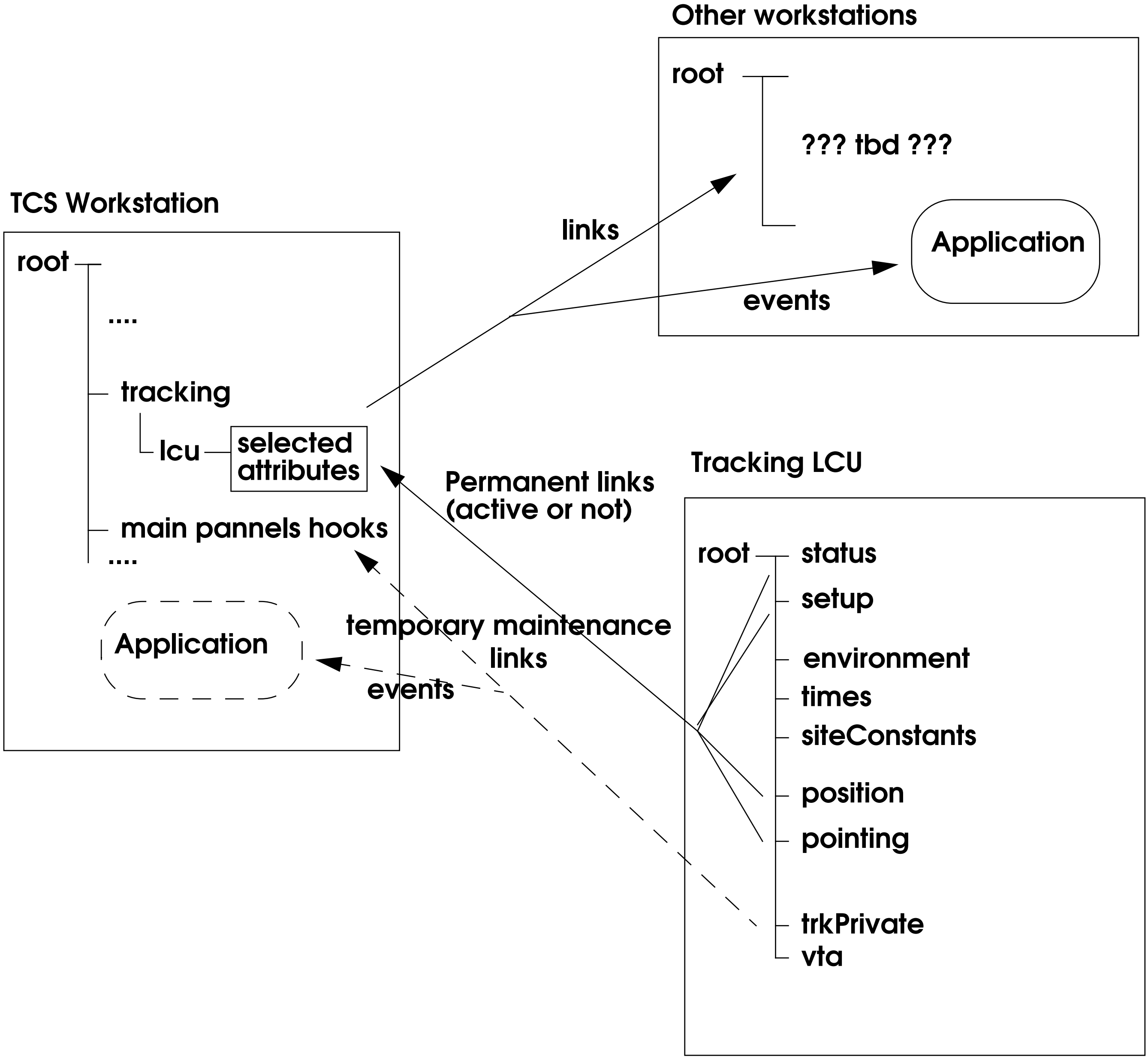
- db
- install
- clean

Keywords:

```
#
# On-Line Database Files
# ----------------------
DBL_CLASSES =
DBL_BCF =
DBL_FLAGS =
xxxx_DBL_FLAGS =

MAKE_VXWORKS = on
```

Tracking Axis Database



Scan Links

•Symmetrical classes

Classes are defined using conditional compilation to support pre-designed scan links.

•Application scan links

Scan links are designed during application development

•Dynamic scan links

Scan links are dynamically configured at run time

Symmetrical classes

- **Class definition using cond. compilation**

```
CLASS BASE_CLASS SHARED_WS_LCU
BEGIN
  #ifdef MAKE_VXWORKS
    ATTRIBUTE  int    lcuAttr
  #else
    ATTRIBUTE  int    wsImage
    BEGIN
SCAN_POLL ATTR_PATH(LCU_PATH,lcuAttr) 2.0
    END
  #endif
END
```

- **Instance on lte23**

```
POINT SHARED_WS_LCU myLte23Point
```

- **WS image for the instance on lte23**

```
#define LCU__PATH
#include “ws-lcu.class”

POINT SHARED_WS_LCU myWsPoint
```

Usage of ALIAS

- Aliases are defined at point instance time.
    - An alias is used to identify the root of a branch
    - Relative addressing starting from the alias is used to navigate in the branch.
- It is not possible to define aaliases for classes

- Usefull extensions?:
- – Alias for classes used only once in the database
- – Alias assignment for subpoints in a class
- – Alias defined with an easier syntax (outside of BEGIN..END block)

DB - code interaction

- Access functions to database attributes:
-
  - Structures parallel to points
  - Functions to load a full point in a structure
  - More independent from database struct.

- Classes (in C++) parallel to database classes:
  - More formal way to implement previous concepts
  - Architecture more easy to understand and design