NAME

cluster − CHSM cluster class

SYNOPSIS

```
#define CHSM_CLUSTER_ARGS     /* ... */
#define CHSM_CLUSTER_INIT     /* ... */

namespace Concurrent_Hierarchical_State_Machine {

    class cluster : public parent {
    public:
        cluster( CHSM_CLUSTER_ARGS );
        void clear();

        // inherited

        typedef state value_type;
        typedef value_type* pointer;
        typedef value_type const* const_pointer;
        typedef value_type& reference;
        typedef value_type const& const_reference;

        class iterator;
        class const_iterator;

        bool active() const;
        iterator begin();
        const_iterator begin() const;
        virtual void deep_clear();
        bool empty() const;
        iterator end();
        const_iterator end() const;
        virtual bool enter( event const &trigger, state* = 0 );
        virtual bool exit ( event const &trigger, state* = 0 );
        reference front();
        const_reference front() const;
        char const* name() const;
        parent* parent_of() const;
    };

}
```

DESCRIPTION

A `cluster` *is-a* **CHSM::parent**(3) that represents a logical-exclusive-or grouping of its child states where *exactly one* child-state is active at any given time.

A `cluster` may have a history. If a `cluster` does not have a history, or it does have a history but has not previously been active, then the child-state entered after it itself is entered is the *default* child-state; if a `cluster` does have a history and it has been visited before, then the child-state entered after it itself is entered is the one that was last active.

A `cluster` may alternatively have a *deep* history. Such a `cluster` behaves exactly as one with an ordinary history; the difference is that all `cluster`s lexically-enclosed by it in the CHSM description also have a history.

Member Function

`void clear()`

Clears the history of this cluster only.

SEE ALSO

CHSM::cluster(3), **CHSM::parent**(3), **CHSM::state**(3), **chsm-c++**(4)

AUTHORS

Paul J. Lucas <*paul@lucasmail.org*>
Fabio Riccardi <*fabio.riccardi@mac.com*>