Two Examples for critical Template Problems (from Tokyo)

Erwin Unruh
Siemens Nixdorf, Germany

Here are my examples from Tokyo regarding the indirect instantiation. The first is unique for seperate translation units. The second was formulated using different translation units, which I reformulated for namespace usage.

```
// ****************************************************

// unit 1                       // unit 2

class D;                        class D { int dd; };
class A { D* aa; };             class A;
                                class B {}

                                A* f(B);

template <class T>              template <class T>
void foo (T t)                  void foo (T t);
{
        f(t)->aa->dd++;
}
                                void bar ()
                                {
                                        foo( B() );
                                }

// ****************************************************

// unit 1               // unit 2               // unit 3

template <class T>
void f(T*);
template <class T>      template <class T>
void g(T*);             void g(T*);

class A1{};             class B1{};
class A2{};             class B2{};

void h(A1*);            void h(B1*);
void h(A2*);            void h(B2*);

A1 *a1;                 B1 *b1;
A2 *a2;                 B2 *b2;

                        template <class T>      template <class T>
                        void f(T* t)            void g(T* t)
                        {                       {
                                g(t);                   h(t);
                                g(b1);
                        }                       }

                        // template <>
                        // void f<B2>(B2* t)
void bar()              void foo (B2* t)
{                       {
        f(a1);                  g(t);
        g(a2);                  g(b1);
}                       }


// ***** and here the version using namespaces (assume the three
```

```
// columns are a single translation unit )

// ***************************************************


namespace N2 {
template <class T>
void f(T*);
} // N2
namespace N3 {          namespace N3 {
template <class T>      template <class T>
void g(T*);             void g(T*);
} // N3                  // N3

namespace N1 {          namespace N2 {          namespace N3 {

class A1{};             class B1{};
class A2{};             class B2{};

void h(A1*);            void h(B1*);
void h(A2*);            void h(B2*);

A1 *a1;                 B1 *b1;
A2 *a2;                 B2 *b2;

                        template <class T>      template <class T>
                        void f(T* t)            void g(T* t)
                        {                       {
                                N3::g(t);               h(t);
                                N3::g(b1);
                        }                       }

                        // template <>
                        // void f<B2>(B2* t)
void bar()              void foo (B2* t)
{                       {
        N2::f(a1);              N3::g(t);
        N3::g(a2);              N3::g(b1);
}                       }


} // N1                 } // N2                 } // N3
```