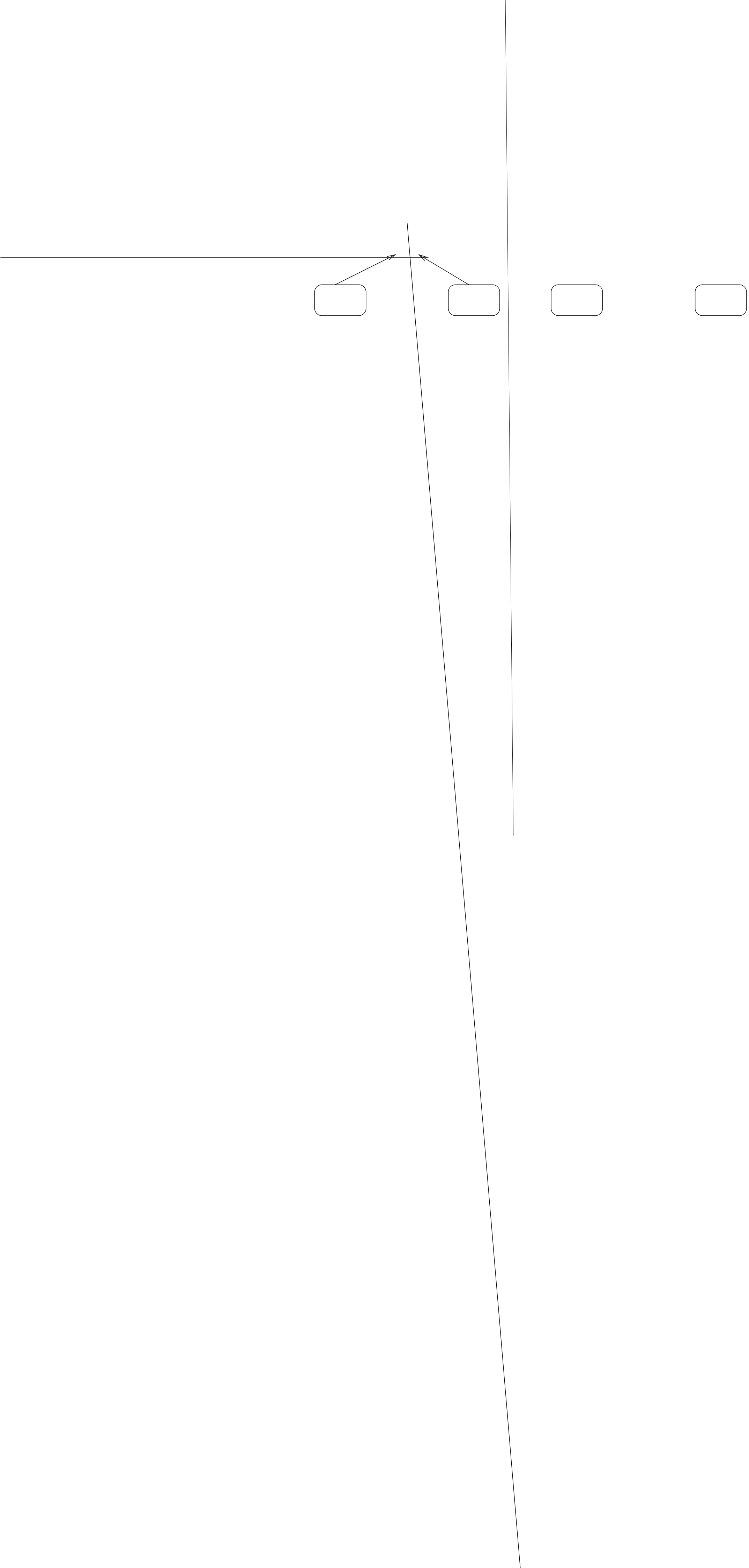
The EYE; B OODBMS

Eric



instantiation
inheritance
indirect inheritance

4.5. Constraints

EYEDB supports all standard constraints:

- the `not null` constraint on a attribute within a class `X` means that no instances of class `X` can have attribute value not assigned.

- the `unique` constraint on a ute within a class `X` means that one cannot create an instance of class `X` hasameute value nribonv instance

the ionstaonfce classon

As the attribute spouse is not a collection and the target attribute spouse is not a collection, this is a one-to-one a specific C++ API from a given schema, built upon the generic API.

Each class in the EYEDB object model is implemented as a C++ class within the C++ API: there is a one-to-one mapping between the object model and the C++ API.

7.1. Transient and Persistent Objects

There are two types of runtime objects: persistent runtime objects and transient runtime objects.
A runtime object is persistent if it is tied to a database object. Otherwise, it is transient.

By default, EYEDB does not provide an automatic synchronisation between persistent runtime objects and database objects.
Wh5 0 Td49.59 8.8(3)Tj36(setting)Tj-(the)Tj-1.58-4(o)Tj(2.9)Tj(461)Tj0 Td 9.96(2)Tj1E)Tj61.3 R18 Td concrete)Tj0 Td13.(b)27 0 Td (a)Tj

The Java binding is very close from the C++ binding: the

9.1. The Generic CORBA Binding

The major external component of the generic binding is the EYEDB IDL. The EYEDB IDL is composed of one module (named EyeDB _ORB) which includes about forty interfaces and a few

```
// instantiating a Person
Person_var person = person_factory->
            makePerson();

// set
```

EYEDB is now about 200,000 lines of