Objectivity Case History

CUSTOMER: LMS Medical Systems
Industry: Medical
Application Domain: Computer Assisted Labor Monitoring (CALM) for Obstetrical care units. Provides comprehensive patient monitoring, information management and decision support capabilities.
Status: Under development with Limited beta – Feb 98, Production by Oct 98
Platform: Windows NT
Compiler: MSVC++ 5.0
Other Tools: Active X

Ease of Use comparison versus ObjectStore (Feb 98)

Learning curve

ObjectStore has a library of over 100 classes each with approximately 10 methods per class. This is a very rich environment and could represent a sizable learning curve. The vendors claim that only a subset of the API is actually used for most applications. At least one ODI reference claimed that his team had to do a lot of fine tuning to achieve good performance. ODI recognizes this but maintains that the complex nature of that application called for the tuning.

Objectivity seems less complex. It also has tools to assist the developer in tuning as well as the capability of adjusting database parameters while the database is online. Developers can also adjust several tuning parameters within applications. In our opinion, Objectivity will require less tinkering to achieve good performance.

Programming Objectivity is more straightforward. The following code fragments should demonstrate this. The function CreatePersons() creates persons in one transaction and commits the insertions all at once. The UpdateAllPersons() function modifies the persons and checkpoints every modification. For simplicity all variables are first defined and then initialized. The ObjectStore code was provided by ODI.

Objectivity Examples

```
void CreatePersons(long l)
{
    ooTrans                         trans;
    ooHandle(ooFDObj)               fdbH;
    ooHandle(ooDBObj)               dbH;
    ooHandle(ooContObjr)    pc;

    trans.start();
    fdbH.open(FDB_NAME, oocUpdate);
    dbH.open(fdbH,DB_NAME,oocUpdate);
    pc  = new(ALLPERSONS,1,10,40,dbH) ooContObj (ALLPERSONS);

    for(long i = 0; i < l; i++)
            new(pc) Person("FirstName,"LastName");

    trans.commit();
}



BOOL UpdateAllPersons()
{
    ooTrans                         trans;
    ooHandle(ooFDObj)               fdbH;
    ooHandle(ooDBObj)               dbH;
    ooHandle(ooContObj)     pc;


    trans.start();
    fdbH.open(FDB_NAME, oocUpdate);
    dbH.open(fdbH,DB_NAME,oocUpdate);
    pc.exist(dbH,ALLPERSONS,oocUpdate))


    ooItr(Person) itr ;
    itr.scan(pc,oocUpdate);
    while(itr.next())
    {
            itr->ooUpdate();
            itr->SetFirstName("John");
            trans.commitAndHold();
    }
    trans.commit();
}
```

ObjectStore Equivalent Examples

```
void CreatePersons(long l, os_reference &ref)
{
	os_database		*db;
	os_typespec		*PersonType;
	os_transaction		*txn;
	os_Bag<Person*>&	allPersons;

	db = os_database::open("test1.db", 0, 0667);
	PersonType = new os_typespec("Person");
	txn = os_transaction::begin();
	allPersons = os_Bag<Person*>::create(db, os_collection::maintain_cursors, l);
	ref = &allPersons;

	for (long i = 0; i < l; i++)
		allPersons.insert(new (db, PersonType) Person("FirstName","LastName"));

	os_transaction::commit();
	delete txn;
}

void UpdateAllPerson(os_reference &ref)
{

	os_database		*db;
	os_transaction		*txn;
	os_Bag<Person*>		*allPersons;
	os_Cursor<Person*>	*cur;
	os_reference		cur_ref;

	db = os_database::open("test1.db", 0, 0667);
	txn = os_transaction::begin();
	allPersons = (os_Bag<Person*>*)ref.resolve();
	cur =	new (db, os_Cursor<Person*>::get_os_typespec())
	os_Cursor<Person*>(*allPersons, os_cursor::safe);
	cur_ref = cur;

	for (Person *p = cur->first(); cur->more(); p = cur->next())
	{
		p->SetFirstName("John");
		os_transaction::checkpoint();
		cur = (os_Cursor<Person*>*) cur_ref.resolve();
	}
	delete cur;
	os_transaction::commit();
	delete txn;
}
```

Contact Information

Objectivity Sales Manager: Vincent R. Ramoutar
Customer Contact: Mario Boisclois, Database Architect
Phone:
Email: