Mitosis Usecase Scenarios

This page is intended to identify and discuss all possible usecase scenarios of Mitosis during the refactoring process

1) When a same entry gets created on more than one master server then only one entry needs to be taken as base and the other needs to be removed. But how do we choose which one to retain/discard.?

• At the moment the one with greater CSN (i.e. the more recent) will be retained just like with any conflicting modification. The older entry is lost - it would be useful if it were moved to a lost+found area and an administrator notified in some way.

2) When one server deletes an entry while another server adds a child entry below at the same time, what do we do to resolve the conflict?

- On a single server a delete followed by a child entry add would give an error, as would a child entry add followed by a delete. Perhaps this means we should just discard the second operation?
- It all depends on the time of each operation. On A, we delete e1 at ti : A.del(e1, ti). On B we add e2 under e1 at tj : B.add(e1/e2, tj). If ti < tj, then we have to rollback the addition on B. If ti > tj, then we have to rollback e1 deletion and add e2 on A : ti < tj => A[] & B[]; ti > tj => A[e1/e2] & B [e1/e2]

3) We have 3 connected servers : A, B and C.

- At t0, we add e1 on B. It is replicated on A and C.
- At t1, the server B is disconnected from B and C
- At tx, we delete e1 from B
- At ty, we add e2 on A, as a child of e1. It is replicated on C
- At tz, B is connecting back to A and C, everything is being replicated

If we consider the order in which the operations have occured, we can build the change lists :

- Case 1, tx < ty at tz :
 - On server A, the list of operations is : rep(add, e1, t0), rep(del, e1, tx), add(e1/e2, ty)
 - On server B, the list of operations is : add(e1, t0), del(e1, tx), rep(add, e1/e2, ty)
 - ° On server C, the list of operations is : rep(add, e1, t0), rep(del, e1, tx), add(e1/e2, ty)
- Case 2, tx > ty at tz :
 - On server A, the list of operations is : rep(add, e1, t0), add(e1/e2, ty), rep(del, e1, tx)
 - On server B, the list of operations is : add(e1, t0), rep(add, e1/e2, ty), del(e1, tx)
 - On server C, the list of operations is : rep(add, e1, t0), add(e1/e2, ty), rep(del, e1, tx)
- The problem is that we will have conflict if we apply the modifications when B reconnects to A and C. Typically :
 - On server A and C : the rep(del, e1, tx) won't be accepted, as e1 has already a child
 - On server B : the rep(add, e1/e2, ty) won't be accepted as the e1 entry does not exist anymore

Question : how do we handle this case ?