Object Transfer Service

Shankar

September 18, 2014
Overview: Object Transfer Service

- **Object**: id and mutable value // eg, page # and contents

- **Service allows systems to share objects**
  - acquire an object, change its value, release it
  - acquired value equals last-released value

- **Objects “at rest” reside with users, not service** // unlike lock
  - object’s **owner**: user that currently holds it
  - object is **unowned** if it currently has no owner
  - objects have initial owners
  - user can acquire an object // blocking
  - service can request user for an object
  - user releases object only when requested
Object transfer service

Parameters
- ADDR: set of addresses
- OID: set of object ids (oids)
- OVAL: possible values of an object
- \{initObjs_j\}: oids of objects with user at j

Main
- objs_j ← initObjs_j
  // objects at user j
- reqs_j ← set()
  // objects requested by user j
- val_{oid, for unowned oid}
  // value of obj at last release
- return \{v_j ← sid()\}
  // access system at j
Object transfer service

- `vj.acq(oid)` // acquire object and its value
  - ic { no ongoing `vj.acq(oid)` and `oid` not in `objsj` }
  - output `rval`
    - oc { `val oid` exists and `rval` = `val oid` }
    - move `oid` from `val` to `objsj`
    - return `rval`

- `vj.rel(oid, oval)` // release object and its value
  - ic { `oid` in `objsj` and in `reqsj` }
    - remove `oid` from `objsj` and from `reqsj`
    - `val oid` ← `oval`
  - oc { true }
  - return
vj.rxReq() // rcv request for object

ic { no ongoing vj.rxReq() }

output oid
   oc { ( oid not in reqsj ) and
       ( oid in objsj or ongoing vj.acq(oid) ) }
   add oid to reqsj
   return oid

atomicity assumption: input parts and output parts
Object transfer service: progress assumption

- every rel call returns
  - ongoing \( j.\text{rel}(x,v) \) \textit{leads-to} no ongoing \( j.\text{rel}(x,v) \)

- if a user wants an object then the owner is informed, provided the owner maintains an ongoing \( rx\text{Req} \) call
  - ( \( \text{objsj} \) not empty \textit{leads-to} ongoing \( j.\text{rxReq} \) ) \( \Rightarrow \)
    - ( \( x \) in \( \text{objsj} \) and ongoing \( k.\text{acq}(x) \) ) \textit{leads-to} \( x \) in \( \text{reqsj} \)

- if a user wants an object then it gets it provided the owner rcvs a request and then releases the object
  - ( \( x \) in \( \text{objsi} \) and ongoing \( k.\text{acq}(x) \) \textit{leads-to} \( x \) in \( \text{reqsi} \) )
    - and ( \( x \) in \( \text{reqsi} \) \textit{leads-to} \( x \) not in \( \text{reqsi} \) )
      - \( \Rightarrow \) ( ongoing \( j.\text{acq}(x) \) \textit{leads-to} no ongoing \( j.\text{acq}(x) \) )