config
$GAME_NAME
$CASE
...
$files
$SUPPLY_TYPES
$POSTURES
...
→ $data
$UNITS
$transportDeliveries [df]
...

$Day
$Org Unit
$SupplyLogNode
$Transport

Alternate

Game
$configs [YS-YS]
$UNITS
$TRANSPORTS
$SUPPLY_REQUESTS

Design Goals

① for arbitrary {set} of games,
allow 'tidy' data frames of key outputs
→ with factor levels not collide

② for any game in set, ~~make~~ convenient
cross-reference to object particulars
eg delivery → logNode → script → Domain
→ transport → domain restriction

③ for "all" games in set arbitrary xref as above
Game 1 LogNode Y Time=20 Domain: STABLE
Game 2 LogNode Z time=30 Domain: Set

gameSet object immutable
addGame
rmGame
df ← getUnits(gameSet)
df ← getOrgDelivs(gameSet)
df ← getFlowData(gameSet)

① Defines a contract to return tidy df of
a,b set of games
③ Arb xref via usual merge(getDelivery(gs),
getLNScript(gs))

readUNITS(lconfig)
~~get~~ units(lconfig)
lconfig$UNITS

④ can refactor implementation of persistence in memory later

get Units(ss)

shortcuts

rbind(getUnits(ss[1]), setUnits(ss[2]), ...)

with CASE columns

NO connecting factor levels

or rbind(readUnits(ss[1]), ...)

possibly caching rbind result in some-set object for later use