Contents

1 PBTypeIntegrated Theory 3

1.1 Datatypes 3

1.2 Theorems 3

2 ssmPBIntegrated Theory 4

2.1 Theorems 5

3 PBIntegratedDef Theory 11

3.1 Definitions 11

3.2 Theorems 12

1 PBTypeIntegrated Theory

Built: 10 June 2018
Parent Theories: OMNIType

1.1 Datatypes

```
omniCommand = ssmPlanPBComplete | ssmMoveToORPComplete
            | ssmConductORPComplete | ssmMoveToPBComplete
            | ssmConductPBComplete | invalidOmniCommand

plCommand = crossLD | conductORP | moveToPB | conductPB
          | completePB | incomplete

slCommand = PL plCommand | OMNI omniCommand

slOutput = PlanPB | MoveToORP | ConductORP | MoveToPB
         | ConductPB | CompletePB | unAuthenticated
         | unAuthorized

slState = PLAN_PB | MOVE_TO_ORP | CONDUCT_ORP | MOVE_TO_PB
        | CONDUCT_PB | COMPLETE_PB

stateRole = PlatoonLeader | Omni
```

1.2 Theorems

[omniCommand_distinct_clauses]

```
⊢ ssmPlanPBComplete ≠ ssmMoveToORPComplete ∧
  ssmPlanPBComplete ≠ ssmConductORPComplete ∧
  ssmPlanPBComplete ≠ ssmMoveToPBComplete ∧
  ssmPlanPBComplete ≠ ssmConductPBComplete ∧
  ssmPlanPBComplete ≠ invalidOmniCommand ∧
  ssmMoveToORPComplete ≠ ssmConductORPComplete ∧
  ssmMoveToORPComplete ≠ ssmMoveToPBComplete ∧
  ssmMoveToORPComplete ≠ ssmConductPBComplete ∧
  ssmMoveToORPComplete ≠ invalidOmniCommand ∧
  ssmConductORPComplete ≠ ssmMoveToPBComplete ∧
  ssmConductORPComplete ≠ ssmConductPBComplete ∧
  ssmConductORPComplete ≠ invalidOmniCommand ∧
  ssmMoveToPBComplete ≠ ssmConductPBComplete ∧
  ssmMoveToPBComplete ≠ invalidOmniCommand ∧
  ssmConductPBComplete ≠ invalidOmniCommand
```

[plCommand_distinct_clauses]

```
⊢ crossLD ≠ conductORP ∧ crossLD ≠ moveToPB ∧
  crossLD ≠ conductPB ∧ crossLD ≠ completePB ∧
  crossLD ≠ incomplete ∧ conductORP ≠ moveToPB ∧
```

$$\begin{array}{l}
\mathtt{conductORP} \neq \mathtt{conductPB} \land \mathtt{conductORP} \neq \mathtt{completePB} \land \\
\mathtt{conductORP} \neq \mathtt{incomplete} \land \mathtt{moveToPB} \neq \mathtt{conductPB} \land \\
\mathtt{moveToPB} \neq \mathtt{completePB} \land \mathtt{moveToPB} \neq \mathtt{incomplete} \land \\
\mathtt{conductPB} \neq \mathtt{completePB} \land \mathtt{conductPB} \neq \mathtt{incomplete} \land \\
\mathtt{completePB} \neq \mathtt{incomplete}
\end{array}$$

[slCommand_distinct_clauses]

$$\vdash \forall a'\ a.\ \mathtt{PL}\ a \neq \mathtt{OMNI}\ a'$$

[slCommand_one_one]

$$\begin{array}{l}
\vdash (\forall a\ a'.\ (\mathtt{PL}\ a = \mathtt{PL}\ a') \iff (a = a')) \land \\
\quad \forall a\ a'.\ (\mathtt{OMNI}\ a = \mathtt{OMNI}\ a') \iff (a = a')
\end{array}$$

[slOutput_distinct_clauses]

$$\begin{array}{l}
\vdash \mathtt{PlanPB} \neq \mathtt{MoveToORP} \land \mathtt{PlanPB} \neq \mathtt{ConductORP} \land \\
\quad \mathtt{PlanPB} \neq \mathtt{MoveToPB} \land \mathtt{PlanPB} \neq \mathtt{ConductPB} \land \\
\quad \mathtt{PlanPB} \neq \mathtt{CompletePB} \land \mathtt{PlanPB} \neq \mathtt{unAuthenticated} \land \\
\quad \mathtt{PlanPB} \neq \mathtt{unAuthorized} \land \mathtt{MoveToORP} \neq \mathtt{ConductORP} \land \\
\quad \mathtt{MoveToORP} \neq \mathtt{MoveToPB} \land \mathtt{MoveToORP} \neq \mathtt{ConductPB} \land \\
\quad \mathtt{MoveToORP} \neq \mathtt{CompletePB} \land \mathtt{MoveToORP} \neq \mathtt{unAuthenticated} \land \\
\quad \mathtt{MoveToORP} \neq \mathtt{unAuthorized} \land \mathtt{ConductORP} \neq \mathtt{MoveToPB} \land \\
\quad \mathtt{ConductORP} \neq \mathtt{ConductPB} \land \mathtt{ConductORP} \neq \mathtt{CompletePB} \land \\
\quad \mathtt{ConductORP} \neq \mathtt{unAuthenticated} \land \mathtt{ConductORP} \neq \mathtt{unAuthorized} \land \\
\quad \mathtt{MoveToPB} \neq \mathtt{ConductPB} \land \mathtt{MoveToPB} \neq \mathtt{CompletePB} \land \\
\quad \mathtt{MoveToPB} \neq \mathtt{unAuthenticated} \land \mathtt{MoveToPB} \neq \mathtt{unAuthorized} \land \\
\quad \mathtt{ConductPB} \neq \mathtt{CompletePB} \land \mathtt{ConductPB} \neq \mathtt{unAuthenticated} \land \\
\quad \mathtt{ConductPB} \neq \mathtt{unAuthorized} \land \mathtt{CompletePB} \neq \mathtt{unAuthenticated} \land \\
\quad \mathtt{CompletePB} \neq \mathtt{unAuthorized} \land \mathtt{unAuthenticated} \neq \mathtt{unAuthorized}
\end{array}$$

[slState_distinct_clauses]

$$\begin{array}{l}
\vdash \mathtt{PLAN\_PB} \neq \mathtt{MOVE\_TO\_ORP} \land \mathtt{PLAN\_PB} \neq \mathtt{CONDUCT\_ORP} \land \\
\quad \mathtt{PLAN\_PB} \neq \mathtt{MOVE\_TO\_PB} \land \mathtt{PLAN\_PB} \neq \mathtt{CONDUCT\_PB} \land \\
\quad \mathtt{PLAN\_PB} \neq \mathtt{COMPLETE\_PB} \land \mathtt{MOVE\_TO\_ORP} \neq \mathtt{CONDUCT\_ORP} \land \\
\quad \mathtt{MOVE\_TO\_ORP} \neq \mathtt{MOVE\_TO\_PB} \land \mathtt{MOVE\_TO\_ORP} \neq \mathtt{CONDUCT\_PB} \land \\
\quad \mathtt{MOVE\_TO\_ORP} \neq \mathtt{COMPLETE\_PB} \land \mathtt{CONDUCT\_ORP} \neq \mathtt{MOVE\_TO\_PB} \land \\
\quad \mathtt{CONDUCT\_ORP} \neq \mathtt{CONDUCT\_PB} \land \mathtt{CONDUCT\_ORP} \neq \mathtt{COMPLETE\_PB} \land \\
\quad \mathtt{MOVE\_TO\_PB} \neq \mathtt{CONDUCT\_PB} \land \mathtt{MOVE\_TO\_PB} \neq \mathtt{COMPLETE\_PB} \land \\
\quad \mathtt{CONDUCT\_PB} \neq \mathtt{COMPLETE\_PB}
\end{array}$$

[stateRole_distinct_clauses]

$$\vdash \mathtt{PlatoonLeader} \neq \mathtt{Omni}$$

2 ssmPBIntegrated Theory

Built: 10 June 2018

Parent Theories: PBIntegratedDef, ssm

2.1 Theorems

[inputOK_def]

$\vdash$ (inputOK (Name PlatoonLeader says prop cmd) $\iff$ T) $\land$
(inputOK (Name Omni says prop cmd) $\iff$ T) $\land$
(inputOK TT $\iff$ F) $\land$ (inputOK FF $\iff$ F) $\land$
(inputOK (prop v) $\iff$ F) $\land$ (inputOK (notf v_1) $\iff$ F) $\land$
(inputOK (v_2 andf v_3) $\iff$ F) $\land$ (inputOK (v_4 orf v_5) $\iff$ F) $\land$
(inputOK (v_6 impf v_7) $\iff$ F) $\land$ (inputOK (v_8 eqf v_9) $\iff$ F) $\land$
(inputOK (v_{10} says TT) $\iff$ F) $\land$ (inputOK (v_{10} says FF) $\iff$ F) $\land$
(inputOK ($v133$ meet $v134$ says prop v_{66}) $\iff$ F) $\land$
(inputOK ($v135$ quoting $v136$ says prop v_{66}) $\iff$ F) $\land$
(inputOK (v_{10} says notf v_{67}) $\iff$ F) $\land$
(inputOK (v_{10} says (v_{68} andf v_{69})) $\iff$ F) $\land$
(inputOK (v_{10} says (v_{70} orf v_{71})) $\iff$ F) $\land$
(inputOK (v_{10} says (v_{72} impf v_{73})) $\iff$ F) $\land$
(inputOK (v_{10} says (v_{74} eqf v_{75})) $\iff$ F) $\land$
(inputOK (v_{10} says v_{76} says v_{77}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{78} speaks_for v_{79}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{80} controls v_{81}) $\iff$ F) $\land$
(inputOK (v_{10} says reps v_{82} v_{83} v_{84}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{85} domi v_{86}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{87} eqi v_{88}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{89} doms v_{90}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{91} eqs v_{92}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{93} eqn v_{94}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{95} lte v_{96}) $\iff$ F) $\land$
(inputOK (v_{10} says v_{97} lt v_{98}) $\iff$ F) $\land$
(inputOK (v_{12} speaks_for v_{13}) $\iff$ F) $\land$
(inputOK (v_{14} controls v_{15}) $\iff$ F) $\land$
(inputOK (reps v_{16} v_{17} v_{18}) $\iff$ F) $\land$
(inputOK (v_{19} domi v_{20}) $\iff$ F) $\land$
(inputOK (v_{21} eqi v_{22}) $\iff$ F) $\land$
(inputOK (v_{23} doms v_{24}) $\iff$ F) $\land$
(inputOK (v_{25} eqs v_{26}) $\iff$ F) $\land$ (inputOK (v_{27} eqn v_{28}) $\iff$ F) $\land$
(inputOK (v_{29} lte v_{30}) $\iff$ F) $\land$ (inputOK (v_{31} lt v_{32}) $\iff$ F)

[inputOK_ind]

$\vdash$ $\forall P$.
($\forall cmd$. P (Name PlatoonLeader says prop cmd)) $\land$
($\forall cmd$. P (Name Omni says prop cmd)) $\land$ P TT $\land$ P FF $\land$
($\forall v$. P (prop v)) $\land$ ($\forall v_1$. P (notf v_1)) $\land$
($\forall v_2\ v_3$. P (v_2 andf v_3)) $\land$ ($\forall v_4\ v_5$. P (v_4 orf v_5)) $\land$
($\forall v_6\ v_7$. P (v_6 impf v_7)) $\land$ ($\forall v_8\ v_9$. P (v_8 eqf v_9)) $\land$
($\forall v_{10}$. P (v_{10} says TT)) $\land$ ($\forall v_{10}$. P (v_{10} says FF)) $\land$
($\forall v133\ v134\ v_{66}$. P ($v133$ meet $v134$ says prop v_{66})) $\land$
($\forall v135\ v136\ v_{66}$. P ($v135$ quoting $v136$ says prop v_{66})) $\land$
($\forall v_{10}\ v_{67}$. P (v_{10} says notf v_{67})) $\land$
($\forall v_{10}\ v_{68}\ v_{69}$. P (v_{10} says (v_{68} andf v_{69}))) $\land$

```
     (∀v10 v70 v71. P (v10 says (v70 orf v71))) ∧
     (∀v10 v72 v73. P (v10 says (v72 impf v73))) ∧
     (∀v10 v74 v75. P (v10 says (v74 eqf v75))) ∧
     (∀v10 v76 v77. P (v10 says v76 says v77)) ∧
     (∀v10 v78 v79. P (v10 says v78 speaks_for v79)) ∧
     (∀v10 v80 v81. P (v10 says v80 controls v81)) ∧
     (∀v10 v82 v83 v84. P (v10 says reps v82 v83 v84)) ∧
     (∀v10 v85 v86. P (v10 says v85 domi v86)) ∧
     (∀v10 v87 v88. P (v10 says v87 eqi v88)) ∧
     (∀v10 v89 v90. P (v10 says v89 doms v90)) ∧
     (∀v10 v91 v92. P (v10 says v91 eqs v92)) ∧
     (∀v10 v93 v94. P (v10 says v93 eqn v94)) ∧
     (∀v10 v95 v96. P (v10 says v95 lte v96)) ∧
     (∀v10 v97 v98. P (v10 says v97 lt v98)) ∧
     (∀v12 v13. P (v12 speaks_for v13)) ∧
     (∀v14 v15. P (v14 controls v15)) ∧
     (∀v16 v17 v18. P (reps v16 v17 v18)) ∧
     (∀v19 v20. P (v19 domi v20)) ∧
     (∀v21 v22. P (v21 eqi v22)) ∧
     (∀v23 v24. P (v23 doms v24)) ∧
     (∀v25 v26. P (v25 eqs v26)) ∧ (∀v27 v28. P (v27 eqn v28)) ∧
     (∀v29 v30. P (v29 lte v30)) ∧ (∀v31 v32. P (v31 lt v32)) ⇒
     ∀v. P v
```

[PBNS_def]

```
 ⊢ (PBNS PLAN_PB (exec [SOME (SLc (PL crossLD))]) =
    MOVE_TO_ORP) ∧
   (PBNS MOVE_TO_ORP (exec [SOME (SLc (PL conductORP))]) =
    CONDUCT_ORP) ∧
   (PBNS CONDUCT_ORP (exec [SOME (SLc (PL moveToPB))]) =
    MOVE_TO_PB) ∧
   (PBNS MOVE_TO_PB (exec [SOME (SLc (PL conductPB))]) =
    CONDUCT_PB) ∧
   (PBNS CONDUCT_PB (exec [SOME (SLc (PL completePB))]) =
    COMPLETE_PB) ∧ (PBNS s (trap v0) = s) ∧
   (PBNS s (discard v1) = s)
```

[PBNS_ind]

```
 ⊢ ∀P.
     P PLAN_PB (exec [SOME (SLc (PL crossLD))]) ∧
     P MOVE_TO_ORP (exec [SOME (SLc (PL conductORP))]) ∧
     P CONDUCT_ORP (exec [SOME (SLc (PL moveToPB))]) ∧
     P MOVE_TO_PB (exec [SOME (SLc (PL conductPB))]) ∧
     P CONDUCT_PB (exec [SOME (SLc (PL completePB))]) ∧
     (∀s v0. P s (trap v0)) ∧ (∀s v1. P s (discard v1)) ∧
     (∀v8. P v8 (exec [])) ∧
     (∀v11 v10. P v11 (exec (NONE::v10))) ∧
     (∀v16 v13 v15. P v16 (exec (SOME (ESCc v13)::v15))) ∧
     P MOVE_TO_ORP (exec [SOME (SLc (PL crossLD))]) ∧
```

P CONDUCT_ORP (exec [SOME (SLc (PL crossLD))]) $\land$
P MOVE_TO_PB (exec [SOME (SLc (PL crossLD))]) $\land$
P CONDUCT_PB (exec [SOME (SLc (PL crossLD))]) $\land$
P COMPLETE_PB (exec [SOME (SLc (PL crossLD))]) $\land$
P PLAN_PB (exec [SOME (SLc (PL conductORP))]) $\land$
P CONDUCT_ORP (exec [SOME (SLc (PL conductORP))]) $\land$
P MOVE_TO_PB (exec [SOME (SLc (PL conductORP))]) $\land$
P CONDUCT_PB (exec [SOME (SLc (PL conductORP))]) $\land$
P COMPLETE_PB (exec [SOME (SLc (PL conductORP))]) $\land$
P PLAN_PB (exec [SOME (SLc (PL moveToPB))]) $\land$
P MOVE_TO_ORP (exec [SOME (SLc (PL moveToPB))]) $\land$
P MOVE_TO_PB (exec [SOME (SLc (PL moveToPB))]) $\land$
P CONDUCT_PB (exec [SOME (SLc (PL moveToPB))]) $\land$
P COMPLETE_PB (exec [SOME (SLc (PL moveToPB))]) $\land$
P PLAN_PB (exec [SOME (SLc (PL conductPB))]) $\land$
P MOVE_TO_ORP (exec [SOME (SLc (PL conductPB))]) $\land$
P CONDUCT_ORP (exec [SOME (SLc (PL conductPB))]) $\land$
P CONDUCT_PB (exec [SOME (SLc (PL conductPB))]) $\land$
P COMPLETE_PB (exec [SOME (SLc (PL conductPB))]) $\land$
P PLAN_PB (exec [SOME (SLc (PL completePB))]) $\land$
P MOVE_TO_ORP (exec [SOME (SLc (PL completePB))]) $\land$
P CONDUCT_ORP (exec [SOME (SLc (PL completePB))]) $\land$
P MOVE_TO_PB (exec [SOME (SLc (PL completePB))]) $\land$
P COMPLETE_PB (exec [SOME (SLc (PL completePB))]) $\land$
($\forall v_{24}$. P v_{24} (exec [SOME (SLc (PL incomplete))])) $\land$
($\forall v_{26}$ v_{25} v_{22} v_{23}.
  P v_{26} (exec (SOME (SLc (PL v_{25}))::v_{22}::v_{23}))) $\land$
($\forall v_{28}$ v_{19} v_{27}. P v_{28} (exec (SOME (SLc (OMNI v_{19}))::v_{27}))) $\Rightarrow$
$\forall v$ v_1. P v v_1

[PBOut_def]

$\vdash$ (PBOut PLAN_PB (exec [SOME (SLc (PL crossLD))]) =
 MoveToORP) $\land$
(PBOut MOVE_TO_ORP (exec [SOME (SLc (PL conductORP))]) =
 ConductORP) $\land$
(PBOut CONDUCT_ORP (exec [SOME (SLc (PL moveToPB))]) =
 MoveToPB) $\land$
(PBOut MOVE_TO_PB (exec [SOME (SLc (PL conductPB))]) =
 ConductPB) $\land$
(PBOut CONDUCT_PB (exec [SOME (SLc (PL completePB))]) =
 CompletePB) $\land$ (PBOut s (trap v_0) = unAuthorized) $\land$
(PBOut s (discard v_1) = unAuthenticated)

[PBOut_ind]

$\vdash \forall P$.
P PLAN_PB (exec [SOME (SLc (PL crossLD))]) $\land$
P MOVE_TO_ORP (exec [SOME (SLc (PL conductORP))]) $\land$
P CONDUCT_ORP (exec [SOME (SLc (PL moveToPB))]) $\land$
P MOVE_TO_PB (exec [SOME (SLc (PL conductPB))]) $\land$

```
     P CONDUCT_PB (exec [SOME (SLc (PL completePB))]) ∧
     (∀s v0. P s (trap v0)) ∧ (∀s v1. P s (discard v1)) ∧
     (∀v8. P v8 (exec [])) ∧
     (∀v11 v10. P v11 (exec (NONE::v10))) ∧
     (∀v16 v13 v15. P v16 (exec (SOME (ESCc v13)::v15))) ∧
     P MOVE_TO_ORP (exec [SOME (SLc (PL crossLD))]) ∧
     P CONDUCT_ORP (exec [SOME (SLc (PL crossLD))]) ∧
     P MOVE_TO_PB (exec [SOME (SLc (PL crossLD))]) ∧
     P CONDUCT_PB (exec [SOME (SLc (PL crossLD))]) ∧
     P COMPLETE_PB (exec [SOME (SLc (PL crossLD))]) ∧
     P PLAN_PB (exec [SOME (SLc (PL conductORP))]) ∧
     P CONDUCT_ORP (exec [SOME (SLc (PL conductORP))]) ∧
     P MOVE_TO_PB (exec [SOME (SLc (PL conductORP))]) ∧
     P CONDUCT_PB (exec [SOME (SLc (PL conductORP))]) ∧
     P COMPLETE_PB (exec [SOME (SLc (PL conductORP))]) ∧
     P PLAN_PB (exec [SOME (SLc (PL moveToPB))]) ∧
     P MOVE_TO_ORP (exec [SOME (SLc (PL moveToPB))]) ∧
     P MOVE_TO_PB (exec [SOME (SLc (PL moveToPB))]) ∧
     P CONDUCT_PB (exec [SOME (SLc (PL moveToPB))]) ∧
     P COMPLETE_PB (exec [SOME (SLc (PL moveToPB))]) ∧
     P PLAN_PB (exec [SOME (SLc (PL conductPB))]) ∧
     P MOVE_TO_ORP (exec [SOME (SLc (PL conductPB))]) ∧
     P CONDUCT_ORP (exec [SOME (SLc (PL conductPB))]) ∧
     P CONDUCT_PB (exec [SOME (SLc (PL conductPB))]) ∧
     P COMPLETE_PB (exec [SOME (SLc (PL conductPB))]) ∧
     P PLAN_PB (exec [SOME (SLc (PL completePB))]) ∧
     P MOVE_TO_ORP (exec [SOME (SLc (PL completePB))]) ∧
     P CONDUCT_ORP (exec [SOME (SLc (PL completePB))]) ∧
     P MOVE_TO_PB (exec [SOME (SLc (PL completePB))]) ∧
     P COMPLETE_PB (exec [SOME (SLc (PL completePB))]) ∧
     (∀v24. P v24 (exec [SOME (SLc (PL incomplete))])) ∧
     (∀v26 v25 v22 v23.
        P v26 (exec (SOME (SLc (PL v25))::v22::v23))) ∧
     (∀v28 v19 v27. P v28 (exec (SOME (SLc (OMNI v19))::v27))) ⇒
     ∀v v1. P v v1
```

[PlatoonLeader_Omni_notDiscard_slCommand_thm]

```
 ⊢ ∀NS Out M Oi Os.
     ¬TR (M,Oi,Os)
        (discard
           [SOME (SLc (PL plCommand));
            SOME (SLc (OMNI omniCommand))])
        (CFG inputOK secContext secAuthorization
           ([Name Omni says prop (SOME (SLc (PL plCommand)));
             Name PlatoonLeader says
             prop (SOME (SLc (OMNI omniCommand)))]::ins) PLAN_PB
           outs)
        (CFG inputOK secContext secAuthorization ins
           (NS PLAN_PB
              (discard
                 [SOME (SLc (PL plCommand));
                  SOME (SLc (OMNI omniCommand))]))
           (Out PLAN_PB
              (discard
                 [SOME (SLc (PL plCommand));
                  SOME (SLc (OMNI omniCommand))])::outs))
```

[PlatoonLeader_PLAN_PB_exec_lemma]

```
 ⊢ ∀M Oi Os.
     CFGInterpret (M,Oi,Os)
       (CFG inputOK secContext secAuthorization
          ([Name Omni says
            prop (SOME (SLc (OMNI ssmPlanPBComplete)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
          outs) ⇒
     (M,Oi,Os) satList
     propCommandList
       [Name Omni says
        prop (SOME (SLc (OMNI ssmPlanPBComplete)));
        Name PlatoonLeader says prop (SOME (SLc (PL crossLD)))]
```

[PlatoonLeader_PLAN_PB_trap_justified_lemma]

```
 ⊢ omniCommand ≠ ssmPlanPBComplete ⇒
   (s = PLAN_PB) ⇒
   ∀NS Out M Oi Os.
     TR (M,Oi,Os)
       (trap
          (inputList
             [Name Omni says
              prop (SOME (SLc (OMNI omniCommand)));
              Name PlatoonLeader says
              prop (SOME (SLc (PL crossLD)))]))
       (CFG inputOK secContext secAuthorization
          ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB outs)
       (CFG inputOK secContext secAuthorization ins
          (NS PLAN_PB
             (trap
                (inputList
                   [Name Omni says
                    prop (SOME (SLc (OMNI omniCommand)));
                    Name PlatoonLeader says
                    prop (SOME (SLc (PL crossLD)))])))
          (Out PLAN_PB
             (trap
                (inputList
                    [Name Omni says
                     prop (SOME (SLc (OMNI omniCommand)));
                     Name PlatoonLeader says
                     prop (SOME (SLc (PL crossLD)))]])::outs)) ⟺
      authenticationTest inputOK
        [Name Omni says prop (SOME (SLc (OMNI omniCommand)));
         Name PlatoonLeader says
         prop (SOME (SLc (PL crossLD)))] ∧
      CFGInterpret (M,Oi,Os)
        (CFG inputOK secContext secAuthorization
           ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
             Name PlatoonLeader says
             prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
           outs) ∧ (M,Oi,Os) sat prop NONE
```

[PlatoonLeader_PLAN_PB_trap_justified_thm]

```
 ⊢ omniCommand ≠ ssmPlanPBComplete ⇒
   (s = PLAN_PB) ⇒
   ∀ NS Out M Oi Os.
     TR (M,Oi,Os)
       (trap
          [SOME (SLc (OMNI omniCommand));
           SOME (SLc (PL crossLD))])
       (CFG inputOK secContext secAuthorization
          ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB outs)
       (CFG inputOK secContext secAuthorization ins
          (NS PLAN_PB
             (trap
                [SOME (SLc (OMNI omniCommand));
                 SOME (SLc (PL crossLD))]))
          (Out PLAN_PB
             (trap
                [SOME (SLc (OMNI omniCommand));
                 SOME (SLc (PL crossLD))])::outs)) ⟺
     authenticationTest inputOK
       [Name Omni says prop (SOME (SLc (OMNI omniCommand)));
        Name PlatoonLeader says
        prop (SOME (SLc (PL crossLD)))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputOK secContext secAuthorization
          ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
          outs) ∧ (M,Oi,Os) sat prop NONE
```

[PlatoonLeader_PLAN_PB_trap_lemma]

```
 ⊢ omniCommand ≠ ssmPlanPBComplete ⇒
   (s = PLAN_PB) ⇒
```

```
∀M  Oi  Os.
  CFGInterpret (M,Oi,Os)
    (CFG inputOK secContext secAuthorization
       ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
         Name PlatoonLeader says
         prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
       outs) ⇒
  (M,Oi,Os) sat prop NONE
```

3 PBIntegratedDef Theory

Built: 10 June 2018

Parent Theories: PBTypeIntegrated, aclfoundation

3.1 Definitions

[secAuthorization_def]

```
 ⊢ ∀xs. secAuthorization xs = secHelper (getOmniCommand xs)
```

[secContext_def]

```
 ⊢ (∀xs.
       secContext PLAN_PB xs =
       if getOmniCommand xs = ssmPlanPBComplete then
         [prop (SOME (SLc (OMNI ssmPlanPBComplete))) impf
          Name PlatoonLeader controls
          prop (SOME (SLc (PL crossLD)))]
       else [prop NONE]) ∧
   (∀xs.
       secContext MOVE_TO_ORP xs =
       if getOmniCommand xs = ssmMoveToORPComplete then
         [prop (SOME (SLc (OMNI ssmMoveToORPComplete))) impf
          Name PlatoonLeader controls
          prop (SOME (SLc (PL conductORP)))]
       else [prop NONE]) ∧
   (∀xs.
       secContext CONDUCT_ORP xs =
       if getOmniCommand xs = ssmConductORPComplete then
         [prop (SOME (SLc (OMNI ssmConductORPComplete))) impf
          Name PlatoonLeader controls
          prop (SOME (SLc (PL moveToPB)))]
       else [prop NONE]) ∧
   (∀xs.
       secContext MOVE_TO_PB xs =
       if getOmniCommand xs = ssmConductORPComplete then
         [prop (SOME (SLc (OMNI ssmMoveToPBComplete))) impf
          Name PlatoonLeader controls
          prop (SOME (SLc (PL conductPB)))]
       else [prop NONE]) ∧
    ∀xs.
      secContext CONDUCT_PB xs =
      if getOmniCommand xs = ssmConductPBComplete then
        [prop (SOME (SLc (OMNI ssmConductPBComplete))) impf
         Name PlatoonLeader controls
         prop (SOME (SLc (PL completePB)))]
      else [prop NONE]
```

[secHelper_def]

```
 ⊢ ∀cmd.
      secHelper cmd =
      [Name Omni controls prop (SOME (SLc (OMNI cmd)))]
```

3.2 Theorems

[getOmniCommand_def]

```
 ⊢ (getOmniCommand [] = invalidOmniCommand) ∧
   (∀xs cmd.
      getOmniCommand
        (Name Omni says prop (SOME (SLc (OMNI cmd)))::xs) =
      cmd) ∧
   (∀xs. getOmniCommand (TT::xs) = getOmniCommand xs) ∧
   (∀xs. getOmniCommand (FF::xs) = getOmniCommand xs) ∧
   (∀xs v2. getOmniCommand (prop v2::xs) = getOmniCommand xs) ∧
   (∀xs v3. getOmniCommand (notf v3::xs) = getOmniCommand xs) ∧
   (∀xs v5 v4.
      getOmniCommand (v4 andf v5::xs) = getOmniCommand xs) ∧
   (∀xs v7 v6.
      getOmniCommand (v6 orf v7::xs) = getOmniCommand xs) ∧
   (∀xs v9 v8.
      getOmniCommand (v8 impf v9::xs) = getOmniCommand xs) ∧
   (∀xs v11 v10.
      getOmniCommand (v10 eqf v11::xs) = getOmniCommand xs) ∧
   (∀xs v12.
      getOmniCommand (v12 says TT::xs) = getOmniCommand xs) ∧
   (∀xs v12.
      getOmniCommand (v12 says FF::xs) = getOmniCommand xs) ∧
   (∀xs v134.
      getOmniCommand (Name v134 says prop NONE::xs) =
      getOmniCommand xs) ∧
   (∀xs v144.
      getOmniCommand
        (Name PlatoonLeader says prop (SOME v144)::xs) =
      getOmniCommand xs) ∧
   (∀xs v146.
      getOmniCommand
        (Name Omni says prop (SOME (ESCc v146))::xs) =
     getOmniCommand xs) ∧
  (∀xs v150.
     getOmniCommand
       (Name Omni says prop (SOME (SLc (PL v150)))::xs) =
     getOmniCommand xs) ∧
  (∀xs v68 v136 v135.
     getOmniCommand (v135 meet v136 says prop v68::xs) =
     getOmniCommand xs) ∧
  (∀xs v68 v138 v137.
     getOmniCommand (v137 quoting v138 says prop v68::xs) =
     getOmniCommand xs) ∧
  (∀xs v69 v12.
     getOmniCommand (v12 says notf v69::xs) =
     getOmniCommand xs) ∧
  (∀xs v71 v70 v12.
     getOmniCommand (v12 says (v70 andf v71)::xs) =
     getOmniCommand xs) ∧
  (∀xs v73 v72 v12.
     getOmniCommand (v12 says (v72 orf v73)::xs) =
     getOmniCommand xs) ∧
  (∀xs v75 v74 v12.
     getOmniCommand (v12 says (v74 impf v75)::xs) =
     getOmniCommand xs) ∧
  (∀xs v77 v76 v12.
     getOmniCommand (v12 says (v76 eqf v77)::xs) =
     getOmniCommand xs) ∧
  (∀xs v79 v78 v12.
     getOmniCommand (v12 says v78 says v79::xs) =
     getOmniCommand xs) ∧
  (∀xs v81 v80 v12.
     getOmniCommand (v12 says v80 speaks_for v81::xs) =
     getOmniCommand xs) ∧
  (∀xs v83 v82 v12.
     getOmniCommand (v12 says v82 controls v83::xs) =
     getOmniCommand xs) ∧
  (∀xs v86 v85 v84 v12.
     getOmniCommand (v12 says reps v84 v85 v86::xs) =
     getOmniCommand xs) ∧
  (∀xs v88 v87 v12.
     getOmniCommand (v12 says v87 domi v88::xs) =
     getOmniCommand xs) ∧
  (∀xs v90 v89 v12.
     getOmniCommand (v12 says v89 eqi v90::xs) =
     getOmniCommand xs) ∧
  (∀xs v92 v91 v12.
     getOmniCommand (v12 says v91 doms v92::xs) =
     getOmniCommand xs) ∧
  (∀xs v94 v93 v12.
     getOmniCommand (v12 says v93 eqs v94::xs) =
     getOmniCommand xs) ∧
  (∀ xs v96 v95 v12.
     getOmniCommand (v12 says v95 eqn v96::xs) =
     getOmniCommand xs) ∧
  (∀ xs v98 v97 v12.
     getOmniCommand (v12 says v97 lte v98::xs) =
     getOmniCommand xs) ∧
  (∀ xs v99 v12 v100.
     getOmniCommand (v12 says v99 lt v100::xs) =
     getOmniCommand xs) ∧
  (∀ xs v15 v14.
     getOmniCommand (v14 speaks_for v15::xs) =
     getOmniCommand xs) ∧
  (∀ xs v17 v16.
     getOmniCommand (v16 controls v17::xs) =
     getOmniCommand xs) ∧
  (∀ xs v20 v19 v18.
     getOmniCommand (reps v18 v19 v20::xs) =
     getOmniCommand xs) ∧
  (∀ xs v22 v21.
     getOmniCommand (v21 domi v22::xs) = getOmniCommand xs) ∧
  (∀ xs v24 v23.
     getOmniCommand (v23 eqi v24::xs) = getOmniCommand xs) ∧
  (∀ xs v26 v25.
     getOmniCommand (v25 doms v26::xs) = getOmniCommand xs) ∧
  (∀ xs v28 v27.
     getOmniCommand (v27 eqs v28::xs) = getOmniCommand xs) ∧
  (∀ xs v30 v29.
     getOmniCommand (v29 eqn v30::xs) = getOmniCommand xs) ∧
  (∀ xs v32 v31.
     getOmniCommand (v31 lte v32::xs) = getOmniCommand xs) ∧
  ∀ xs v34 v33.
    getOmniCommand (v33 lt v34::xs) = getOmniCommand xs
```

[getOmniCommand_ind]

```
 ⊢ ∀ P.
     P [] ∧
     (∀ cmd xs.
        P (Name Omni says prop (SOME (SLc (OMNI cmd)))::xs)) ∧
     (∀ xs. P xs ⇒ P (TT::xs)) ∧ (∀ xs. P xs ⇒ P (FF::xs)) ∧
     (∀ v2 xs. P xs ⇒ P (prop v2::xs)) ∧
     (∀ v3 xs. P xs ⇒ P (notf v3::xs)) ∧
     (∀ v4 v5 xs. P xs ⇒ P (v4 andf v5::xs)) ∧
     (∀ v6 v7 xs. P xs ⇒ P (v6 orf v7::xs)) ∧
     (∀ v8 v9 xs. P xs ⇒ P (v8 impf v9::xs)) ∧
     (∀ v10 v11 xs. P xs ⇒ P (v10 eqf v11::xs)) ∧
     (∀ v12 xs. P xs ⇒ P (v12 says TT::xs)) ∧
     (∀ v12 xs. P xs ⇒ P (v12 says FF::xs)) ∧
     (∀ v134 xs. P xs ⇒ P (Name v134 says prop NONE::xs)) ∧
```

```
(∀ v144 xs.
   P xs ⇒
   P (Name PlatoonLeader says prop (SOME v144)::xs)) ∧
(∀ v146 xs.
   P xs ⇒ P (Name Omni says prop (SOME (ESCc v146))::xs)) ∧
(∀ v150 xs.
   P xs ⇒
   P (Name Omni says prop (SOME (SLc (PL v150)))::xs)) ∧
(∀ v135 v136 v68 xs.
   P xs ⇒ P (v135 meet v136 says prop v68::xs)) ∧
(∀ v137 v138 v68 xs.
   P xs ⇒ P (v137 quoting v138 says prop v68::xs)) ∧
(∀ v12 v69 xs. P xs ⇒ P (v12 says notf v69::xs)) ∧
(∀ v12 v70 v71 xs. P xs ⇒ P (v12 says (v70 andf v71)::xs)) ∧
(∀ v12 v72 v73 xs. P xs ⇒ P (v12 says (v72 orf v73)::xs)) ∧
(∀ v12 v74 v75 xs. P xs ⇒ P (v12 says (v74 impf v75)::xs)) ∧
(∀ v12 v76 v77 xs. P xs ⇒ P (v12 says (v76 eqf v77)::xs)) ∧
(∀ v12 v78 v79 xs. P xs ⇒ P (v12 says v78 says v79::xs)) ∧
(∀ v12 v80 v81 xs.
   P xs ⇒ P (v12 says v80 speaks_for v81::xs)) ∧
(∀ v12 v82 v83 xs.
   P xs ⇒ P (v12 says v82 controls v83::xs)) ∧
(∀ v12 v84 v85 v86 xs.
   P xs ⇒ P (v12 says reps v84 v85 v86::xs)) ∧
(∀ v12 v87 v88 xs. P xs ⇒ P (v12 says v87 domi v88::xs)) ∧
(∀ v12 v89 v90 xs. P xs ⇒ P (v12 says v89 eqi v90::xs)) ∧
(∀ v12 v91 v92 xs. P xs ⇒ P (v12 says v91 doms v92::xs)) ∧
(∀ v12 v93 v94 xs. P xs ⇒ P (v12 says v93 eqs v94::xs)) ∧
(∀ v12 v95 v96 xs. P xs ⇒ P (v12 says v95 eqn v96::xs)) ∧
(∀ v12 v97 v98 xs. P xs ⇒ P (v12 says v97 lte v98::xs)) ∧
(∀ v12 v99 v100 xs. P xs ⇒ P (v12 says v99 lt v100::xs)) ∧
(∀ v14 v15 xs. P xs ⇒ P (v14 speaks_for v15::xs)) ∧
(∀ v16 v17 xs. P xs ⇒ P (v16 controls v17::xs)) ∧
(∀ v18 v19 v20 xs. P xs ⇒ P (reps v18 v19 v20::xs)) ∧
(∀ v21 v22 xs. P xs ⇒ P (v21 domi v22::xs)) ∧
(∀ v23 v24 xs. P xs ⇒ P (v23 eqi v24::xs)) ∧
(∀ v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
(∀ v27 v28 xs. P xs ⇒ P (v27 eqs v28::xs)) ∧
(∀ v29 v30 xs. P xs ⇒ P (v29 eqn v30::xs)) ∧
(∀ v31 v32 xs. P xs ⇒ P (v31 lte v32::xs)) ∧
(∀ v33 v34 xs. P xs ⇒ P (v33 lt v34::xs)) ⇒
∀ v. P v
```

Index

PBIntegratedDef Theory, 11
- Definitions, 11
  - secAuthorization_def, 11
  - secContext_def, 11
  - secHelper_def, 12
- Theorems, 12
  - getOmniCommand_def, 12
  - getOmniCommand_ind, 14

PBTypeIntegrated Theory, 3
- Datatypes, 3
- Theorems, 3
  - omniCommand_distinct_clauses, 3
  - plCommand_distinct_clauses, 3
  - slCommand_distinct_clauses, 4
  - slCommand_one_one, 4
  - slOutput_distinct_clauses, 4
  - slState_distinct_clauses, 4
  - stateRole_distinct_clauses, 4

ssmPBIntegrated Theory, 4
- Theorems, 5
  - inputOK_def, 5
  - inputOK_ind, 5
  - PBNS_def, 6
  - PBNS_ind, 6
  - PBOut_def, 7
  - PBOut_ind, 7
  - PlatoonLeader_Omni_notDiscard_slCommand_thm, 8
  - PlatoonLeader_PLAN_PB_exec_lemma, 9
  - PlatoonLeader_PLAN_PB_trap_justified_lemma, 9
  - PlatoonLeader_PLAN_PB_trap_justified_thm, 10
  - PlatoonLeader_PLAN_PB_trap_lemma, 10