Common Policies Expressed in the Logic

1 Policies from Cornell's Institute for Digital Collections

These policies resulted from Sandy Payette's interview with Anne Kenney and Peter Hirtle, the Co-Directors of Cornell's Institute for Digital Collections. The discussion focused on what policies are of particular interest to the digital library community.

For ease of exposition, the policies are not always written in the correct form; namely $\forall x_1, \ldots, x_n(f => g)$ where f is quantifier-free and g is the, possibly negated, **Permitted** predicate. For example, I write $\forall x_1, \ldots, x_n A \Rightarrow B \wedge C$ instead of $(\forall x_1, \ldots, x_n(A \Rightarrow B)) \wedge (\forall x_1, \ldots, x_n(A \Rightarrow C))$. Similarly, I will occasionally write $(A_1 \Rightarrow B) \wedge \ldots \wedge (A_m \Rightarrow B) \wedge (B \Rightarrow C)$ instead of $(A_1 \Rightarrow C) \wedge \ldots \wedge (A_m \Rightarrow C)$. In some formulas universal quantification is 'pushed in' for clarity. In all of these cases, however, the necessary re-writing to get the correct policy form is straight-forward.

1.1

Policy Expressed in English
Object: Course Reserve Packet (CRP) contains several scanned articles
Subject: course members
Actions: read all articles

Subject: professor
Actions: read or delete articles, delete CRP

Subject: all others
Actions: no access

Policy Expressed in the Logic
∀ p ∈ Principals
  ¬Professor(p) ∧ Course_member(p) ⇒
    Permitted(p, read CRP articles) ∧ ¬Permitted(p, delete articles) ∧ ¬Permitted(p, delete CRP)
∧ ∀ p ∈ Principals
  Professor(p) ⇒
    Permitted(p, read CRP articles) ∧ Permitted(p, delete articles) ∧ Permitted(p, delete CRP)
∧ ∀ p ∈ Principals
  ¬Professor(p) ∧ ¬Course_member(p) ⇒
    ¬Permitted(p, read CRP articles) ∧ ¬Permitted(p, delete articles) ∧ ¬Permitted(p, delete CRP)

If we assume that all actions which are not explicitly permitted are implicitly denied, then we can simplify the formula to:

∀ p ∈ Principals
  ¬Professor(p) ∧ Course_member(p) ⇒
    Permitted(p, read CRP articles)
∧ ∀ p ∈ Principals
  Professor(p) ⇒

Permitted(p, read CRP articles) $\wedge$ Permitted(p, delete articles) $\wedge$ Permitted(p, delete CRP) If we assume that a principal can only belong to one group, then the formula can be further simplified to:

$\forall$ p $\in$ Principals
Course_member(p) $\Rightarrow$
Permitted(p, read CRP articles)
$\wedge$ $\forall$ p $\in$ Principals
Professor(p) $\Rightarrow$
Permitted(p, read CRP articles) $\wedge$ Permitted(p, delete articles) $\wedge$ Permitted(p, delete CRP)

If, however, the professor is a course member, then the last formula is not consistent, because the professor is both permitted and not permitted to do deletions.

These simplifying assumptions could be made with the rest of the policies, but for brevity, I'll just give the general forms.

1.2

Policy Expressed in English
Object: CRP contains syllabus, multiple articles, multiple reports
Subject: course members
Actions: read syllabus, articles, and reports

Subject: professor
Actions: read, add, or delete articles and reports, delete CRP

Subject: all others
Actions: read syllabus

Policy Expressed in the Logic
$\forall$ p $\in$ Principals
$\neg$Professor(p) $\wedge$ Course_member(p) $\Rightarrow$
Permitted(p, read CRP syllabus) $\wedge$ Permitted(p, read CRP articles) $\wedge$
Permitted(p, read CRP reports) $\wedge$ $\neg$Permitted(p, add articles) $\wedge$
$\neg$Permitted(p, add reports) $\wedge$ $\neg$Permitted(p, delete articles) $\wedge$
$\neg$Permitted(p, delete reports) $\wedge$ $\neg$Permitted(p, delete CRP)
$\wedge$
$\forall$ p $\in$ Principals
Professor(p) $\Rightarrow$
Permitted(p, read CRP syllabus) $\wedge$ Permitted(p, read CRP articles) $\wedge$
Permitted(p, read CRP reports) $\wedge$ Permitted(p, add articles) $\wedge$
Permitted(p, add reports) $\wedge$ Permitted(p, delete articles) $\wedge$
Permitted(p, delete reports) $\wedge$ Permitted(p, delete CRP)
$\wedge$
$\forall$ p $\in$ Principals
$\neg$Professor(p) $\wedge$ $\neg$Course_member(p) $\Rightarrow$
Permitted(p, read CRP syllabus) $\wedge$ $\neg$Permitted(p, read CRP articles) $\wedge$
$\neg$Permitted(p, read CRP reports) $\wedge$ $\neg$Permitted(p, add articles) $\wedge$
$\neg$Permitted(p, add reports) $\wedge$ $\neg$Permitted(p, delete articles) $\wedge$
$\neg$Permitted(p, delete reports) $\wedge$ $\neg$Permitted(p, delete CRP)

1.3

Policy Expressed in English

Object: Book - bib record and scanned images

Subject: users who have not purchased the rights to access the book

Actions:

- read bib record and table of contents
- pay fee #1 for read (download/print) of low resolution page images
- pay fee #2 for read (download/print) of high resolution page images
- pay fee #3 for read (download/print) of whole book of high resolution page images
- pay fee #4 for read (download/print) of book updates

Subject: users who have purchased right to access the book

Actions:

- read (download/print) whole book in high or low resolution until expiration date
- read updates until expiration date

Policy Expressed in the Logic

∀ p ∈ Principals
  ∀ time t
    ¬Purchased_book_access(p) ∧ ¬Paid_fee(p)⇒
      Permitted(p, read bib record) ∧ Permitted(p, read table of contents) ∧
      Permitted(p, pay fee #1) ∧ Permitted(p, pay fee #2) ∧
      Permitted(p, pay fee #3) ∧ Permitted(p, pay fee #4) ∧
      ¬Permitted(p, read(low, page), t) ∧ ¬Permitted(p, read(high, page), t) ∧
      ¬Permitted(p, read(high, book), t) ∧ ¬Permitted(p, read(low, book), t) ∧
      ¬Permitted(p, read(updates), t)
∧
  Did(p, pay fee #1) ⇒ Paid_ fee_#1(p)
∧
  Did(p, pay fee #2) ⇒ Paid_ fee_#2(p)
∧
  Did(p, pay fee #3) ⇒ Paid_ fee_#3(p)
∧
  Did(p, pay fee #4) ⇒ Paid_ fee_#4(p)
∧
  ∀ time t
    Paid_ fee_#1(p) ⇒
      Paid_fee(p) ∧ Permitted(p, read bib record) ∧
      Permitted(p, read table of contents) ∧ ¬Permitted(p, pay fee #1) ∧
      Permitted(p, pay fee #2) ∧ Permitted(p, pay fee #3) ∧
      Permitted(p, pay fee #4) ∧ Permitted(p, read(low, page), t) ∧
      ¬Permitted(p, read(high, page, t)) ∧ ¬Permitted(p, read(high, book), t) ∧
      ¬Permitted(p, read(low, book), t) ∧ ¬Permitted(p, read(updates), t)
∧
  ∀ time t
    Paid_ fee_#2(p) ⇒
      Paid_fee(p) ∧ Permitted(p, read bib record) ∧
      Permitted(p, read table of contents) ∧ Permitted(p, pay fee #1) ∧
      ¬Permitted(p, pay fee #2) ∧ Permitted(p, pay fee #3) ∧
      Permitted(p, pay fee #4) ∧ ¬Permitted(p, read(low, page), t) ∧
      Permitted(p, read(high, page), t) ∧ ¬Permitted(p, read(high, book), t) ∧
      ¬Permitted(p, read(low, book), t) ∧ ¬Permitted(p, read(updates), t)
∧

$\forall$ time t
  Paid_ fee_#3(p) $\Rightarrow$
    Paid_fee(p) $\wedge$ Permitted(p, read bib record) $\wedge$
    Permitted(p, read table of contents) $\wedge$ Permitted(p, pay fee #1) $\wedge$
    Permitted(p, pay fee #2) $\wedge$ $\neg$Permitted(p, pay fee #3) $\wedge$
    Permitted(p, pay fee #4) $\wedge$ $\neg$Permitted(p, read(low, page), t) $\wedge$
    $\neg$Permitted(p, read(high, page, t)) $\wedge$ Permitted(p, read(high, book), t) $\wedge$
    $\neg$Permitted(p, read(low, book), t) $\wedge$ $\neg$Permitted(p, read(updates), t)

$\wedge$

$\forall$ time t
  Paid_ fee_#4(p) $\Rightarrow$
    Paid_fee(p) $\wedge$ Permitted(p, read bib record) $\wedge$
    Permitted(p, read table of contents) $\wedge$ Permitted(p, pay fee #1) $\wedge$
    Permitted(p, pay fee #2) $\wedge$ Permitted(p, pay fee #3) $\wedge$
    $\neg$Permitted(p, pay fee #4) $\wedge$ $\neg$Permitted(p, read(low, page), t) $\wedge$
    $\neg$Permitted(p, read(high, page), t) $\wedge$ $\neg$Permitted(p, read(high, book), t) $\wedge$
    $\neg$Permitted(p, read(low, book), t) $\wedge$ Permitted(p, read(updates), t)

$\wedge$

Purchased_book_access(p) $\Rightarrow$
  $\forall$ possible expiration times t_{exp}
    Expires(t_{exp}(p) $\Rightarrow$
      $\forall$ $t < t_{exp}$
        Permitted(p, read(high, book), t) $\wedge$ Permitted(p, read(low, book), t) $\wedge$
        Permitted(p, read(updates), t)
      $\wedge$
      $\forall$ $t \geq t_{exp}$
        $\neg$Permitted(p, read(high, book), t) $\wedge$ $\neg$Permitted(read, low, t) $\wedge$
        $\neg$Permitted(p, read(updates), t)) $\wedge$ Permitted(p, read bib record) $\wedge$
        Permitted(p, read table of contents) $\wedge$ Permitted(p, pay fee #1) $\wedge$
        Permitted(p, pay fee #2) $\wedge$ Permitted(p, pay fee #3) $\wedge$
        Permitted(p, pay fee #4) $\wedge$ $\neg$Permitted(p, read(low, page), t) $\wedge$
        $\neg$Permitted(p, read(high, page), t)

Note: The Did operator could be used in place of the paid fee memberships, but the LHS of implications would need to explicitly state if payment was not done. Otherwise, the formula would be inconsistent.

1.4

Policy Expressed in English
Object: a remotely-stored article
Subject: members of Cornell community (CC), identified by IP address
Actions: read (download/print) whole article

Subject: all others
Actions: none

Policy Expressed in the Logic
$\forall$ p $\in$ Principals
  Said(IP address identifier, CC(p)) $\Rightarrow$ CC(p) $\wedge$
  CC(p) $\Rightarrow$ Permitted(p, read article) $\wedge$
  $\neg$CC(p) $\Rightarrow$ $\neg$Permitted(p, read article)

1.5

Policy Expressed in English
Object: an article in an inter-library loan agreement
Subject: CC
Actions:
read until CC has made X accesses to the article in Y years
read until CC has made X1 accesses to the article's parent journal
pay copyright fee to read article if CC has lost read right

Subject: all others
Actions: pay copyright fee to read article

Policy Expressed in the Logic
$\forall$ times $t_1, ..., t_{X+1}$
$\forall \{p_1, ..., p_{X+1}\} \subseteq$ Principals
$CC(p_1) \wedge CC(p_2) \wedge ... \wedge CC(p_{X+1}) \wedge$
$Did(p_1$, access article, $t_1) \wedge \neg Did(p_1$, paid for access, $t_1) \wedge$
$Did(p_2$, access article, $t_2) \wedge \neg Did(p_2$, paid for access, $t_2)$
$\wedge ... \wedge$
$Did(p_{X+1}$, access article, $t_{X+1}) \wedge \neg Did(p_{X+1}$, paid for access, $t_{X+1}) \Rightarrow$
$t_1 = t_2 \vee t_1 = t_3 \vee ... \vee t_{X+1} = t_X$ {2 times must be equal} $\vee$
$|t_1 - t_2| > Y \vee |t_1 - t_3| > Y \vee ... \vee |t_{X+1} - t_X| > Y$ {2 times are more than Y units apart}
$\wedge$
$\forall$ times $t_1, ..., t_{X1+1}$
$\forall \{p_1, ..., p_{X1+1}\} \subseteq$ Principals
$CC(p_1) \wedge CC(p_2) \wedge ... \wedge CC(p_{X1+1}) \wedge$
$Did(p_1$, access parent, $t_1) \wedge \neg Did(p_1$, paid for access, $t_1) \wedge$
$Did(p_2$, access parent, $t_2) \wedge \neg Did(p_2$, paid for access, $t_2)$
$\wedge ... \wedge$
$Did(p_{X1+1}$, access parent, $t_{X+1}) \wedge \neg Did(p_{X1+1}$, paid for access, $t_{X1+1}) \Rightarrow$
$t_1 = t_2 \vee t_1 = t_3 \vee ... \vee t_{X1+1} = t_{X1}$ {2 times must be equal}
$\wedge$
$\forall$ Principals p, $\forall$ times t, Permitted(p, pay for access, t)
$\wedge$
$\forall$ Principals p, $\forall$ times $t1 \leq t2$, Did(p, pay for access, t1) $\Rightarrow$ Permitted(p, access article, t2)

Note: The policy expression assumes that only one access can occur at any time t. A better assumption would be that a particular principal can only access the article once at any time t, but this would require a longer formula. Since the change is fairly obvious and requires more typing, I won't write it up. I'm also assuming that if p is a member of CC and p did action a, then p did the action when p was a member. This assumption could be removed by adding an interval in the Member predicate and requiring that the 'Did's occured during the interval of membership. I have also allowed CC members to pay for access when it's already allowed. This is a minor change from the specification, but it does make the formula much shorter. To restrict payment to only be allowed if free access isn't, we could forbid paying when X1 accesses haven't been made to the parent (more than 2 times are equal) and more than t accesses haven't been made to the article in the last Y years (either fewer than x+1 accesses in last Y yrs or at least 2 occured at the same time.) Paying would only be allowed if the principal was not a member of CC or if either X1 access have been made to the parent or X access have been made within Y years of the request.

1.6

Policy Expressed in English
Object: slide collection (donated from dubious origin)

Subject: CC
Actions:
read access for 3 yrs (while copyright is being secured)
after 3 yrs., read access to slides that have CUL copyright clearence
after 3 yrs., read access to slides that are in the public domain
after 3 yrs., pay fee to read slides with CUL NO copyright clearence

Subject: members of the Yale community
Actions:
no access for first 3 yrs.
after 3 yrs., read access to slides with CUL obtained 'special' copyright clearence
after 3 yrs., read access to slides that are in the public domain
after 3 yrs., pay fee for read access to all other slides

Subject: all others
Actions: read access to public domain slides after 3 yrs.

Policy Expressed in the Logic
Given a constant t_3 provided by the application:
$\forall$ p $\in$ Principals
CC(p) $\Rightarrow$
$\forall$ time t $< t_3$
Permitted(p, read(slide), t)
$\wedge$
$\forall$ time $t' \geq t \geq t_3$
Permitted(p, read(CUL special), t) $\wedge$ Permitted(p, read(CUL not special), t) $\wedge$
Permitted(p, read(public), t) $\wedge$ Permitted(p, pay(CUL NO), t) $\wedge$
(Did(p, pay(CUL NO), t) $\Rightarrow$ Permitted(p, read(CUL NO), t'))
$\wedge\forall$ p $\in$ Principals
Yale(p) $\Rightarrow$
$\forall$ time t $< t_3$
$\neg$Permitted(p, read(slide), t)
$\wedge$
$\forall$ time $t' \geq t \geq t_3$
Permitted(p, read(CUL special), t) $\wedge$ Permitted(p, read(public), t) $\wedge$
Permitted(p, pay(CUL NO),t) $\wedge$ Permitted(p, pay(CUL not special), t) $\wedge$
(Did(p, pay(CUL NO), t) $\Rightarrow$ Permitted(p, read(CUL NO), t')) $\wedge$
(Did(p, pay(CUL not special), t) $\Rightarrow$ Permitted(p, read(CUL not special), t'))
$\wedge\forall$ p $\in$ Principals
$\neg$CC(p) $\wedge$ $\neg$Yale(p) $\Rightarrow$
$\forall$ time t $\geq t_3$
Permitted(p, read(public), t) $\wedge$ $\neg$Permitted(p, read(CUL), t) $\wedge$ $\neg$Permitted(p, read(CUL NO), t)

1.7

Policy Expressed in English
Object: film clip
Subject: all users
Actions: access (download/view) first 10 seconds, pay fee to access entire clip

Policy Expressed in the Logic
$\forall$ p $\in$ Principals
$\neg$Paid_for_ clip(p) $\Rightarrow$

Permitted(p, d/v(0, 10)) $\wedge$ $\neg$Permitted(p, d/v(0, end)) $\wedge$ Permitted(p, pay for clip)
$\wedge$
Did(p, pay for clip) $\Rightarrow$ Paid_for_ clip(p)
$\wedge\forall$ p $\in$ Principals
Paid_for_ clip(p) $\Rightarrow$
Permitted(p, d/v(0, 10)) $\wedge$ Permitted(p, d/v(0, end)) $\wedge$ $\neg$Permitted(p, pay for clip)

1.8

Policy Expressed in English
Object: book with copyright of text belonging to person A and copyright of images to person B
Subject: all users
Actions:
pay copyright to view text until A's rights expire
pay copyright to view images until B's rights expire
read text and view images after relevant rights have expired

Policy Expressed in the Logic
Given a constants t^A_{Exp} and t^B_{Exp} provided by the application:
$\forall$ p $\in$ Principals
$\forall$ t $> t^A_{Exp}$ Permitted(p, read text, t) $\wedge$
$\forall$ t $> t^B_{Exp}$ Permitted(p, view image, t)) $\wedge$
$\forall$ t $\leq$ t2 $\leq t^A_{Exp}$
Permitted(p, pay to read text, t) $\wedge$
(Did(p, pay to read text, t) $\Rightarrow$ Paid_to_read_text(p, t2))$\wedge$
Paid_to_read_text(p, t2) $\Rightarrow$ Permitted(p, read text, t2)
$\neg$Paid_to_read_text(p, t2) $\Rightarrow$ $\neg$Permitted(p, read text, t2)
$\forall$ t $\leq$ t2 $\leq t^B_{Exp}$
Permitted(p, pay to view image, t) $\wedge$
(Did(p, pay to view image, t) $\Rightarrow$ Paid_to_view_image(p, t2))$\wedge$
Paid_to_view_image(p, t2) $\Rightarrow$ Permitted(p, view image, t2)
$\neg$Paid_to_view_image(p, t2) $\Rightarrow$ $\neg$Permitted(p, view image, t2)

2 Policies from Library of Congress

The policies in this section were taken from William Yeo Arm's paper, "Implementing Policies for Access Management" which was published in D-Lib magazine, Feb. 1998. Some of the policies given in the paper are not expressed here, because they are directly analogues to the ones given above.

2.1

Policy Expressed in English
(adapted from paper - same idea just fewer cases)
Object: a book
Subject: member of the library
Actions: check-out book for 2 weeks

Subject: all others
Actions: none

Policy Expressed in the Logic
$\forall$ p $\in$ Principals

Member(p, library) $\Rightarrow$ Permitted(p, check-out(book, 2 weeks)) $\wedge$
$\neg$Member(p, library) $\Rightarrow$ $\neg$Permitted(p, check-out(book, 2 weeks))

Note: This approach actually 'hides' what is meant by a 2 week check-out. An interesting aspect of the logic is how easy it is to move security details into the application through the action definitions. We do not need to put the time period in the action. But, to do otherwise, requires a somewhat clever use of the Did predicate in which if the book that was checked out at time t and not returned by time t+6 then an action that is always forbidden is permitted.

2.2

Policy Expressed in English
Object: a set of roles, possible roles are:
All (any user)
Educational (holds educational site license)
In_LC (in Library of Congress)
LC_Staff (Library of Congress staff member)

Subject: correct ID and password for an educational site
Actions: may act in Education role

Subject: correct ID and password for LC staff member
Actions: may act in LC_Staff role
Subject: IP Address matches a computer in LC
Actions: may act In_LC role

Subject: any
Actions: may act in All role

Policy Expressed in the Logic
$\forall$ p $\in$ Principals
Said(verifier, has(p, correct ID and password for Ed)) $\Rightarrow$ Member(p, Education)$\wedge$
Said(verifier, has(p, correct ID and password for Staff)) $\Rightarrow$ Member(p, LC_Staff)) $\wedge$
Said(verifier, has(p, correct IP Addrs for in LC)) $\Rightarrow$ Member(p, IN_LC)) $\wedge$
Member(p, All)

2.3

Policy Expressed in English
Object: work for which at least one of the following holds:
the copyright has expired and the work is published or registered
the work is associated with a deed of gift from the donor permitting universal access
LC review of work revealed no extant right, universal access permitted
information on work is incomplete and LC review indicates low risk
Subject: all users
Actions: general operations

Policy Expressed in the Logic
$\forall$ p $\in$ Principals, obj $\in$ objects meeting given criteria, Permitted(p, general operations(obj))

2.4

Policy Expressed in English
Object: work for which information is incomplete, LC review indicates high risk
Subject: In_L
Actions: general and statement-1 must be shown with each access

Subject: all others
Actions: no access

Policy Expressed in the Logic
$\forall$ p $\in$ Principals, obj $\in$ objects meeting given criteria, (Member(p, In_LC) $\Rightarrow$ Permitted(p, general operations(obj))) $\wedge$ ($\neg$Member(p, In_LC) $\Rightarrow$ $\neg$Permitted(p, general operations(obj))) $\wedge$ ($\forall$ time t, Did(p, general operations(obj), t) $\Rightarrow$ Did(security system, show statement-1, t))

3 Misc. Policies

3.1

Policy Expressed in English
Object: a policy
Subject: senior library staff
Actions: view and change policy

Subject: all others
Actions: view policy

Policy Expressed in the Logic
Given an encoding of a policy so that it can be treated as an object rather than a formula.
$\forall$ p $\in$ Principals Permitted(p, view policy) $\wedge$ (Senior_staff(p) $\Rightarrow$ Permitted(p, change policy)) $\wedge$ ($\neg$Senior_staff(p) $\Rightarrow$ $\neg$Permitted(p, change policy))

3.2

Policy Expressed in English
Object: a collaborative work between persons A and B
Subject: A
Actions: change work

Subject: B
Actions: change work with consent of A

Policy Expressed in the Logic
Permitted(A, change work) $\wedge$ $\neg$Permitted(B, change work) $\wedge$ Permitted(B with A's permission, change work) $\wedge$ (Said(A, Permitted(B, change work)) $\Rightarrow$ Member(B, B with A's permission))

Note: if B must get A's permission for each change, then a time component will be needed to allow membership revocation when B makes a change. Also, an encoding is needed since predicates cannot have formulas as arguements.

3.3

Policy Expressed in English
Object: new object

Subject: library staff
Actions: can add new objects to repository but additions must come with a description and a policy.

Subject: all others
Actions: cannot create objects

Policy Expressed in the Logic
$\forall$ p $\in$ Principals, obj $\in$ objects meeting given criteria, Staff(p) $\Rightarrow$ Permitted(p, add obj) $\wedge$ $\neg$Staff(p) $\Rightarrow$ $\neg$Permitted(p, add obj)

3.4

Policy Expressed in English
Object: an object
Subject: trusted third party
Actions: label object as 'verified'

Subject: all others
Actions: cannot label object as 'verified'

Policy Expressed in the Logic
$\forall$ p $\in$ Principals Member(p, trusted third party(obj)) $\Rightarrow$ Permitted(p, label(obj, verified)) $\wedge$ $\neg$Member(p, trusted third party(obj)) $\Rightarrow$ $\neg$Permitted(p, label(obj, verified))