Ontology of Manipulation Actions

Goal

The goal of this report is to define an ontology of manipulation actions based on touching relation of objects. To this end, we categorize objects by the type of changes in their relations to other objects. Using these objects we create enumerate the list of all possible actions. Each action is defined by its objects, Semantic Event Chain (SEC) matrix, and primitives. The primitives show how the action is executed by a robot arm/hand system.

Object definition

Object definitions are shown in Table.1.

Object	Definition	Relation	
Hand	The object that performs the action	not touching anything at the beginning and the end of action. It touches at least one object.	
Main	The object which is directly in contact with the hand	not touching the hand at the beginning and the end of action. It touches the hand at least once.	
Primary	The object from which the main object separates	initially touches the main object. Changes its relation to not-touching during the action	
Secondary	The object to which the main object joins	initially does not touch the main object. Changes its relation to touching during the action	
Load	The object which is indirectly manipulated	does not touch the hand. During the action either touches/untouches the mai and untouches/touches container	
Container	The object whose relation with load changes and it is not the main object	touches or untouches the load object	
Main support	The object on which the main object is located	touching the main object all the time	
Primary support	The object on which the primary object is located	touching the primary object all the time	
Secondary support	The object on which the secondary object is located	touching the secondary object all the time	
Tool	The object which is used by the hand to enhance the quality of some actions	touching the hand all the time	

Table 1. Definition of objects in ACAT project and the modified definitions used in this ontology

Action instantiations

Based on the object definitions in Table.1, we find all the possible manipulation actions. We do this in a systematic way. For an action to exist, there are at least two necessary objects: *hand* and *main*. We will categorize the actions based on the additional objects present in the action.

We find the actions in two different categories:

- A) Actions with a main support
- B) Actions without main support.
- C) Actions with a load and container object

A- Actions with main support

This category contains actions in which there is a main support. Main support is an object which keeps contact with the main object, from the beginning to the end of the action. Usually, the main object is on top of the main support. The actions in this category could involve other objects (primary, secondary etc.).

A1.Actions with hand (and possibly tool), main and main support. (No other objects)



Action	P1	P2	P1	P2
punch	move(main)		move(free)	

flick	move(main)	hand_move()	-	-
poke	move(main)		move(free)	
turn (bore)	move(main)		rotate(X)	move(free)
scratch	move(main)		move(main)	move(free)
squash (squeeze)	move(main)	grasp	ungrasp	move(free)
push/pull (no grasp)	move(main)		move(main)	move(free)
stir	move(main)		move(stir)	move(free)
knead	move(main)		move(main)	move(free)
rub/ massage	move(main)		move(main)	move(free)
lever	move(main)		move(main)	move(free)
align	move (main)	grasp	rotate(Z)	ungrasp
rotate by grasping	move (main)	grasp	rotate(Z)	ungrasp
press	move(main)		move(main)	move(free)

A2. Actions with hand, main, primary and main support. (optionally: primary support)

	1	2	3		4
			Ý		Y
pm	pm	р	m	р	m
p.s m.s	p.s m.s	p.s	m.s	p.s	m.s

SEC:

hand, main	Ν	Т	Т	Ν						
main, primary	Т	Т	Ν	Ν						

Action	P1	P2	P1	P2	P1	P2
push apart	move(main)		move(primary)		move(free)	



		3		X ⁵
m	m	P m	p m	pm
m.s	m.s	m.s	m.s	m.s

SEC:

hand, main	Ν	Т	Т	Т	Ν						
main, primary	A	А	Т	N	Ν						

Action	P1	P1 P2		P2	P1	P2	P1	P2
cut	move(main)		exert_force	move(main)	move(primary)		move(free)	
chop	move(main)		move(main)		move(primary)		move(free)	
scratch	move(main)		move(main)		move(primary)		move(free)	
scissor cut (pinch)	move(main)		move(main)		hand_move		move(free)	
squash (squeeze)	move(main)		?		?		?	
draw	?		?		?		?	
scoop (ladle)	?		?		?		?	

A3. Actions with hand, main, main support, secondary and secondary support.

	`	2	>	3	2	١	4			
m s m.s s.s	m s	5 S.S	r m.s	n s s s.s	l [m s n.s s.s]			
SEC:										
hand, main	Ν]]	Г	Т		N				
main, secondary	N N		N	Т		Т				

Primitives:

Action	P1 P2		P1	P2	P1	P2
push together	move(main)		move(secondary)		move(free)	

A4. Actions with hand, main, main support, primary, primary support, secondary and secondary support.

Ý				3			4			× ⁵						
p p.s	m m.s	S S.S		p p.s	m m.s	s s.s	p p.s	m.s	S S.S	p p.s	m.s	s s.s	p p.s	m.s	s s.s	

SEC:

hand, main	N	Т	Т	Т	N
main, primary	Т	Т	N	Ν	Ν
main, secondary	N	Ν	N	Т	Т

Action	P1	P2	P1	P2	P1	P2	P1	P2
Push from x to y	move(main)		move(primary)		move(secondary)		move(free)	



SEC:

hand, main	Ν	Т	Т	Т	Т	Ν
main, primary	А	А	Т	Ν	Ν	Ν
main, secondary	Ν	Ν	Ν	Ν	Т	Т

Action	P1	P2	P1	P2	P1	P2	P1	P2
???	move(main)		move(primary)		move(secondary)		move(free)	

B- Actions without main support

This category contains actions in which there is no main support. The actions in this category could involve other objects (primary, secondary etc.). We try to find these actions.

B1. Actions with hand, main, primary, primary support, secondary support

Ý	1	\$	2	N	3		4		X ⁵
pm	S.S	pm	S	p	S	p	m s	p	m s
p.s		p.s	S.S	p.s	S.S	p.s	S.S	p.s	S.S

Action:		Pick fro	om side and place	on side	
SEC					
hand, main	Ν	Т	Т	Т	Ν
main, primary	Т	Т	Ν	Ν	Ν
main, secondary	Ν	Ν	Ν	Т	Т
main, p.s	Т	Т	Ν	Ν	Ν
main, s.s	Ν	Ν	Ν	Т	Т
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	



Action:		Pick fro	om top and place	on side	
SEC					
hand, main	Ν	Т	Т	Т	Ν
main, primary	Т	Т	Ν	Ν	Ν
main, secondary	Ν	Ν	Ν	Т	Т
main, p.s	Ν	Ν	Ν	Ν	Ν
main, s.s	Ν	Ν	Ν	Т	Т
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	

×	1	\	2	×	3				ð _ ⁵
m		m		m]		m		m
р	S	р	S	р	S	р	S	р	S
p.s	S.S	p.s	S.S	p.s	S.S	p.s	S.S	p.s	S.S

Action:		Pick fr	om top and place	on top	
SEC					
hand, main	Ν	Т	Т	Т	Ν
main, primary	Т	Т	Ν	Ν	Ν
main, secondary	Ν	Ν	Ν	Т	Т
main, p.s	Ν	N	Ν	Ν	Ν
main, s.s	Ν	Ν	Ν	Ν	Ν
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	



Action:		Pick fro	om side and place	e on top	
SEC					
hand, main	Ν	Т	Т	Т	Ν
main, primary	Т	Т	N	N	Ν
main, secondary	Ν	Ν	Ν	Т	Т
main, p.s	Т	Т	Ν	Ν	Ν
main, s.s	Ν	Ν	N	N	Ν
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	



Action:			Break off and	l place on top		
SEC	1	2	3	4	5	6
hand, main	Ν	Т	Т	Т	Т	Ν
main, primary	А	А	Т	Ν	Ν	Ν
main, secondary	Ν	Ν	Ν	Ν	Т	Т
main, p.s	А	А	Т	Ν	Ν	Ν
main, s.s	Ν	Ν	Ν	Ν	Ν	Ν
Primitives						
P1	move(main)	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp				move(free)	

m s p s p s p m s p s	Ý	1	\$	2	<u>ک</u>	3	N	4		5	,	è
	m p.s	S S.S	m p.s	S S.S	p m p.s	S S.S	p p.s	S	p p.s	m s s.s	p p.s	m s S.S

Action:	Break off and place on side							
SEC								
hand, main	N	Т	Т	Т	Т	Ν		
main, primary	A	A	Т	Ν	Ν	N		
main, secondary	N	N	N	Ν	Т	Т		
main, p.s	А	А	Т	Ν	Ν	Ν		
main, s.s	N	Ν	Ν	Ν	Т	Т		
Primitives								
P1	move(main)	move(main)	move(prim.)	move(sec.)	ungrasp			
P2	grasp				move(free)			



Action:	Unscrew (a cap or lid from a bottle or a jar)							
SEC								
hand, main	Ν	Т	Т	Т	Ν			
main, primary	Т	Т	Ν	Ν	Ν			
main, secondary	Ν	Ν	Ν	Т	Т			
main, p.s	Ν	Ν	Ν	Ν	Ν			
main, s.s	Ν	Ν	Ν	Ν	Ν			
Primitives								
P1	move(main)		move(sec.)	ungrasp				
P2	grasp	move(prim.)		move(free)				



Action:	Screw (a cap or lid to a bottle or a jar)							
SEC								
hand, main	Ν	Т	Т	Т	Ν			
main, primary	Т	Т	Ν	Ν	Ν			
main, secondary	Ν	Ν	Ν	Т	Т			
main, p.s	Ν	Ν	Ν	Ν	Ν			
main, s.s	Ν	Ν	Ν	Ν	Ν			
Primitives								
P1	move(main)	move(prim.)	move(sec.)	rotate(Z)				
P2	grasp			ungrasp				

h		h		*			1 - 1	•	h
m		m		m			m		m
р	S	р	s	р	s	р	s	р	S
p.s	s.s	p.s	S.S	p.s	s.s	p.s	s.s	p.s	s.s

Action:	Peg in hole							
SEC								
hand, main	Ν	Т	Т	Т	Ν			
main, primary	Т	Т	Ν	Ν	Ν			
main, secondary	Ν	Ν	Ν	Т	Т			
main, p.s	Ν	Ν	Ν	Ν	Ν			
main, s.s	Ν	Ν	Ν	Ν	Ν			
Primitives								
P1	move(main)	move(prim.)	move(sec.)	ungrasp				
P2	grasp		insert	move(free)				



Action:		Shake							
SEC									
hand, main	Ν	Т	Т	Т	Ν				
main, primary	Т	Т	Ν	Ν	Ν				
main, secondary	Ν	Ν	N	Т	Т				
main, p.s	Ν	Ν	N	Ν	Ν				
main, s.s	Ν	Ν	Ν	Ν	Ν				
Primitives									
P1	move(main)	move(prim.)	move_periodic (V)	ungrasp					
P2	grasp		move(sec.)	move(free)					

Note: The action of sub-categories B2 to B5 are derived from the actions of B1 plus applying the stated constraint. In each of B2-B5 we show only one action from B1, not all of them.

p m s p.s = s.s	1 pm p.s =	2 s p s . s	3 n s p = s.s p.s	4 m s = s.s	p m s p.s = s.s
SEC					
hand, main	Ν	Т	Т	Т	N
main, primary	Т	Т	Ν	Ν	N
main, secondary	Ν	Ν	Ν	Т	Т
main, p.s	Т	Т	Ν	Т	Т
main, s.s	Т	Т	Ν	Т	Т
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	

B2. Actions of category B1, with the constraint that primary support = secondary support.

B3. Actions of category B1, with the constraint that primary = secondary support (Secondary object is on primary object.)

	2 • • •	m s	m 4	∞ m ⁵ s
p.s	p.s	p p.s	p p.s	p p.s

SEC					
hand, main	Ν	Т	Т	Т	Ν
main, primary	Т	Т	Ν	Ν	Ν
main, secondary	Ν	Ν	Ν	Т	Т
main, p.s	Т	Т	Ν	Т	Т
main, s.s	Т	Т	Ν	Ν	Ν
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	

B4. Actions of category B1, with the constraint that primary support = secondary (Primary object is on secondary object.)

pm s S.s	1 pm s s.s		s s s s s s s s s s s s s s s s s s s	4 m s s.s	pm s s.s
SEC					
hand, main	Ν	Т	Т	Т	N
main, primary	Т	Т	N	N	N
main, secondary	Т	Т	Ν	Т	Т
main, p.s	Т	Т	Ν	Т	Т
main, s.s	Ν	Ν	Ν	Ν	Ν
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	

B5. Actions of category B1, with the constraint that primary = secondary (Primary and Secondary objects are the same objects.)

p=s m p.s = s.s	1 p=s m p.s =	2 p=s p.s	3 m = s.s p.s	4 (s m) = 5.5	p=s m p.s = s.s
SEC					
hand, main	Ν	Т	Т	Т	N
main, primary	Т	Т	Ν	Т	Т
main, secondary	Т	Т	Ν	Т	Т
main, p.s	Т	Т	Ν	Т	Т
main, s.s	Т	Т	Ν	Т	Т
Primitives					
P1	move(main)	move(prim.)	move(sec.)	ungrasp	
P2	grasp			move(free)	

conclusion: In this part, we tried to find all the possible actions, using the object definitions. The first goal is achieved.

C- Actions with Load object

This category contains actions in which there is a *load* object. The actions in this category are divided into two sub-categories: Loading and Unloading actions.

In loading actions the transition between main and load is N to T, while in Unloading this transition is T to N.

C1. Loading actions



Action:	Load (e.g. pipetting)							
SEC								
hand, main	Ν	Т	Т	Т	Т	Т	Ν	
main, primary	Т	Т	Ν	Ν	Ν	Ν	Ν	
main, secondary	N	Ν	Ν	Т	Т	Т	Т	
main, load	Ν	Ν	Ν	Т	Т	Т	Т	
load, container	Т	Т	Т	Т	Ν	Ν	Ν	
Primitives								
P1	move(main)	move(prim.)	move(load)	move(container)	move(secondary)	ungrasp		
P2	grasp					move(free)		

C2. Unloading actions



Action:	Unload (e.g Pouring, Dropping)									
SEC										
hand, main	Ν	Т	Т	Т	Т	Т	Ν			
main, primary	Т	Т	Ν	Ν	Ν	Ν	Ν			
main, secondary	Ν	Ν	Ν	Ν	Ν	Т	Т			
main, load	Т	Т	Т	Т	Ν	Ν	Ν			
load, container	N	N	Ν	Т	Т	Т	Т			
Primitives										
P1	move(main)	move(prim.)	move(container)	move(container)	move(secondary)	ungrasp				
P2	grasp					move(free)				

Action Categories

In this part, we want to achieve the second goal: find the categories for actions of the previous part.

We found two major categories:

- A) Actions with the main support object
- A1 Action with Hand, main and main support.
- A2 Action with Hand, main, main support and primary
- A3 Action with Hand, main, main support and secondary
- A4 Action with Hand, main, main support, primary and secondary.

B) Actions without main support object. These actions always include: Hand, main, primary, primary support, secondary and secondary support. Based on the additional constraints between these objects, we can divide them into 5 sub-categories, as follows:

- B1 primary \neq secondary. primary support \neq secondary support.
- B2 primary \neq secondary. primary support = secondary support.
- B3 primary \neq secondary. primary = secondary support. (secondary is on the primary)
- B4 primary \neq secondary. primary support = secondary . (primary is on the secondary)
- B5 primary = secondary. primary support = secondary support.

C) Actions with load and load target object:

- C1 Loading actions: The load makes N to T transition wrt to the main
- C2 Unloading actions: The load makes T to N transition wrt to the main

Rules for execution

7

8

9

These rules try to find these primitives based on the transitions in SEC.										
Rule	Relation		transition	primitive 1	primitive 2					
1	hand	main	N> T	move(main)	grasp (optional)					
2	hand	main	T> N	ungrasp, etc. (optional)	move(free)					
3	main	primary	T> N	move(primary)						
4	main	primary	A> T	move(main)						
5	main	secondary	N> T	move(secondary)						
6	main	load	N> T	move(load)						

T --> N

N --> T

T --> N

In this part we want to find the rules for execution of actions. To execute actions, we need to find the primitives at each state (chunk of ADT) to send commands to the control systems. These rules try to find these primitives based on the transitions in SEC.

Note 1- Each primitive has some parameters, which should be designed or learned from demonstration. Note 2- The primitives which can not be derived by the these rules, should be designed or learned from demonstration.

move(container)

move(container)

move(container)

List of Action Primitives

main

load

load

load

container

container

There are a number of action primitives used in definition of actions. These primitives are commands which are possible to implement on robot arms and hands. The primitives used in this framework are as follows:

Robot arm primitives:

1- move(*object*,*P*): This commands moves the robot arm to a pose (position and orientation) which is related to the pose of its *object* argument. The goal pose is not necessarily the same as *object* pose. The relation of goal pose and *object* pose is stored in the parameter *P*. Based on the value of *P*, the move() primitive may move the arm toward or away from the *object*.

2- rotate(*axis*,*angle*): Rotate the robot arm according to the given *axis* and *angle* arguments.

3- move_periodic(v): Move the robot arm periodically along the vector v. This primitive is used in actions like shake.

4- insert(): This primitive is specially used in the peg-in-hole action. Implementation of insert() needs force and position control.

Robot hand primitives:

1- hand_move(*angles*): This primitive moves the robot hand to a configuration given in *angles*. It is implemented in the joint space of the hand.

- 2- grasp(): Performs a grasp, possibly using the tactile feedback.
- 3- ungrasp(): Released the previously grasped object.