Task Description	Shortened Name
1. Place all necessary clamps onto the PA structure (repeat for all clamps)	
Robot picks up a clamp from where the clamps are stored.	PickClampFromStorage
Robot places the clamp onto the PA structure and releases it.	PlaceClampToStructure
2. Bring the next beam to be assembled	
Robot picks up a gripper from where the gripper is stored.	PickGripperFromStorage
Robot picks up the next beam from the operator at the loading station	PickBeamFromStorage
a. If this is a beam with no clamps	
Robot places the beam on the ground. The operator fixes it.	PlaceBeamInClamp
b. If this beam used clamps	
Robots and clamps move in synchronisation to clamp the joints.	AssembleBeamWithClamps
Robot places the gripper back to its storage location.	PlaceGripperToStorage
3. Retrieve all clamps from the PA structure (repeat for all clamps)	
Robot picks up a used clamp from the PA structure.	PickClampFromStructure
Robot places the clamp back in its storage location.	PlaceClampToStorage

Assembly Model Parameters Required	**Process Model Parameters Required**
Fixed from the previous Assembly Design Phase	Computed or chosen later in the next steps
Assembly Sequence: **bid** = beam_id = assembly_sequence[i] where: i = integer iterator beam = assembly.beams[bid] where: type(assembly) = Assembly where: type(beam) = Beam **Joints and Neighbours:** **jid** = joint_id = beam.joints_to_assemble[j] where: j = integer iterator decomposing: jid = (bid, nid) where: nid = neighbour_beam_id	**Tool Choice:** **cid** = clamp_id = beam.clamp_on_joint[jid] **gid** = gripper_id = beam.gripper_choice

Task Agents	**Controller**	**Low-Level Task Available**
RFL Robot	L2 ROS RRC Driver for RFL Robot	• **Linear Motion (LM)** • **Free Motion (FM)**
Clamp	L2 Clamp Controller	• **Clamp Jaw Motion**
RFL Robots and Clamp(s)	L3 Process Controller	• **Clamp and Robot Sync Linear Motion (SM)**
Clamp	L2 ROS RRC Driver for RFL Robot (Digital IO)	• **Open attachment gripper** • **Close attachment gripper**
Parallel Gripper		• **Open Gripper** • **Close Gripper**
Docking Adapter		• **Lock Tool** • **Unlock Tool**
Operator	Operator	• **Place Beams from Storage in Robot Gripper**

Relation Level	**Design Phase**	**Designer**	**Description**	**Data Type**
Process	TD	TE	Robot Model	Robot Model
	PD	**PE**	Environment Models	3D Geometry
	PD	**PE**	Position of Assembly Model relative to the Robot	T [World Frame]
Assembly	AD	DE / PE	Assembly Sequence	List of Beam ID
	AD	Auto	Beam Connectivity	Graph of Beams and Joints
Beam	AD	DE	Designed Location	T [World Frame]
	AD	DE	Assembly Direction	T [Beam Frame]
	PD	**PE** / Auto	Gripper Choice	Gripper ID
	PD	**PE** / Auto	Grasp Face	Face ID
	PD	**PE** / Auto	Grasp Position	Length along beam
	PD	Auto	Grasp Pose	Inverse T [Beam Frame]
	PD	Auto	Beam Pickup Location	T [World Frame]
	PD	Auto	Clamp Jaw Approach Direction (only for beams that require clamps)	T [World Frame]
Joint	AD	Auto	Joint Position	T [Beam Frame]
	TD	TE	Allowable Clamp Choices	
	PD	**PE** / Auto	Clamp Choice	Clamp Type
	PD	Auto	Assigned Clamp Device	Clamp ID
	TD	TE	Allowable Clamp Attach Poses	T [Joint Frame]
	PD	**PE** / Auto	Clamp Attach Pose	T [Joint Frame]
	AD	Auto	Allowable Assembly Direction	T [Joint Frame]
All DiRT Tools (Gripper and Clamps)	**PD**	**PE**	Storage Location (Tool Storage Station)	T [World Frame]
	TM	TE	Storage Retract Direction	T [Tool Base Frame]
	TM	TE	Kinematic Chain Link FK	T [Tool Base Frame]
Clamp	TM	TE	Beam-Approach-Jaw Direction	T [Tool Base Frame]
	TM	TE	Clamp-Attach-To-Beam Direction (x2 steps)	T [Tool Base Frame]
Gripper	TM	TE	Gripper-Approach-Beam Direction	T [Tool Base Frame]
	PD	**PE**	Gripper Location for Beam Pickup	T [World Frame]
	PD	**PE** / Auto	Beam Pickup Retract Direction	T [World Frame]
Docking Adapter	TM	TE	Dock Approach Direction	T [Adapter Base Frame]
	TM	TE	Tool Link FK	T [Adapter Base Frame]

* T = Transformation Data Type, value in the bracket is the transformation’s base frame.

Object Type	**Parameters**	**Parameter Type**
RFL Robot	● Joint Configuration ● Target Frame	● List of Joint Values ● Rigid Transformation / Frame
Docking Adapter	● Base Frame ● Kinematic Attachment	● Rigid Transformation / Frame ● Parent Object, Rigid Transformation
Tools (Clamps and Grippers)	● Tool Frame ● Joint Configuration ● Kinematic Attachment	● Rigid Transformation / Frame ● List of Joint Values ● Parent Object, Rigid Transformation
Beams	● Beam Frame ● Kinematic Attachment	● Rigid Transformation / Frame ● Parent Object, Rigid Transformation
Environment Model	● Object Frame	● Rigid Transformation / Frame

Task Agents	Low Level Task	AffectedParameters in UpdateState()
RFL Robot	Linear Motion Free Motion	Robot/TargetFrame = TargetFrame DockingAdapter/BaseFrame = TargetFrame Tool/ToolFrame = TargetToolFrame * Beam/BeamFrame = TargetBeamFrame *
RFL Robots and Clamp(s)	Clamp and Robot Sync Linear Motion	Robot/TargetFrame = TargetFrame DockingAdapter/BaseFrame = TargetFrame Tool/ToolFrame = TargetToolFrame Beam/BeamFrame = TargetBeamFrame Clamp/JointConfig = TargetJawPosition
Clamp	Clamp Jaw Motion	Clamp/JointConfig/JawPosition = TargetJawPosition
	Open Hanging Gripper	Clamp/JointConfig/Gripper= Open
	Close Hanging Gripper	Clamp/JointConfig/Gripper = Close
Parallel Gripper	Open Gripper (Beam)	Gripper/JointConfig/Gripper = Open Beam/KinematicAttachment = None, None
	Close Gripper (Beam, Grasp)	Gripper/JointConfig/Gripper = Close Beam/KinematicAttachment = Gripper, Grasp
Docking Adapter	Lock Tool (Tool)	Tool/KinematicAttachment = None, None
	Unlock Tool (Tool)	Tool/KinematicAttachment = DockingAdapter, DockingAdapter.[Base]T[Child]
Operator	Place Beam in Robot Gripper	Beam/BeamFrame = TargetBeamFrame

* Whether Tool and Beam Frame are updated depends on the kinematic chain configuration of that Robot Motion Task

Level 3 Controller	**Com**	**Level 2 Controller (e.g. ABB Robot)**
Long Command 1 (e.g. Movement)	→	Move to execution queue
Long Command 2 (e.g. Movement)	→	Move to execution queue
After Command 1 is completed		
Update Command 1 status (e.g. Completed)	←	Status Update for Command 1
Emergency Stop during the execution of Command 2		
		Status Update is not initiated
Controller is not informed about the stop, e.g. the clamps are still running		

Level 3 Controller	Com	Level 2 Controller (e.g. ABB Robot)
Long Command 1 (e.g. Movement)	→	Move to execution queue
Long Command 2 (e.g. Movement)	→	Move to execution queue
After Command 1 is completed		
Update Command 1 status (e.g. Completed)	←	Status Update for Command 1
Emergency Stop during the execution of Command 2		
		Status Update is not initiated
Controller realised the ABB Robot Controller is not running using the second channel		
Controller can propagate stop signal to other Level 2 controllers, e.g. stop the clamps		

States	**Meaning**
Queued	Command is queued for execution because other commands are still running.
Queue Full Cancelled	Command cannot be queued because the buffer is full.
Running	Command is currently running
Paused	Command is paused by the L3 controller after it has started running. It can be unpaused by the L3 controller and continue.
Completed	Command is completed successfully
Error	Command failed and cannot be continued. L2 Controller is likely to enter an error state and will not execute further until L3 cancels the error.
Cancelled	The command is cancelled by the request of L3.

Level 3 Controller	**Com**	**Level 2 Controller (e.g. ABB Robot)**
Long Command 1 (e.g. Movement)	→	Move to execution queue
Update Command 1 status (e.g. Running)	←	Non-blocking acknowledge / Status Update
Long Command 2 (e.g. Movement)	→	Move to execution queue
Update Command 2 status (e.g. Queued)	←	Non-blocking acknowledge / Status Update
After Command 1 is completed		
Update Command 1 status (e.g. Completed)	←	Status Update for Command 1
Update Command 2 status (e.g. Running)	←	Status Update for Command 2
Emergency Stop during the execution of Command 2		
Update Command 2 status (e.g. Error)	←	Status Update for Command 2
Controller can propagate stop signal to other Level 2 controllers, e.g. stop the clamps		
Operator decides to clear the error		
Cancel Command 2	→	Acknowledge / Remove command 2 from buffer