

G. Ordering Dynamic Model

This appendix forms an integral part of the CIP specification.

This appendix provides a detailed description of the use of the order operations supported by the CIP. The operations are illustrated with two state transition diagrams: the quotation and validation of orders is described in Figure G-1, and the submission of an order is described in Figure G-2. The states included in the diagrams are then described in detail in Table G-1, and the transitions are described in Table G-2.

In the state transition diagrams, ellipses represent the various states for an order, whilst arrows represent the transitions from one state to another. A transition is triggered by an event. An event may be guarded, in which case the event is triggered only if a guard condition is true. Additionally, an action, which represents an operation whose duration is not significant compared to the resolution of the state diagram, may be associated with an event. This is represented with in the diagrams with the notation "event[condition]/action".

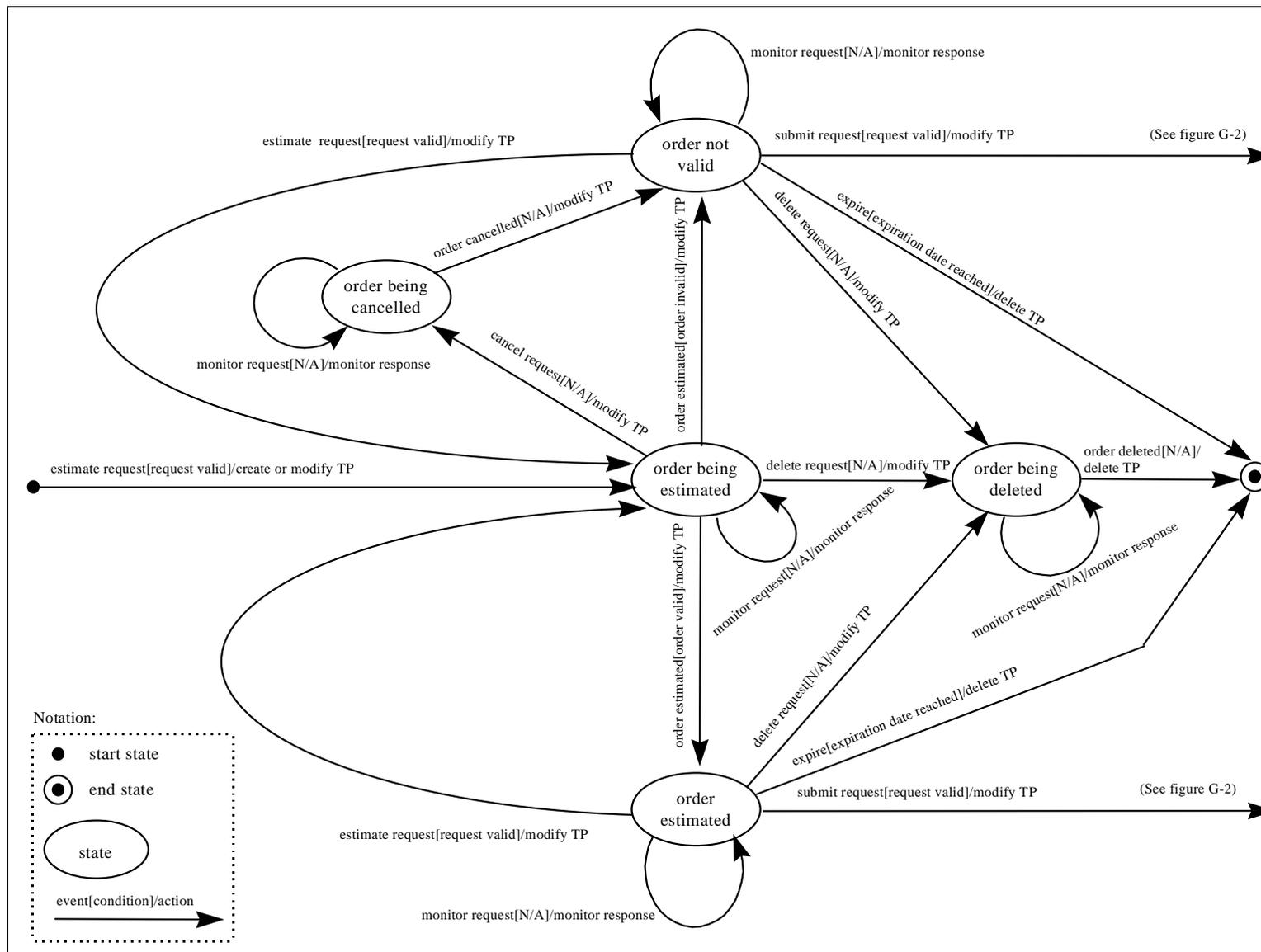


Figure G-1: Order Estimation State Diagram

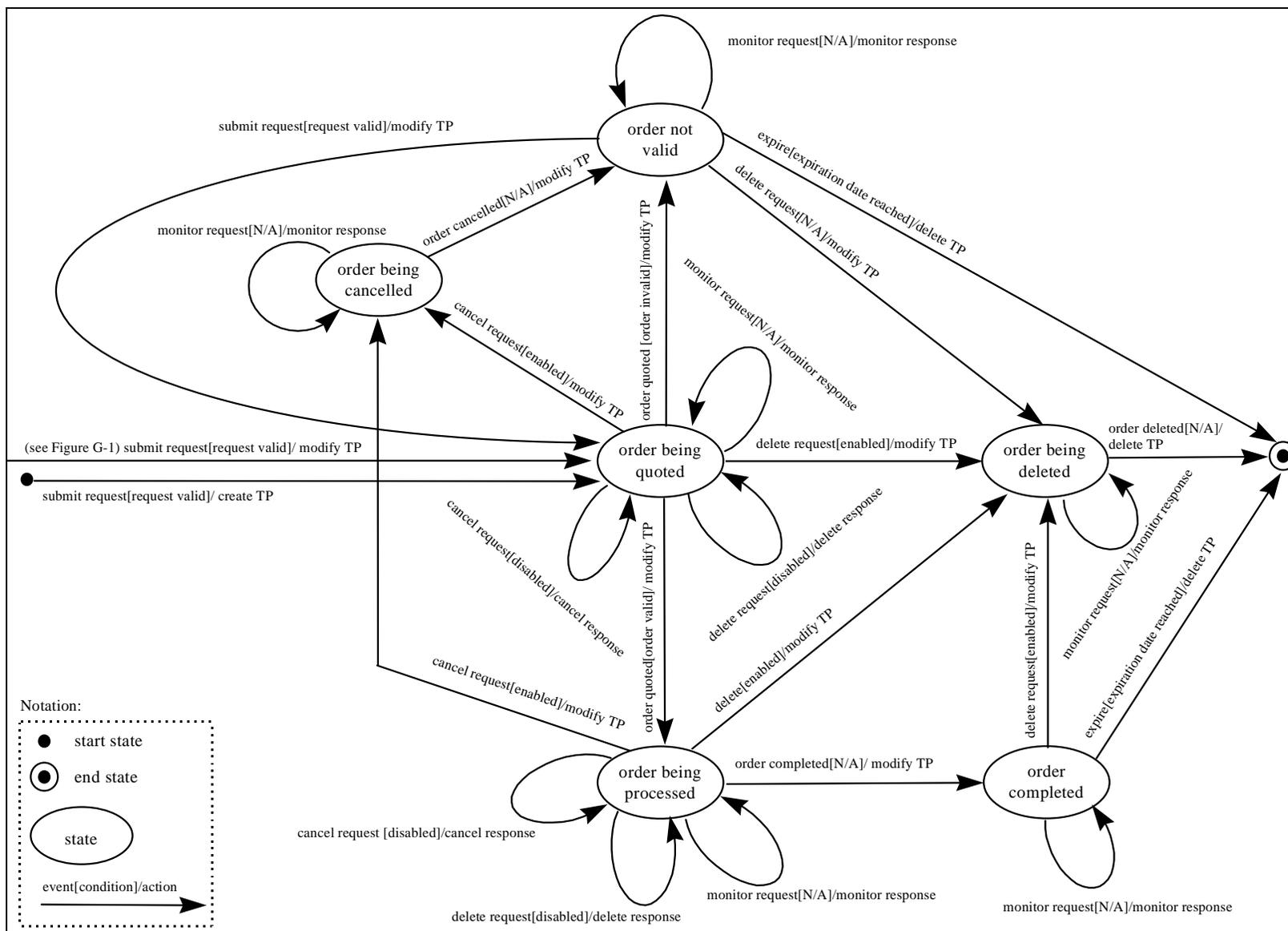


Figure G-2: Order Submission State Diagram

Table G-1: Ordering States Description

State	State Description
Start state	Start state of the transition diagram.
Order being estimated	The order specification is being estimated and validated at the LOHS, i.e. the LOHS calculates an approximate and non-binding estimate for the order.
Order being quoted	The order specification is being quoted and validated at the LOHS, i.e. the LOHS calculates a precise and binding quote for the order.
Order estimated	The estimation and validation has been successfully performed and the order specification therefore contains an estimated and validated order.
Order being cancelled	An order currently being estimated , quoted or processed is being cancelled by the <i>target</i> upon request by the <i>origin</i> .
Order being deleted	An order is being deleted by the <i>target</i> upon request by the <i>origin</i> .
Order not valid	An order specification has been defined by the <i>origin</i> but is not valid (or not yet validated).
Order being processed	The order is being processed at the LOHS.
Order completed	The order has been completed. The <i>task package</i> describing the completed order is still available to the <i>origin</i> .
End state	The order has been deleted.

The transitions from one state to another are described below. For each transition, the following information is provided:

- **Initial State**, which defines the state of the order before the transition has been performed.
- **Event**, which describe the event which triggers the transition from the initial state to the final state. For each event, the following information is provided:
 - **Event Source**, which describe the source of the event, i.e. either the *origin* or the *target* of the CIP Order ES operation.
 - **Event Description**, which provides a description of the event.
- **Condition**, which describes the condition which may guard the event, i.e. the condition which must be fulfilled for the event to be triggered.
- **Actions**, which describes the actions which are performed during the transition. For each action, the following information is provided:
 - **Action Source**, which describe the source of the action.
 - **Task Package**, which describe the action performed on the *task package* containing the order, i.e. none (no change), create, modify or delete.

- **ES Response**, describes, when appropriate, the CIP Order *ES Response* which is sent back to the *origin*. The timing and content of the response depends on the *wait action* which was defined by the *origin* in the CIP Order *ES Request*. The following three *wait actions* are supported by the CIP Order *ES*:
 - **Wait**, which indicates that the *target* must wait for the task to be performed before issuing an CIP Order *ES Response*.
 - **Wait if possible**, which indicates that the *target* should perform the *task* before sending the CIP Order *ES Response* and include the *task package* in the *response*, but that it may return the *response* earlier and without the *task package* if this is not possible. This *wait action* therefore leaves the choice to the *target* whether it should wait or not. When appropriate, alternative behaviours are included in parenthesis.
 - **Do not wait**, which indicates that the *origin* does not request that the *target* performs the *task* before sending the CIP Order *ES Response*. However, the *target* may wait for the *task* to be performed before issuing the *response*.
- **Final State**, which defines the state of the order after the transition has been performed.

Table G-2: Ordering Transitions Description

Initial State	Event		Condition	Actions					Final State
	Event Source	Event Description		Action Source	Task Package	ES Response			
						Wait	Wait if possible	Do not wait	
Start state	origin	CIP Order ES Request: function create action orderEstimate	request valid ¹	target	create	none	(accept estimate)	accept	Order being estimated
Start state	origin	CIP Order ES Request: function create action orderQuoteAndSubmit	request valid	target	create	none	(accept quote and submit)	accept	Order being quoted
Order being estimated	target	Validation and estimation process is finished	order valid, i.e. validation and estimation was successful	target	modify	done	(done estimate)	none	Order estimated
Order being estimated	origin	CIP Order ES Request: function modify action orderCancel	none	target	modify	none	(accept cancel)	accept	Order being cancelled

¹ The *ES request* itself has been validated by the *target*. This validation includes checks such as the verification that the user is authorised to use the CIP Order *ES*, that the user does not exceed quotas (if any), etc.

Initial State	Event		Condition	Actions					Final State
	Event Source	Event Description		Action Source	Task Package	ES Response			
						Wait	Wait if possible	Do not wait	
Order being estimated	target	estimation process is finished	order invalid, i.e. estimation was not successful	target	modify	done	(done cancel)	none	Order not valid
Order being estimated	origin	CIP Order ES Request: function delete	none	target	modify	none	(accept delete)	accept	Order being deleted
Order being estimated	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order being estimated
Order being quoted	target	quotation process is finished	order invalid, i.e. quotation was not successful	target	modify	done	(done cancel)	none	Order not valid
Order being quoted	target	quotation process is finished	order valid, i.e. quotation was successful	target	modify	none	none	none	Order being processed
Order being quoted	origin	CIP Order ES Request: function modify action orderCancel	order cancellation is enabled (i.e. the cancellation date is not passed)	target	modify	none	(accept cancel)	accept	Order being cancelled
Order being quoted	origin	CIP Order ES Request: function modify action orderCancel	order cancellation is disabled (i.e. the cancellation date is passed)	target	none	failure	failure	failure	Order being quoted
Order being quoted	origin	CIP Order ES Request: function delete	order cancellation is disabled (i.e. the cancellation date is passed)	target	none	failure	failure	failure	Order being quoted
Order being quoted	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order being quoted
Order being quoted	origin	CIP Order ES Request: function delete	order cancellation is enabled (i.e. the cancellation date is not passed)	target	modify	none	(accept delete)	accept	Order being deleted
Order estimated	origin	CIP Order ES Request: function modify action orderEstimate	request valid	target	modify	none	(accept estimate)	accept	Order being estimated

Initial State	Event		Condition	Actions					Final State
	Event Source	Event Description		Action Source	Task Package	ES Response			
						Wait	Wait if possible	Do not wait	
Order estimated	origin	CIP Order ES Request: function modify action orderQuoteAndSubmit	request valid	target	modify	none	(accept quote and submit)	accept	Order being quoted
Order estimated	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order estimated
Order estimated	target	taskPackage expired	expiration date is passed	target	delete	none	none	none	End state
Order estimated	origin	CIP Order ES Request: function delete	none	target	modify	none	(accept delete)	accept	Order being deleted
Order being cancelled	target	cancellation process is finished	none	target	modify	done	(done cancel)	none	Order not valid
Order being cancelled	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order being cancelled
Order being deleted	target	deletion process is finished	none	target	delete	done	(done delete)	none	End state
Order being deleted	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order being deleted
Order not valid	origin	CIP Order ES Request: function modify action orderEstimate	request valid	target	modify	none	(accept estimate)	accept	Order being estimated
Order not valid	origin	CIP Order ES Request: function modify action orderQuoteAndSubmit	request valid	target	modify	none	(accept quote and submit)	accept	Order being quoted
Order not valid	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order not valid
Order not valid	target	taskPackage expired	expiration date is passed	target	delete	none	none	none	End state

Initial State	Event		Condition	Actions					Final State
	Event Source	Event Description		Action Source	Task Package	ES Response			
						Wait	Wait if possible	Do not wait	
Order not valid	origin	CIP Order ES Request: function delete	none	target	modify	none	(accept delete)	accept	Order being deleted
Order being processed	target	order process is completed	none	target	modify	done	(done submit)	none	Order completed
Order being processed	origin	CIP Order ES Request: function modify action orderCancel	order cancellation is disabled (i.e. the cancellation date is passed)	target	none	failure	failure	failure	Order being processed
Order being processed	origin	CIP Order ES Request: function delete	order cancellation is disabled (i.e. the cancellation date is passed)	target	none	failure	failure	failure	Order being processed
Order being processed	origin	CIP Order ES Request: function modify orderMonitor	none	target	none	done	done	done	Order being processed
Order being processed	origin	CIP Order ES Request: function modify action orderCancel	order cancellation is enabled (i.e. the cancellation date is not passed)	target	modify	none	(accept cancel)	accept	Order being cancelled
Order being processed	origin	CIP Order ES Request: function delete	order cancellation is enabled (i.e. the cancellation date is not passed)	target	modify	none	(accept delete)	accept	Order being deleted
Order completed	origin	CIP Order ES Request: function modify action orderMonitor	none	target	none	done	done	done	Order completed
Order completed	target	taskPackage expired	expiration date is passed ²	target	delete	none	none	none	End state
Order completed	origin	CIP Order ES Request: function delete	none	target	modify	none	(accept delete)	accept	Order being deleted

² The expiration date of a *task package* at the *target* is determined in accordance with the LOHS. For this reason, the target can delete a task package without consulting the LOHS when the expiration date is passed. Moreover, contrarily to what happens following the request from the origin to delete a task package, the deletion of the task package following the expiration of the task package at the target does not imply the deletion of the corresponding order at the LOHS (which is performed according to the LOHS's policy).