The Hermes Standard

for vendor independent machine-to-machine communication in SMT Assembly

Version 1.1

Contributing companies:

Achat Engineering GmbH
ASM AS GmbH
ASYS Automatisierungssysteme GmbH
BESI
CTI Systems
CYBEROPTICS
ERSA GmbH
eXelsius
FLEXLINK
GÖPEL electronic GmbH
HANWHA
Heller Industries
IPTE
ITW EAE
JAPAN UNIX Co. Ldt.
KIC
KOH YOUNG Technology Inc.
Kulicke & Soffa
MAGIC RAY Technology
MIRTEC
MYCRONIC AB
Nordson ASYMTEK
Nutek Europe B.V.
OMRON Corporation
PARMI
Pemtron
Rehm Thermal Systems GmbH
RG Elektrotechnologie
SAKI Corp
SEICA Automation
Shenzhen Rejoint Automachine Equip. Co. Ltd.
SMT Thermal Discoveries
SONIC Technology
SPEA S.p.A.
Test Research, Inc.
VISCOM AG
ViTrox
YJ Link Co., Ltd.
YXLON

Contents:

1 Scope of The Hermes Standard Specification 4

2 Technical concept 5

2.1 Prerequisites and topology 5

2.2 Remote configuration 6

2.3 Connecting, handshake and detection of connection loss 6

2.4 Normal operation 8

2.5 Transport error handling 9

Scenario U1a 10

Scenario U1b 11

Scenario U2 12

Scenario U3 13

Scenario D1 14

Scenario D2 15

Scenario D3 16

2.6 Handling of BoardForecast 17

2.7 Protocol states and protocol error handling 20

2.8 Board IDs 21

3 Message definition 22

3.1 Message format 22

3.2 Root element 22

3.3 CheckAlive 23

3.4 ServiceDescription 23

3.5 Notification 24

3.6 BoardAvailable 25

3.7 RevokeBoardAvailable 27

3.8 MachineReady 27

3.9 RevokeMachineReady 29

3.10 StartTransport 29

3.11 StopTransport 29

3.12 TransportFinished 29

3.13 SetConfiguration 30

3.14 GetConfiguration 32

3.15 CurrentConfiguration 32
3.16 BoardForecast 32
3.17 QueryBoardInfo 34
3.18 SendBoardInfo 34

4 Appendix 38

4.1 Special scenarios 38
4.1.1 Board tracking when board is torn out from the line 38
4.1.2 Board tracking when board is temporarily removed from the line 39
4.1.3 Board tracking when board was transferred without data 40

4.2 Glossary, abbreviations 42

4.3 References 42

4.4 History 43

1 Scope of The Hermes Standard Specification

The aim of this specification is to create a state-of-the-art communication protocol for surface-mount technology (SMT) production lines. Therefore, this new communication protocol has to cope with the following:

- Replace the electrical SMEMA interface as specified in [IPC_SMEMA_9851]
- Extend the interface to communicate:
    - Unique identifiers for the handled printed circuit boards (PCBs)
    - Equipment identifiers of the first machine noticing a PCB
    - Barcodes
    - Conveyor speed
    - Product type specific information:
        - Product type identifier
        - Length
        - Width
        - Thickness
        - …
    - …

With respect to version numbers The Hermes Standard adheres to the rules of Semantic Versioning 2.0.0 [SemVer_2.0.0].

Hints on naming:

- Wherever a feature is described by the word „shall“, it is mandatory.
- The word “machine” is used for any equipment which can be found in a SMT production line (e.g. printers, placement machines, ovens, AOIs, transport modules, shuttles, stackers …).
- The term “PCB” may also refer to carriers transporting PCBs.
- The word “Hermes” is used as abbreviation for “The Hermes Standard”.

2 Technical concept

2.1 Prerequisites and topology

This specification is based on the prerequisite, that any application implementing this protocol has to provide connectivity based on Internet Protocol (IP) [IETF_RFC_791]/[IETF_RFC_2460] via Transmission Control Protocol (TCP) [IETF_RFC_793] (ISO/OSI model [ISO_7498-1] layer 3) to the adjacent machines.

Any machine in a line offers one TCP server per lane on its downstream side. Further servers per lane might also be necessary, e.g. if reverse transportation is supported. The TCP port number is not specified but can be configured by the user. The recommended port numbers are 50100 plus lane identifier (ID) with lanes being enumerated looking downstream from right to left beginning with 1 (e.g. for the left lane of a dual lane machine, the upstream machine server accepts connections on port 50102). For every further server plus 10 is recommended to be added to the port number.
The downstream machine opens one connection for every lane and every transportation interface on its upstream side to the upstream machine(s). So every PCB handover point corresponds to one TCP connection per exchange direction.

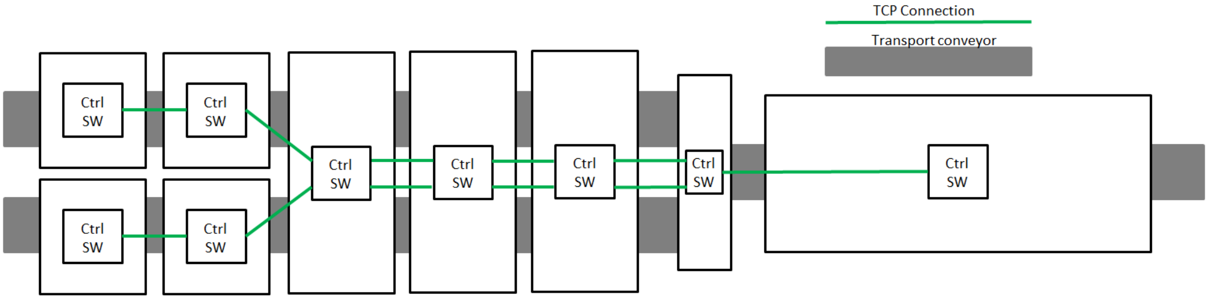

Fig. 1 TCP connections in a line

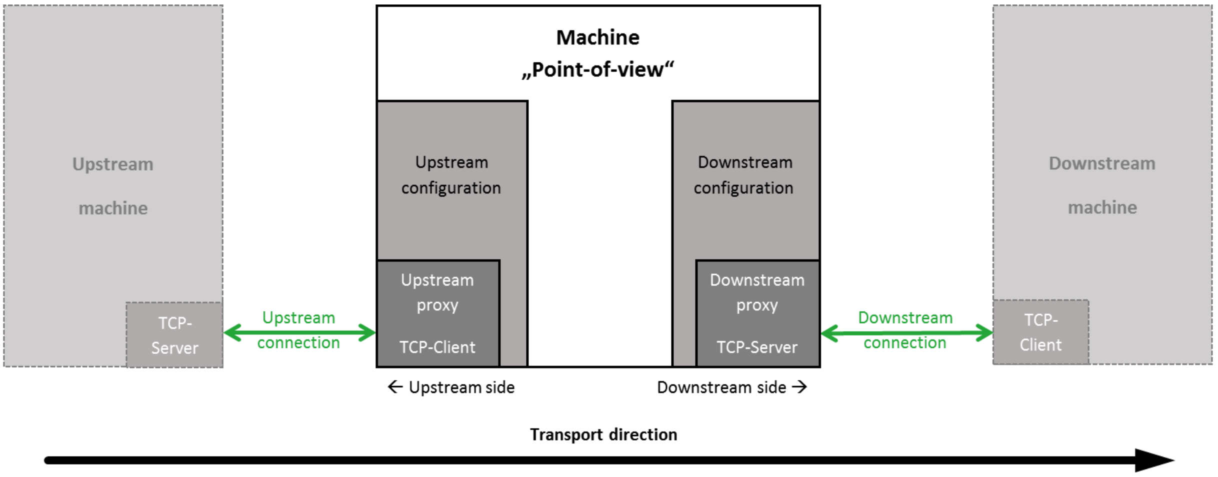

Fig. 2 Upstream and downstream from the perspective of the machine

2.2 Remote configuration

Although a machine may offer the possibility to configure the Hermes TCP port(s) and the IP address(es) of its upstream machine(s) locally (e.g. via a graphical user interface of the machine controller), every machine implementing this protocol shall offer a possibility to configure these properties remote via TCP. Therefore, the machine shall offer a TCP server on port 1248 on at least one network adapter where it accepts configuration messages (see sections 3.13 to 0 for detailed information).

A SetConfiguration message shall contain the full configuration for all Hermes interfaces of a machine. Any existing configuration is overwritten when a SetConfiguration message is received. Whenever a configuration is not applicable (e.g. bad IP address format), the SetConfiguration message is answered with a Notification message (see section 3.5).

Every time the configuration is changed, open Hermes connection will be reset at the next appropriate moment. It is possible to read the current configuration through the GetConfiguration message answered by a CurrentConfiguration message. The configuration shall be persisted until it is changed.

2.3 Connecting, handshake and detection of connection loss

After booting, the downstream machine starts cyclic connection attempts to the configured upstream machines. When a connection is established, the downstream machine starts sending a ServiceDescription message whereupon the upstream machine answers with its own ServiceDescription. This ServiceDescription message contains the lane ID and interface ID (optional) of the sending machine related to this TCP connection. It also contains a list of features which are implemented by the client. The features of the Hermes specification 1.0 have to be supported by any implementation and shall not be included explicitly.

If a downstream machine is already connected to the lane and the transportation interface, this connection will be retained. A Notification message shall be sent to the new connection before it is closed.

After exchanging the handshake messages, both machines may begin to send BoardAvailable/ MachineReady messages (see section 2.4).

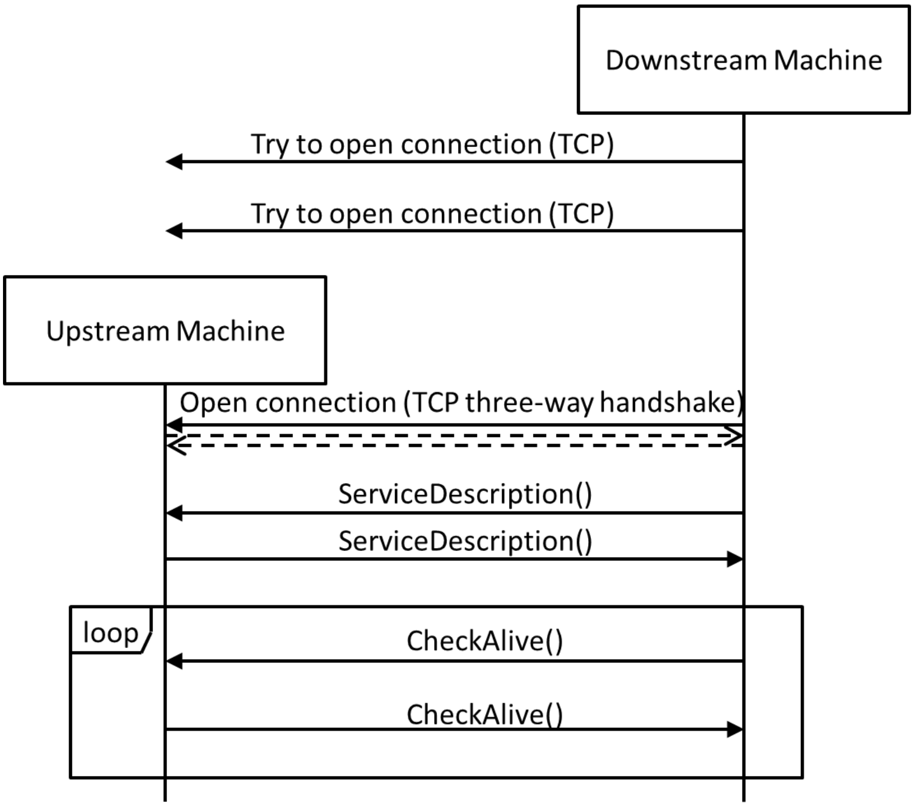

Fig. 3 Connection, handshake and connection loss detection

The connections are kept open all the time. As TCP by itself does not detect connection losses ("Half-open connections" caused by e.g. process-/computer crash, unplugged network cables …) both sides of a connection have to send cyclic CheckAlive messages. Those messages do not have to be answered by the remote side – the TCP stack will detect a connection loss when trying to send the packet. If the server detects a connection loss, it ends the connection and waits for a new connection by the client. If the client detects a connection loss, it ends the connection and re-starts with the cyclic connection attempts.

As not all TCP stacks recognize correctly the loss of connection when sending messages it is possible to extend the implementation of this functionality to an exchange of CheckAlive messages. Machines which have implemented this function do have the tag FeatureCheckAliveResponse in the ServiceDescription.

The exchange of CheckAlive messages then works like shown in Fig. 4.

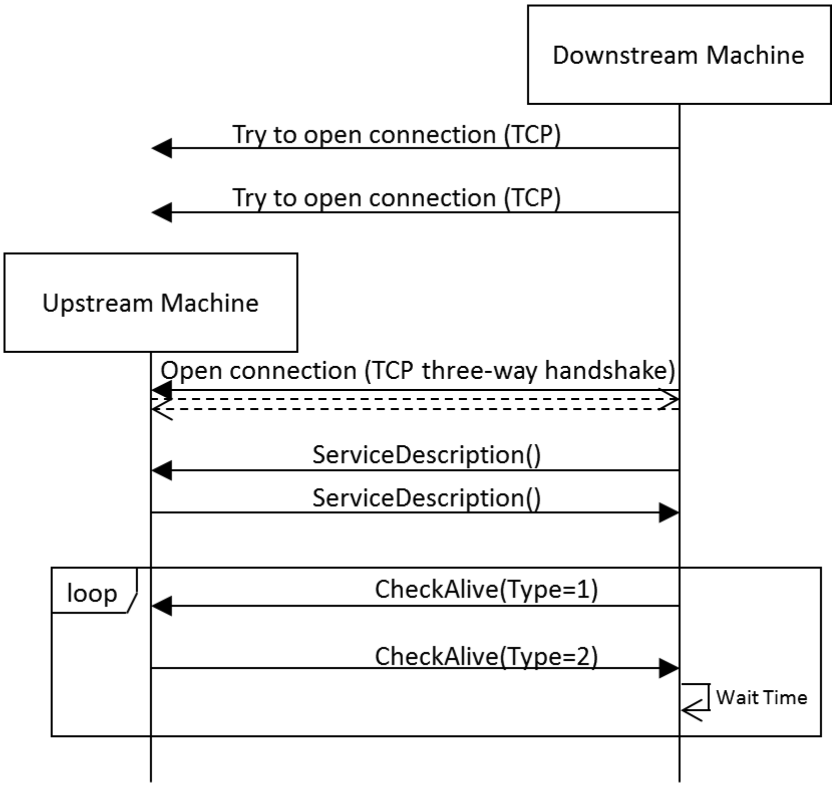

Fig. 4 Example for connection loss detection with FeatureCheckAliveResponse

One of the machines (in the figure the downstream machine but it could be also the upstream machine) sends a (ping) CheckAlive message, that is a CheckAlive message with the attribute Type set to 1. The peer machine then responds immediately with a (pong) CheckAlive message, that is a CheckAlive message with the attribute Type set to 2 and the Id matching the Id of the (ping) CheckAlive message.
A missing response (it is recommended to wait for 3 seconds.) indicates a connection loss.

2.4 Normal operation

When an upstream machine has a PCB available for handover, it sends a BoardAvailable message while a downstream machine ready to accept a PCB sends a MachineReady message. The naming of these messages is inspired by the electrical SMEMA interface. However, the messages do not represent the state of a machine's interface directly but are events for initiating a PCB handover.

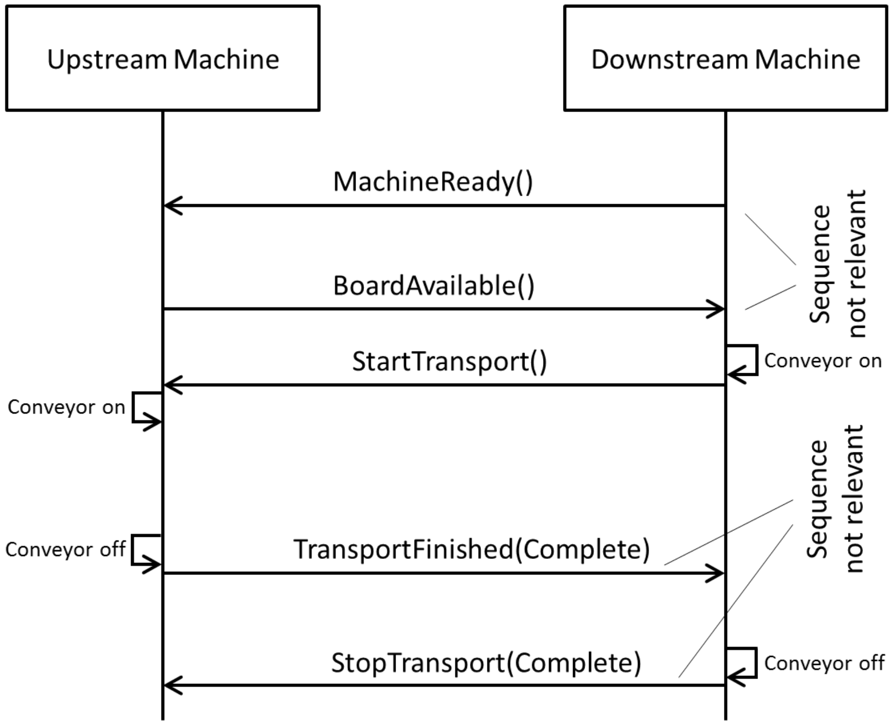

Fig. 5 Communication sequence for board transport

When both machines have indicated their readiness to handover the PCB, the downstream machine initiates the transfer by switching on its conveyor and sending the StartTransport message. Upon receiving this message, the upstream machine switches on its conveyor and the PCB moves into the downstream machine.

When the upstream machine is able to state that the PCB has fully left the machine, it sends the TransportFinished message. When the downstream machine has full control of the board, it sends the StopTransport message. The handover of a PCB is finished and is ready to start over.

If the upstream machine receives a StopTranport message and has not sent the TransportFinished message yet, it has to stop its conveyor and send the TransportFinished message.

The MachineReady message does not trigger an action on one of the machines directly. However it still is necessary to realize machines like e.g. shuttles which have to react to the availability of their downstream machines.

2.5 Transport error handling

To keep this protocol hardware independent, the handling of transport errors is described based on a very simple model of the board handover. The handover process is structured into the three phases

1. "NotStarted": The board is fully inside the upstream machine.
2. "Incomplete": The board is partly inside both machines.
3. "Complete": The board is fully inside the downstream machine.

Any state or event which prevents one or both machines from handing over a PCB is interpreted as an error. An error may be detected by any of the machines in any of the three handover phases. It is up to the application how to detect the current handover phase, how to detect errors and how to solve them eventually (e.g. sensors, model based prediction, timeouts, user interaction …).

The following sequence charts give an overview of the communication within this protocol depending on the machine which detects the error and the phase in which it is detected. The point in the sequence where the error is detected is marked by the following symbol: $\bullet\!\xrightarrow{\text{Stop}}$

Scenario U1a

- Error detected by the upstream machine
- PCB fully inside the upstream machine
- Error detected before StartTransport has been received

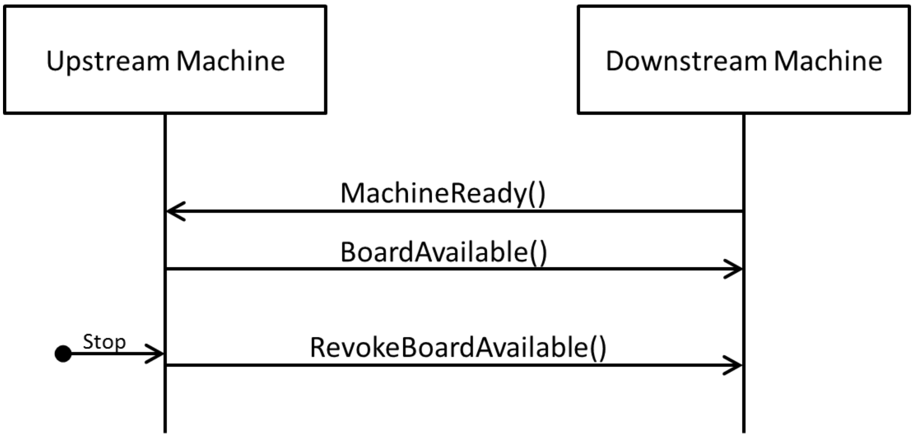

Fig. 6 Communication sequence in scenario U1a

Error detection: The error is detected before any transport started.
Reaction on upstream machine: The upstream machine sends a RevokeBoardAvailable message.
Reaction on downstream machine: None.
Resolution: After the error is solved, the regular transport sequence can start from the beginning.

Scenario U1b

- Error detected by the upstream machine
- PCB fully inside the upstream machine
- Error detected after StartTransport has been received

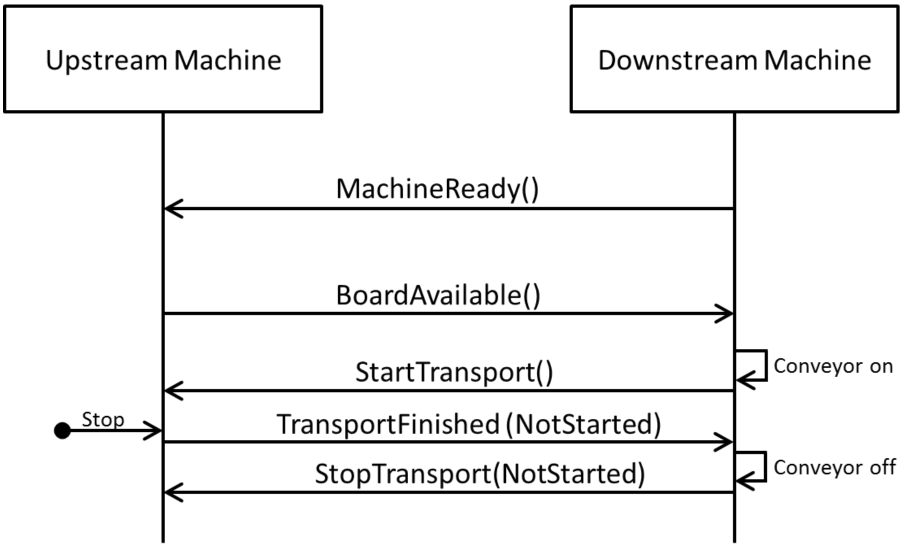

Fig. 7 Communication sequence in scenario U1b

Error detection: The error is detected after the downstream machine started its conveyor and has sent the StartTransport message.

Reaction on upstream machine: The upstream machine sends a TransportFinished message indicating that it has not started the transport.

Reaction on downstream machine: Upon the TransportFinished message, the downstream machine stops its conveyor and sends a StopTransport message indicating that no transport has started.

Resolution: After the error is solved, the regular transport sequence can start from the beginning.

Scenario U2

- Error detected by the upstream machine
- PCB partly inside both machines

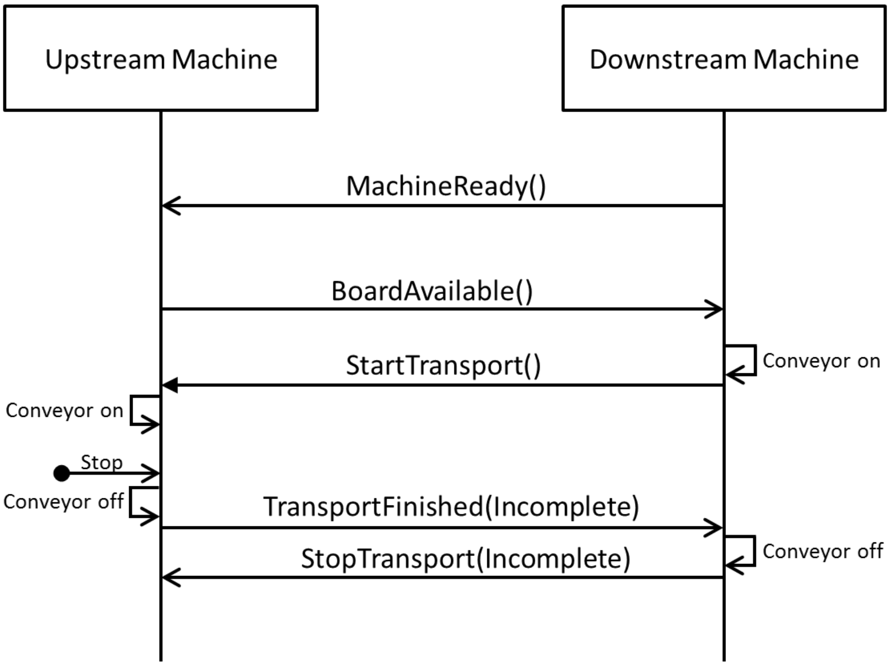

Fig. 8 Communication sequence in scenario U2

Error detection: The error is detected after both machines started their conveyors. The upstream machine assumes that the PCB may have partly entered the downstream machine.

Reaction on upstream machine: The upstream machine sends a TransportFinished message indicating that the PCB might be located between the machines.

Reaction on downstream machine: Upon the TransportFinished message, the downstream machine stops its conveyor and sends a StopTransport message indicating the state of the PCB handover. Note that in Fig. 8 the StopTransport message is represented with parameter “Incomplete”. However in this scenario, the downstream machine could send any of the allowed transport states.

Resolution: After the error is solved, the regular transport sequence can start from the beginning. The regular transport message sequence also applies to a PCB located between the two machines.

Scenario U3

- Error detected by the upstream machine
- PCB fully inside the downstream machine

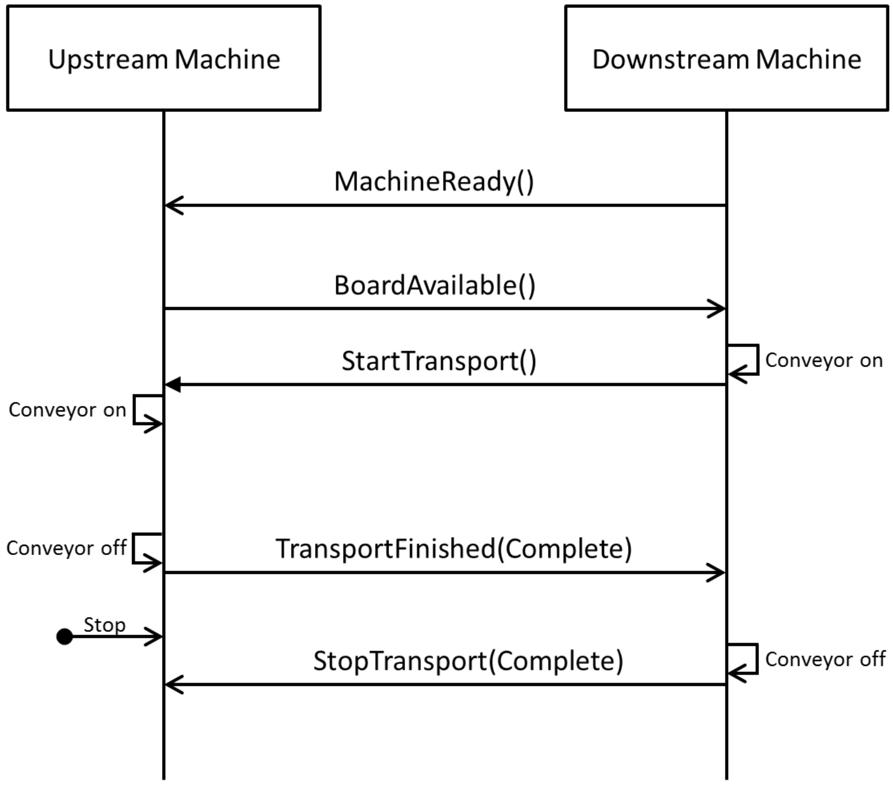

Fig. 9 Communication sequence in scenario U3

Error detection: The error is detected after the PCB is fully inside the downstream machine.
Reaction on upstream machine: None. Although the machine detected an error, it is irrelevant for the handover process.
Reaction on downstream machine: None. The downstream machine is not aware of any error.
Resolution: This scenario is irrelevant for the Hermes protocol. It is just listed for completeness.

Scenario D1

- Error detected by the downstream machine
- PCB fully inside the upstream machine
- Error detected before StartTransport has been sent

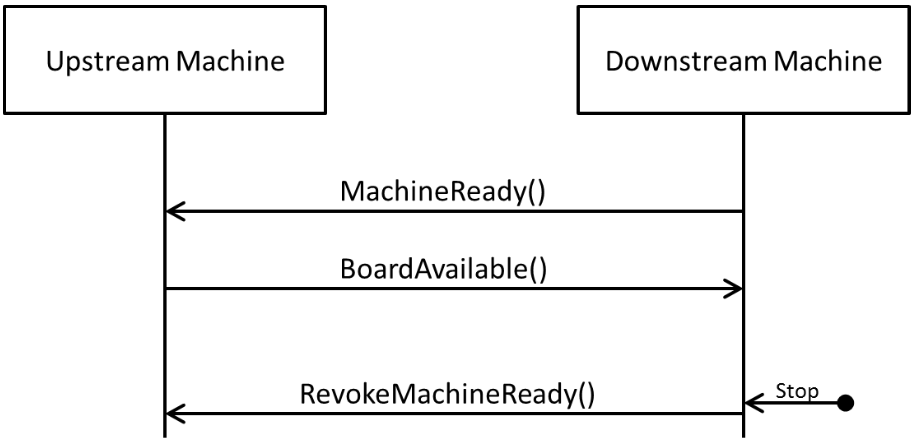

Fig. 10 Communication sequence in scenario D1

Error detection: The error is detected before any transport started.
Reaction on upstream machine: None.
Reaction on downstream machine: The downstream machine sends a RevokeMachineReady message.
Resolution: After the error is solved, the regular transport sequence can start from the beginning.

Scenario D2

- Error detected by the downstream machine
- PCB partly inside both machines

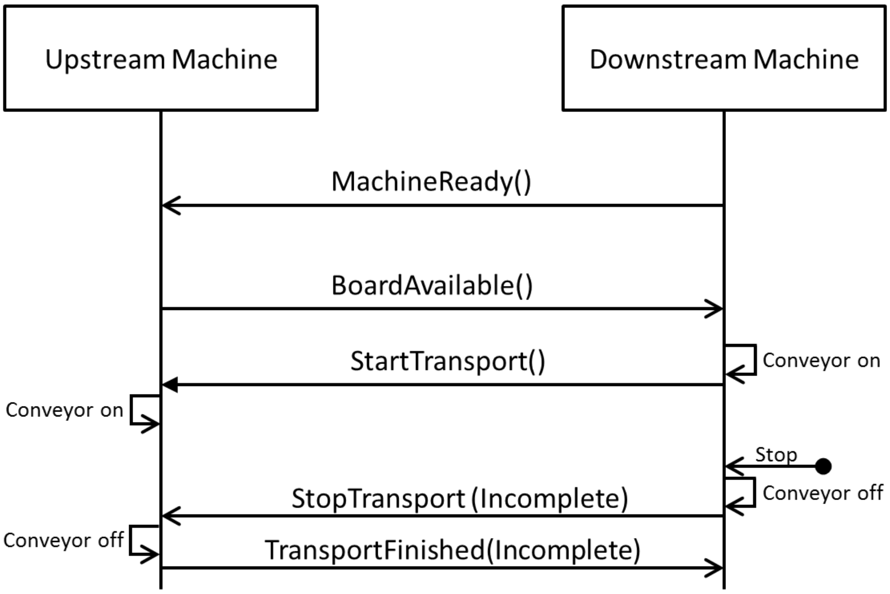

Fig. 11 Communication sequence in scenario D2

Error detection: The error is detected after both machines started their conveyors. The downstream machine assumes that the PCB may already has entered its conveyor.

Reaction on upstream machine: Upon the StopTransport message from the downstream machine, the upstream machine stops its conveyor and sends a TransportFinished message indicating the state of the PCB handover. Note that in Fig. 11Fig. 11 the TransportFinished message is represented with parameter "Incomplete". However in this scenario, the upstream machine could send any of the allowed transport states.

Reaction on downstream machine: The downstream machine stops its conveyor and notifies the upstream machine of the error by sending a StopTransport message indicating an incomplete PCB handover.

Resolution: After the error is solved, the regular transport sequence can start from the beginning. The regular transport message sequence also applies for a PCB located in between the two machines.

Scenario D3

- Error detected by the downstream machine
- PCB fully inside the downstream machine

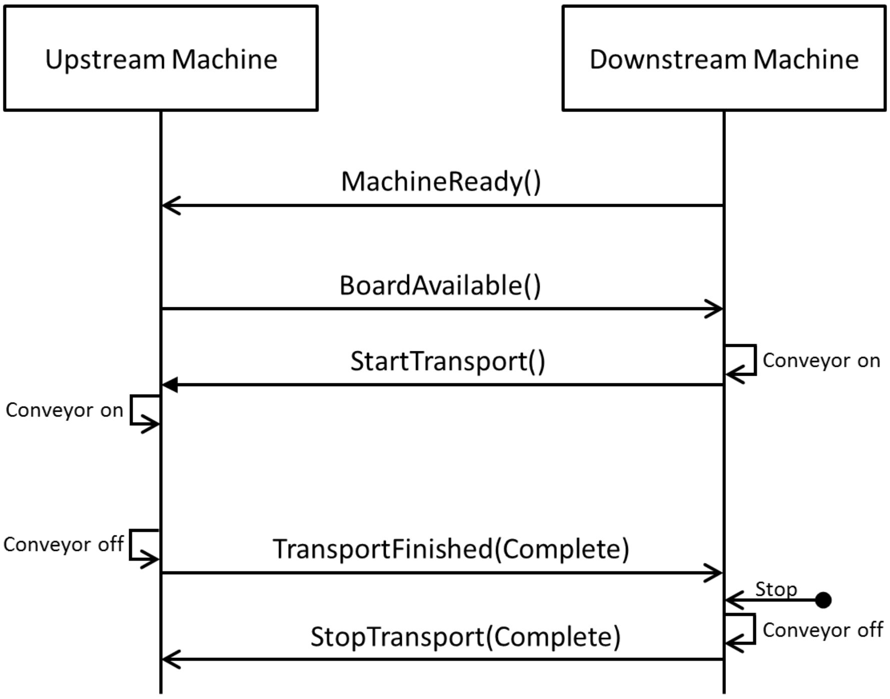

Fig. 12 Communication sequence in scenario D3

Error detection: The error is detected after the PCB is fully inside the downstream machine.
Reaction on upstream machine: None. The upstream machine is not aware of any error.
Reaction on downstream machine: None (at least in the scope of this protocol).
Resolution: This scenario is irrelevant for the Hermes protocol. As transport sequences are always initiated by the downstream machine sending StartTransport, trouble-shooting (possibly including running the conveyor of the downstream machine) can be executed independently from the upstream machine.

2.6 Handling of BoardForecast

Among others the BoardForecast may be used in following scenarios:

- Scenario 1: Anticipating a product change without a board (e.g. because upstream machine does not have stoppers / band that can be stopped).
- Scenario 2: Sending an estimated time to downstream machine until a board will be available (e.g. to allow downstream machine to choose between several upstream machines to get next available board).

Scenario 1:
Upstream machine is processing a changeover (new product type) and wants to ensure that the downstream machine is simultaneously also processing a changeover. Upstream machine also needs to check that this actually happens. It sends a BoardForecast with a (forecast-)ID, to which the downstream machine at some point must respond with a MachineReady with the same ID. Upon receiving this MachineReady, the upstream machine can assume that the product change was successful.

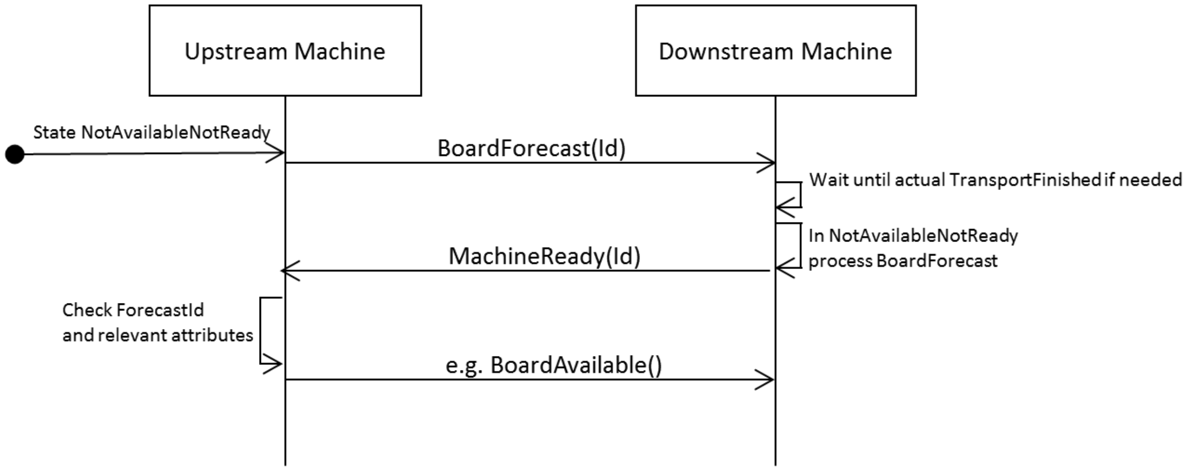

Fig. 13 Example of communication sequence for BoardForecast

Note: If starting the BoardForecast handling in the state MachineReady, the downstream machine must sent a RevokeMachineReady message (see Fig. 14).

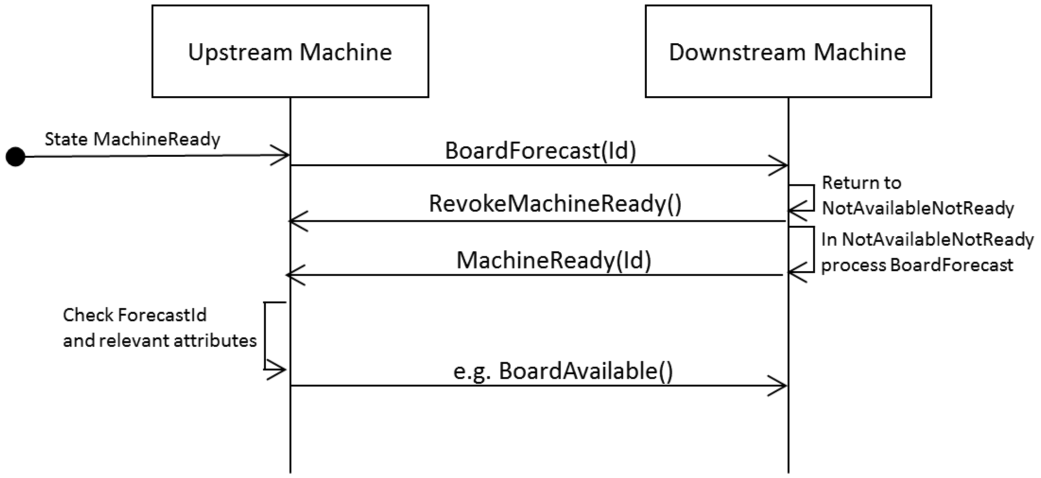

Fig. 14 Example of communication sequence for BoardForecast with RevokeMachineReady

If several BoardForecast messages (e.g. with different ProductTypeId) are sent in a short delay, the downstream machine may process only the last BoardForecast message:

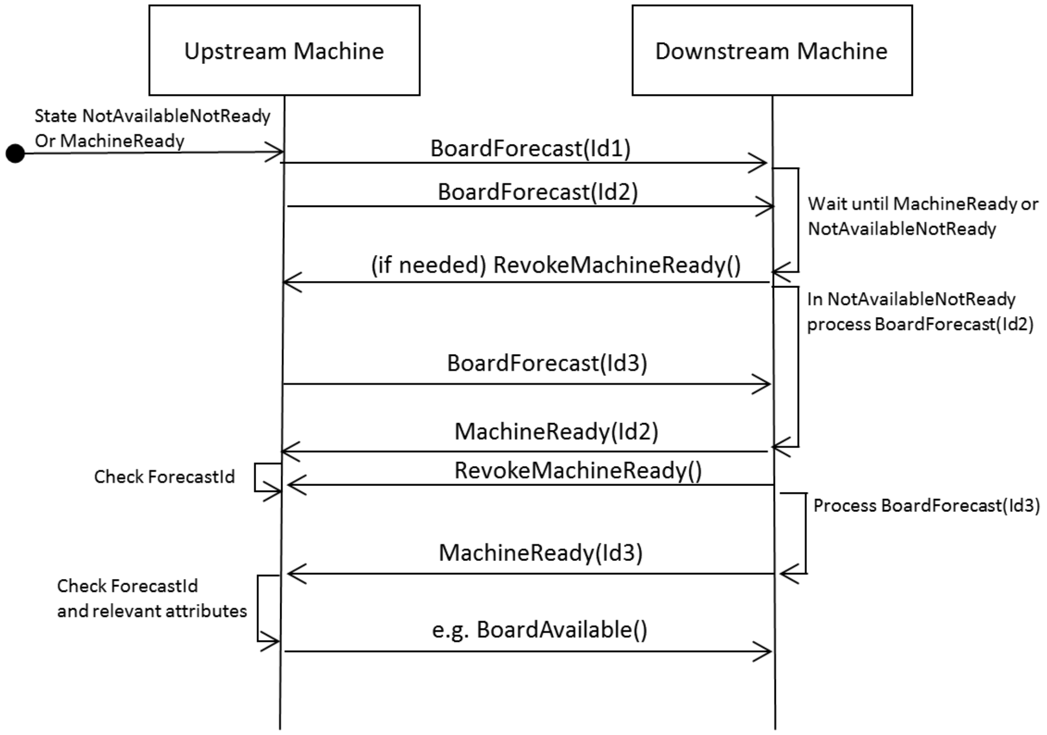
Upstream Machine
Downstream Machine
State NotAvailableNotReady Or MachineReady
BoardForecast(Id1)
BoardForecast(Id2)
Wait until MachineReady or NotAvailableNotReady
(if needed) RevokeMachineReady()
In NotAvailableNotReady process BoardForecast(Id2)
BoardForecast(Id3)
MachineReady(Id2)
Check ForecastId
RevokeMachineReady()
Process BoardForecast(Id3)
MachineReady(Id3)
Check ForecastId and relevant attributes
e.g. BoardAvailable()

Fig. 15 Example of communication sequence with several BoardForecast

Scenario 1 (error handling):
If the downstream machine cannot accept the product exchange (e.g. unknown ProductId or width is physically impossible in machine) it will respond after a RevokeMachineReady with a notification of type

"BoardForecastError". The upstream machine must then do some error handlind (e.g. ask operator if machine should retry the BoardForecast or if the operator wants to remove the board).

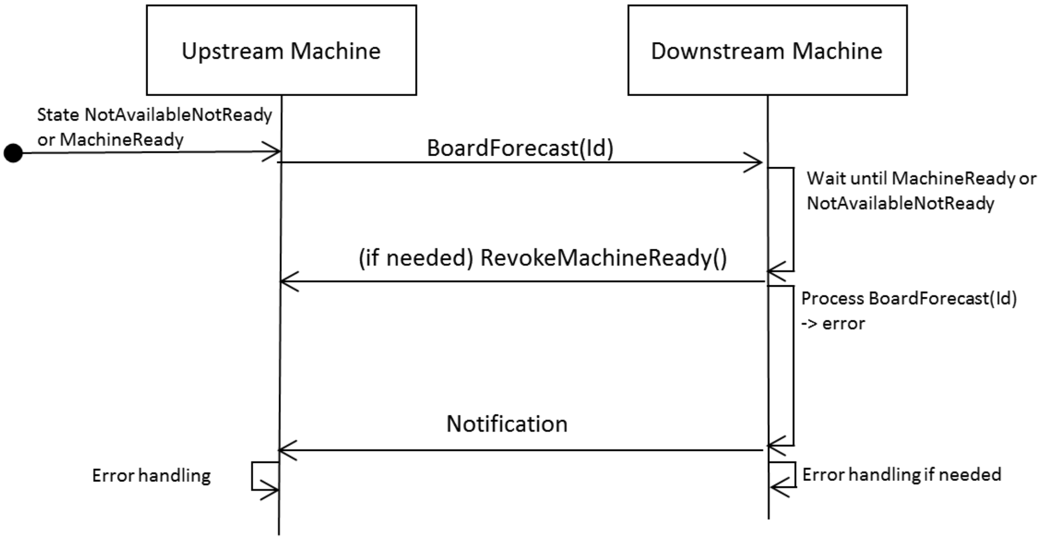
Upstream Machine
Downstream Machine
State NotAvailableNotReady or MachineReady
BoardForecast(Id)
Wait until MachineReady or NotAvailableNotReady
(if needed) RevokeMachineReady()
Process BoardForecast(Id) -> error
Notification
Error handling
Error handling if needed

Fig. 16 Example of communication sequence in case with error handling

Scenario 2:
As BoardForecast in that case usually only gives some information to the downstream machine, several BoardForecast may be sent. However, error handling or checking are not needed on the side of the downstream machine. In that scenario ForecastId will not be sent.

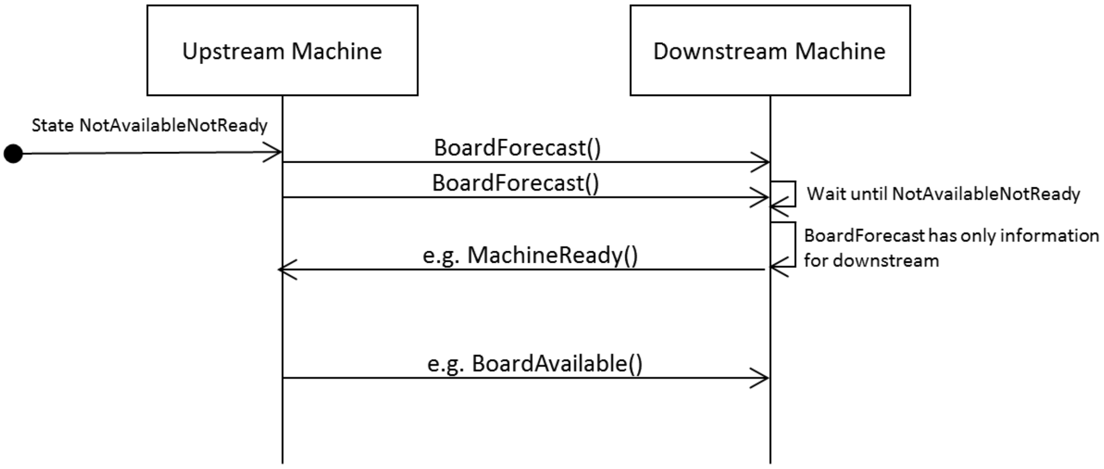
Upstream Machine
Downstream Machine
State NotAvailableNotReady
BoardForecast()
BoardForecast()
Wait until NotAvailableNotReady
BoardForecast has only information for downstream
e.g. MachineReady()
e.g. BoardAvailable()

Fig. 17 Example of communication sequence BoardForecast without product change

Note: The function of BoardForecast is optional. If FeatureBoardForecast is specified in the ServiceDescription, it must be fully supported. Otherwise it can be ignored.

2.7 Protocol states and protocol error handling

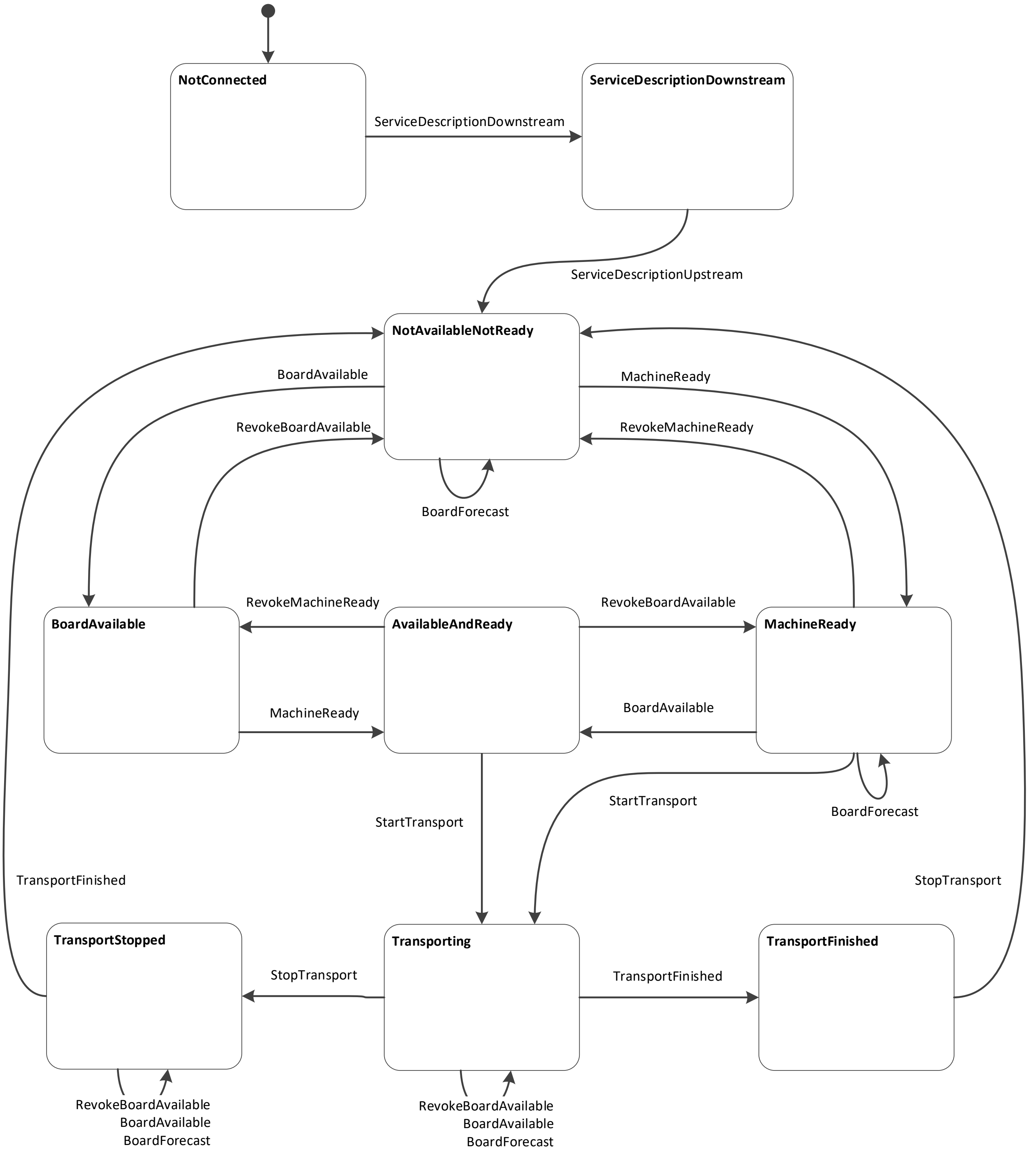

Fig. 18 Hermes interface states

Fig. 18 lists all states and transitions of a Hermes interface corresponding to the machine-to-machine (M2M) communication. The state is the comprehensive state of the interface rather than the state of one of the involved machines.

The messages may only be sent if they trigger the corresponding transition shown in the state chart. Any message, except “Notification”, “CheckAlive”, “QueryBoardInfo” and “SendBoardInfo”, which is received not triggering a transition is interpreted as a protocol error (e.g. a MachineReady message when the interface is in the state Transporting). In case of a protocol error, any running transport shall be stopped and the connection is terminated. The interface may start over with a new connection.

Note that due to race conditions, a RevokeBoardAvailable message may overlap with a StartTransport message or even a StopTransport message, so this shall not be treated as a protocol error (transition from MachineReady to Transporting and self-transitions on Transporting and TransportStopped).

2.8 Board IDs

Board individuals are identified by board IDs. These must be Globally Unique Identifiers (GUIDs) according to [ITU-T_REC_X.667], e.g. 123e4567-e89b-12d3-a456-426655440000. They are generated by the first machine in a consecutive row of machines implementing the Hermes protocol. The board ID is passed from machine to machine. If a machine in a line does not implement the Hermes protocol, the board ID is lost and a new one will be generated by the next machine implementing Hermes.

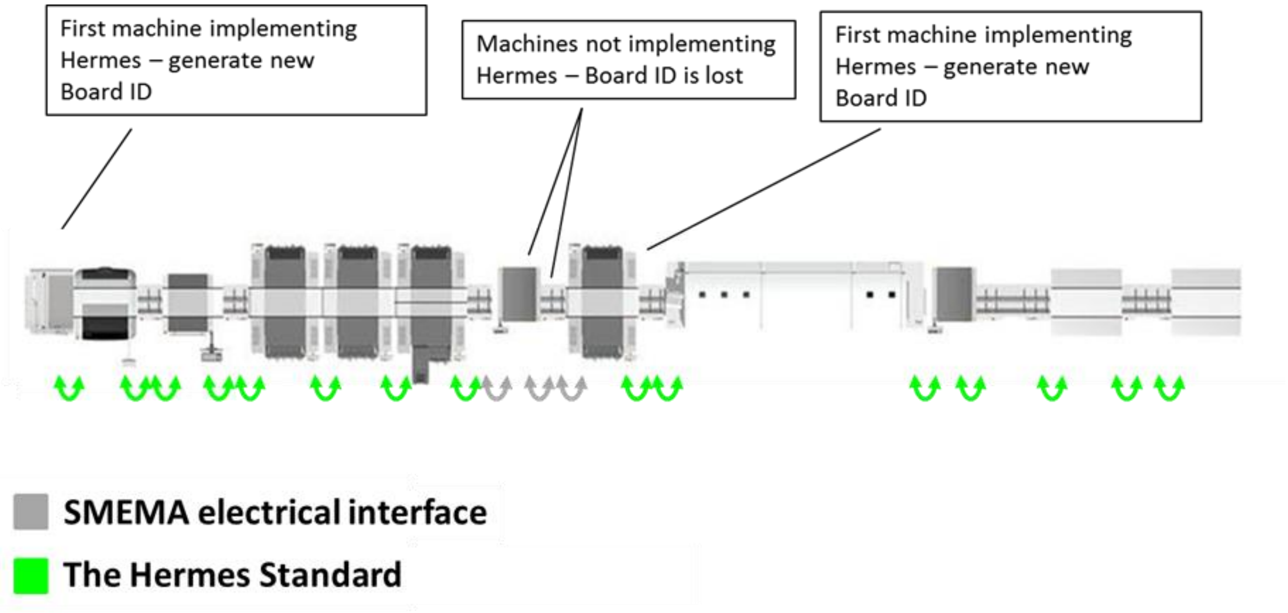
First machine implementing Hermes – generate new Board ID
Machines not implementing Hermes – Board ID is lost
First machine implementing Hermes – generate new Board ID
SMEMA electrical interface
The Hermes Standard

Fig. 19 Generation of Board IDs

3 Message definition

3.1 Message format

Messages use the Extensible Markup Language (XML) format, where at least version 1.1 of XML shall be supported [W3C_XML_1.1].

For character encoding UTF-8 has to be used (No other encoding may be specified in the XML declaration).

In the following sections of the document, for a better readable description of the XML data structures, tables are used instead of commonly used schema definitions.

Maximum size for every message is 64 kByte, i.e. 65536 bytes. For every string parameter there is either a fixed or minimum size that must be supported (individual values see tables).

In the tables, XML attributes are marked with the image “◆” and XML child nodes are marked with the image “📁”, which in turn may consist of more XML structures.

The representation of data types (e.g. floating point numbers, boolean attributes …) shall comply with the W3C XML schema recommendation [W3C_XML_Schema].

To keep upward compatibility, any message or attribute unknown by an implementation can be ignored and discarded.

3.2 Root element

Every message is enveloped by a common root element with tag <Hermes>. The root element optionally includes a timestamp attribute with the following format (based on the W3C note “Date and Time Formats” [W3C_DATE_TIME]):

```
YYYY-MM-DDThh:mm:ss.s
```

where:

```
     YYYY = four-digit year
     MM   = two-digit month (01=January, etc.)
     DD   = two-digit day of month (01 through 31)
     hh   = two digits of hour (00 through 23) (am/pm NOT allowed)
     mm   = two digits of minute (00 through 59)
     ss   = two digits of second (00 through 59)
     s    = one or more digits representing a decimal fraction of a second
```

The decimal fraction of the second shall be given with 3 digit precision.

The timestamp is optional and intended for diagnostic purposes only.

An example for a CheckAlive message would be:

```
<Hermes TimeStamp=”2017-07-16T19:20:30.452“>
  <CheckAlive />
</Hermes>
```

A machine is not required to emit a precise timestamp, since this attribute is intended mainly for debugging purposes.

Recommendation: Synchronize all machines in a line to a common time source. For machines that do not have an absolute time source, the year should be set to "0000". At any rate, the timestamp should be monotonic.

3.3 CheckAlive

The CheckAlive message is used to detect connection losses. It therefore does not have to transport data and can be ignored by the receiver. Accordingly there is no response.

However, if a machine supports the FeatureCheckAliveResponse, it must answer CheckAlive messages with Type set to 1 with a CheckAlive message with Type set to 2 and the same Id as the received CheckAlive message.

CheckAlive	Type	Range	Optional	Description
◆Type	int	1..2	yes	Ping / Pong message type
◆Id	string	any string (minimum supported length: 80 bytes)	yes	Idenfier of the message

Type may be one of the following values:

1 Ping: CheckAlive request

2 Pong:CheckAlive response

The machine sending CheckAlive message with Type set to 1 chooses a unique for Id (e.g. GUID or time stamp). The machine responding with CheckAlive message with Type set to 2 has to answer using the same Id.

3.4 ServiceDescription

The ServiceDescription message is sent by both machines after a connection is established. The downstream machine sends its ServiceDescription first whereupon the upstream machine answers by sending its own ServiceDescription.

ServiceDescription	Type	Range	Optional	Description
MachineId	string	any string (minimum supported length: 80 bytes)	no	ID/name of the sending machine for identifying it in a Hermes enabled production line.
LaneId	int	1 .. n	no	The sending machine's lane to which this connection is relating to. Lanes are enumerated looking downstream from right to left beginning with 1
InterfaceId	string	any string (minimum supported length: 80 bytes)	yes	The ID of the sending machine's transportation interface to which this connection is relating to.
Version	string	xxx.yyy (7 bytes)	no	The implemented interface version of the machine
SupportedFeatures	Feature []		no	List of supported features (empty for version 1.0)

Feature	Type	Range	Optional	Description
FeatureCheckAliveResponse	FeatureCheckAliveResponse		yes	Indication of CheckAliveResponse function implementation
FeatureBoardForecast	FeatureBoardForecast		yes	In the upstream role: Machine emits BoardForecast messages
FeatureQueryBoardInfo	FeatureQueryBoardInfo		yes	Indication of QueryBoardInfo function implementation
FeatureSendBoardInfo	FeatureSendBoardInfo		yes	Indication of SendBoardInfo function implementation

xxx.yyy must match the regular expression

```
[1-9][0-9]{0,2}\.[0-9]{1,3}
```

The features specified in version 1.0 of this protocol have to be provided by any implementation and thus are not listed in the SupportedFeatures list of the ServiceDescription explicitly.

3.5 Notification

The Notification message is sent by both machines before a connection is terminated, e.g. after protocol errors or before shutdown. It could also be used for general notification purposes.

Notification	Type	Range	Optional	Description
NotificationCode	int	1 .. n	no	A notification code of the list below. Notification codes above 1000 are not defined by this protocol and may be used by the application
Severity	int	1 .. 4	no	A severity of the list below
Description	string	any string (minimum supported length: 254 bytes)	no	An English textual description of the notification.

The following NotificationCodes are defined:

1 Protocol error (invalid transition in the state machine, see section 2.6)
2 Connection refused because of an established connection
3 Connection reset because of changed configuration
4 Configuration error
5 Machine shutdown
6 BoardForecast error

Possible values for Severity:

1 Fatal error
2 Error
3 Warning
4 Info

3.6 BoardAvailable

The BoardAvailable message is sent to the downstream machine to indicate the readiness of the upstream machine to handover a PCB. When an optional attribute is received from an upstream machine, then it must be passed on (possibly altered) to the next downstream machine.

BoardAvailable	Type	Range	Optional	Description
BoardId	string	GUID (36 bytes)	no	Indicating the ID of the available board
BoardIdCreatedBy	string	non-empty string (minimum supported length: 80 bytes)	no	MachineId of the machine which created the BoardId (the first machine in a consecutive row of machines implementing this protocol). The MachineId is part of the Hermes configuration.
FailedBoard	int	0 .. 2	no	A value of the list below
ProductTypeId	string	any string (minimum supported length: 254 bytes)	yes	Identifies a collection of PCBs sharing common properties
FlippedBoard	int	0 .. 2	no	A value of the list below
TopBarcode	string	any string (minimum supported length: 254 bytes)	yes	The barcode of the top side of the PCB
BottomBarcode	string	any string (minimum supported length: 254 bytes)	yes	The barcode of the bottom side of the PCB
Length	float	positive numbers	yes	The length of the PCB in millimeter.
Width	float	positive numbers	yes	The width of the PCB in millimeter.
Thickness	float	positive numbers	yes	The thickness of the PCB in millimeter.
ConveyorSpeed	float	positive numbers	yes	The conveyor speed preferred by the upstream machine in millimeter per second
TopClearanceHeight	float	positive numbers	yes	The clearance height for the top side of the PCB in millimeter.
BottomClearanceHeight	float	positive numbers	yes	The clearance height for the bottom side of the PCB in millimeter.
Weight	float	positive numbers	yes	The weight of the PCB in grams.

GUID must match the regular expression

```
[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}
```

FailedBoard may be one of the following values:

0	Board of unknown quality available
1	Good board available
2	Failed board available

FlippedBoard may be one of the following values:

0	Side up is unknown
1	Board top side is up
2	Board bottom side is up

If FlippedBoard is 2 (board bottom side is up) then TopBarcode is facing downwards and BottomBarcode is facing upwards. Same applies for TopClearanceHeight and BottomClearanceHeight.
The definition of board bottom and board top side is outside of the scope of The Hermes Standard and left to the customer.

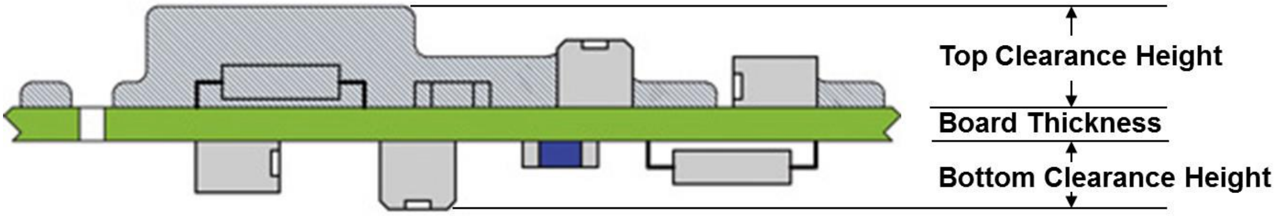

Fig. 20 Explanation for top and bottom clearance height

3.7 RevokeBoardAvailable

With the RevokeBoardAvailable message, the upstream machine signals that it is not ready anymore to handover a PCB.

RevokeBoardAvailable	Type	Range	Optional	Description

3.8 MachineReady

The MachineReady message is sent to the upstream machine to indicate the readiness of the downstream machine to accept a PCB.

MachineReady	Type	Range	Optional	Description
FailedBoard	int	0 .. 2	no	A value of the list below
ForecastId	string	any string (minimum supported length: 80 bytes)	yes/no	If responding to a BoardForecast message mandatory. It indicates the ID of the original BoardForecast message.
BoardId	string	GUID (36 bytes)	yes	Indicates the ID of the board that will be handed over as next. In case of product change this attribute will not be sent.
ProductTypeId	string	any string (minimum supported length: 254 bytes)	yes	Identifies a collection of PCBs sharing common properties
FlippedBoard	int	0 .. 2	yes	A value of the list below
Length	float	positive numbers	yes	The length of the PCB in millimeter.
Width	float	positive numbers	yes	The width of the PCB in millimeter.
Thickness	float	positive numbers	yes	The thickness of the PCB in millimeter.
ConveyorSpeed	float	positive numbers	yes	The conveyor speed used by the upstream machine in millimeter per second
TopClearanceHeight	float	positive numbers	yes	The clearance height for the top side of the PCB in millimeter.
BottomClearanceHeight	float	positive numbers	yes	The clearance height for the bottom side of the PCB in millimeter.
Weight	float	positive numbers	yes	The weight of the PCB in grams.

FlippedBoard may be one of the following values:

0	Side up is unknown
1	Board top side is up
2	Board bottom side is up

FailedBoard may be one of the following values:

0	Ready to accept any board
1	Ready to accept good boards.
2	Ready to accept failed boards

3.9 RevokeMachineReady

With the RevokeMachineReady message, the downstream machine signals that it is not ready anymore to accept a PCB.

RevokeMachineReady	Type	Range	Optional	Description

3.10 StartTransport

The StartTransport message is sent to the upstream machine to initiate the PCB handover process. There is no response to this message.

StartTransport	Type	Range	Optional	Description
◆BoardId	string	GUID (36 bytes)	no	The ID of the board for which the transport shall be started.
◆ConveyorSpeed	float	positive numbers	yes	Optional parameter indicating the selected conveyor speed for the handover in millimeter per second

The downstream machine is responsible for selecting the actual conveyor speed according to the preferred conveyor speed sent in the BoardAvailable message. In general the highest possible speed supported by both machines will be selected.
If a StartTransport message is received for a BoardId which is not the one received with the last BoardAvailable message, the transport shall be canceled. This case is not to be treated as a protocol error.

3.11 StopTransport

The StopTransport message is sent by the downstream machine after it has finished the transport.

StopTransport	Type	Range	Optional	Description
◆TransferState	int	1 .. 3	no	See list below for possible values
◆BoardId	string	GUID (36 bytes)	no	The ID of the board to which the message relates to

Transfer states:

1. NotStarted: The PCB never left and hence is fully inside the upstream machine.
2. Incomplete: The transfer was cancelled in progress.
3. Complete: The transfer ended successfully.

If the BoardId does not match the one from StartTransport, this shall be treated as a protocol error. Therefore, the connection would need to be re-established.

3.12 TransportFinished

The TransportFinished message is sent by the upstream machine after it finished the transport.

TransportFinished	Type	Range	Optional	Description
TransferState	int	1 .. 3	no	See list below for possible values
BoardId	string	GUID (36 bytes)	no	The ID of the board to which the message relates to

Transfer states:

1. NotStarted: The PCB never left and hence is fully inside the upstream machine.
2. Incomplete: The transfer was cancelled in progress.
3. Complete: The transfer ended successfully.

If the BoardId does not match the one from StartTransport, this shall be treated as a protocol error. Therefore, the connection would need to be re-established.

3.13 SetConfiguration

The SetConfiguration message is sent by an engineering station to configure the Hermes interfaces of a machine. If the sent configuration is not accepted, the machine is expected to send a Notification message (see section 3.5).

SetConfiguration	Type	Range/ Multiplicity	Optional	Description
MachineId	string	any string (minimum supported length: 80 bytes)	no	ID/name of this machine for identifying it in a Hermes enabled production line.
UpstreamConfigurations	UpstreamConfiguration []	0 .. n	no	Configuration for upstream lanes
DownstreamConfigurations	DownstreamConfiguration []	0 .. n	no	Configuration for downstream lanes

UpstreamConfiguration	Type	Range/ Multiplicity	Optional	Description
UpstreamLaneId	int	1 .. n	no	The lane on the upstream side Lanes are enumerated looking downstream from right to left beginning with 1
UpstreamInterfaceId	string	any string (minimum supported length: 80 bytes)	yes	The ID of the transportation interface on the upstream side
HostAddress	string	valid IP address or hostname (minimum supported length: 254 bytes)	no	The IP address or hostname of the upstream machine for this lane and transportation interface
Port	int	0 .. 65535	no	Port number on which connections shall be established

DownstreamConfiguration	Type	Range/ Multiplicity	Optional	Description
DownstreamLaneId	int	1 .. n	no	The lane on the downstream side Lanes are enumerated looking downstream from right to left beginning with 1
DownstreamInterfaceId	string	any string (minimum supported length: 80 bytes)	yes	The ID of the transportation interface on the downstream side
ClientAddress	string	valid IP address or hostname (minimum supported length: 254 bytes)	yes	The IP address or hostname of the downstream machine for this lane and transportation interface. If not specified, then connections from any IP address are accepted.
Port	int	0 .. 65535	no	Port number on which the server shall accept connections for this lane

All connections where the machine is acting as board provider are stored in DownstreamConfigurations. All connections where the machine is acting as board receiver are stored in UpstreamConfigurations. These are independent of the board transport direction of the SMT line.

It is up to the user to keep MachineIds unique.

3.14 GetConfiguration

The GetConfiguration message is sent by an engineering station to read out the current configuration of the Hermes interfaces of a machine. The machine is expected to answer with a CurrentConfiguration message.

GetConfiguration	Type	Range/ Multiplicity	Optional	Description

3.15 CurrentConfiguration

The CurrentConfiguration message is sent by a machine in response to the GetConfiguration message.

CurrentConfiguration	Type	Range/ Multiplicity	Optional	Description
◆MachineId	string	any string (minimum supported length: 80 bytes)	yes	ID/name of this machine for identifying it in a Hermes enabled production line.
📁UpstreamConfigurations	UpstreamConfiguration []	0 .. n	no	Configuration of upstream lanes
📁DownstreamConfigurations	DownstreamConfiguration []	0 .. n	no	Configuration of downstream lanes

For the definition of UpstreamConfiguration and DownstreamConfiguration see section 3.13.
If no MachineId has been configured yet, the CurrentConfiguration message does not contain the attribute MachineId.

3.16 BoardForecast

The BoardForecast message is sent to the downstream machine to indicate some changes / command execution are needed or to give a pre-information about the next board but a PCB is not jet available. If the ForecastId field is set then the downstream machine must at some point respond with a MachineReady carrying the same ForecastId. If needed downstream machine must send a RevokeMachineReady message first. If the forecasted product is not fitting to the downstream machine, then it must respond with a Notification of type “BoardForecastError”.

Note: The function of BoardForecast is optional. If FeatureBoardForecast is specified in the ServiceDescription, it must be fully supported. Otherwise it can be ignored.

BoardForecast	Type	Range	Optio-nal	Description
ForecastId	string	any string (minimum supported length: 80 bytes)	yes	Indicating the ID of forecast message. The ID must be unambiguous and e.g. can be a timetamp or a GUID.
TimeUntilAvailable	float	positive numbers	yes	Number of seconds until a board may be available at downstream machine
BoardId	string	GUID (36 bytes)	yes	Indicating the ID of the board that will be handed over as next. e.g. in case of product change this field will not be sent
BoardIdCreatedBy	string	any string (minimum supported length: 80 bytes)	yes	MachineId of the machine which created the BoardId.
FailedBoard	int	0 .. 2	no	A value of the list below
ProductTypeId	string	any string (minimum supported length: 254 bytes)	yes	Identifies a collection of PCBs sharing common properties
FlippedBoard	int	0 .. 2	no	A value of the list below
TopBarcode	string	any string (minimum supported length: 254 bytes)	yes	The barcode of the top side of the next PCB
BottomBarcode	string	any string (minimum supported length: 254 bytes)	yes	The barcode of the bottom side of the next PCB
Length	float	positive numbers	yes	The length of the PCB in millimeter.
Width	float	positive numbers	yes	The width of the PCB in millimeter.
Thickness	float	positive numbers	yes	The thickness of the PCB in millimeter.
ConveyorSpeed	float	positive numbers	yes	The conveyor speed preferred by the upstream machine in millimeter per second
TopClearanceHeight	float	positive numbers	yes	The clearance height for the top side of the PCB in millimeter.

◆BottomClearanceHeight	float	positive numbers	yes	The clearance height for the bottom side of the PCB in millimeter.
◆Weight	float	positive numbers	yes	The weight of the PCB in grams.

The attributes definition are identical to the BoardAvailable message.

FailedBoard may be one of the following values:

0 Ready to accept any board
1 Ready to accept good boards.
2 Ready to accept failed boards

FlippedBoard may be one of the following values:

0 Side up is unknown
1 Board top side is up
2 Board bottom side is up

3.17 QueryBoardInfo

The QueryBoardInfo message is sent to the upstream machine to request information about one of the last boards (see section 4.1.3).

Note: The function of QueryBoardInfo is optional. If FeatureQueryBoardInfo is specified in the ServiceDescription, it must be fully supported. Otherwise it can be ignored.

QueryBoardInfo	Type	Range	Optional	Description
◆TopBarcode	String	any string (minimum supported length: 254 bytes)	yes / no	The barcode of the top side of the PCB. Either top or bottom barcode must be specified.
◆BottomBarcode	String	any string (minimum supported length: 254 bytes)	yes / no	The barcode of the bottom side of the PCB. Either top or bottom barcode must be specified.

3.18 SendBoardInfo

The SendBoardInfo message is sent to the downstream machine as response of a received QueryBoardInfo message to transfer stored information about one of the last boards (see section 4.1.3). If the upstream machine cannot find any board information it will nevertheless send the SendBoardInfo message without the BoardId and BoardCreatedBy attributes.

Machines supporting the feature FeatureSendBoardInfo shall be able to store and supply upon request the info of at least the last 50 handled boards.

Note: The function of SendBoardInfo is optional. If FeatureSendBoardInfo is specified in the ServiceDescription, it must be fully supported. Otherwise it can be ignored.

SendBoardInfo	Type	Range	Optional	Description
◆BoardId	string	GUID (36 bytes)	yes / no	The ID of the board which data has been requested. This attribute will not be sent if the board information has not been found.
◆BoardIdCreatedBy	string	non-empty string (minimum supported length: 80 bytes)	yes / no	MachineId of the machine which created the BoardId. This attribute will not be sent if the board information has not been found.
◆FailedBoard	Int	0 .. 2	Yes/no	A value of the list below. This attribute will not be sent if the board information has not been found.
◆ProductTypeId	string	any string (minimum supported length: 254 bytes)	yes	Identifies a collection of PCBs sharing common properties
◆FlippedBoard	Int	0 .. 2	Yes/no	A value of the list below. This attribute will not be sent if the board information has not been found.
◆TopBarcode	string	any string (minimum supported length: 254 bytes)	yes/no	The barcode of the top side of the next PCB. This attribute is mandatory if it has been in the QueryBoardInfo message.
◆BottomBarcode	string	any string (minimum supported length: 254 bytes)	yes/no	The barcode of the bottom side of the next PCB. This attribute is mandatory if it has been in the QueryBoardInfo message.
◆ Length	float	positive numbers	yes	The length of the PCB in millimeter.
◆Width	float	positive numbers	yes	The width of the PCB in millimeter.
◆Thickness	float	positive numbers	yes	The thickness of the PCB in millimeter.
◆ TopClearanceHeight	float	positive numbers	yes	The clearance height for the top side of the PCB in millimeter.
◆ BottomClearanceHeight	float	positive numbers	yes	The clearance height for the bottom side of the PCB in millimeter.

The attributes definition are identical to the BoardAvailable message.

FailedBoard may be one of the following values:

0	Board of unknown quality available
1	Good board available
2	Failed board available

FlippedBoard may be one of the following values:

0	Side up is unknown
1	Board top side is up
2	Board bottom side is up

4 Appendix

4.1 Special scenarios

The following sections are not part of the Hermes protocol specification. In fact they shall show the application of this protocol in some special scenarios.

4.1.1 Board tracking when board is torn out from the line

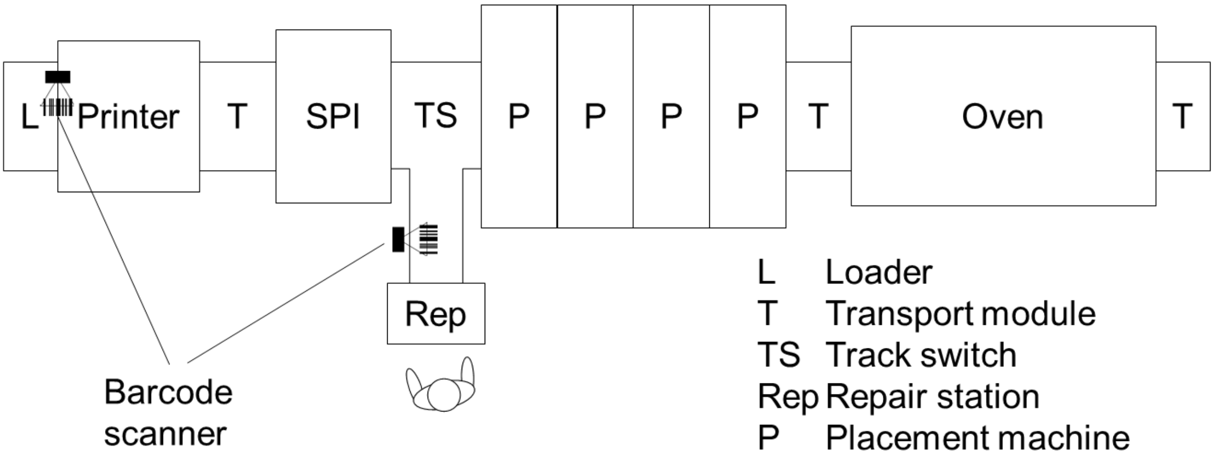

Fig. 21 Line setup with barcode readers and repair station

In this scenario, a repair station is placed behind the SPI. PCBs failing the solder paste inspection are torn out by the track switch and are presented to an operator at the repair station. The operator may take out the PCB for rework and re-insert it later independent of the PCB sequence.

By removing the PCB from the line, the link between the PCB and the barcode respectively the BoardId is lost. So when the PCB is re-inserted, different approaches are possible to re-establish the tracking of the PCB:

a) Create a new Hermes BoardId, read the barcode and report the from now on used tracking information. The tracking information can be merged later by an external system (e.g. MES) using the barcodes.
b) Read the barcode first and request the corresponding Hermes BoardId from the external system (e.g. MES). The tracking can be continued using the primarily assigned Hermes BoardId.
c) The machine blocks the production of the re-inserted PCB until the operator scans the barcode using a mobile barcode scanner or enters it manually and specifies which board side is currently up. Then the original Hermes BoardId and all the needed information is requested from the upstream machine via the QueryBoardInfo message. The downstream machine sends the QueryBoardInfo with the top or bottom barcode and gets back a SendBoardInfo message from the upstream machine including BoardId. If information for that barcode was not available then the attribute BoardId will be omitted.

d) Simplest but most unsecure approach: The repair station prompts the operator to confirm that the inserted PCB is the same which was last removed from the station

Option a and b are realized with an MES system. Option c and d enables the re-insertion of boards directly at the machine without having an MES system for that line (relying only on functions of The Hermes Standard).

4.1.2 Board tracking when board is temporarily removed from the line

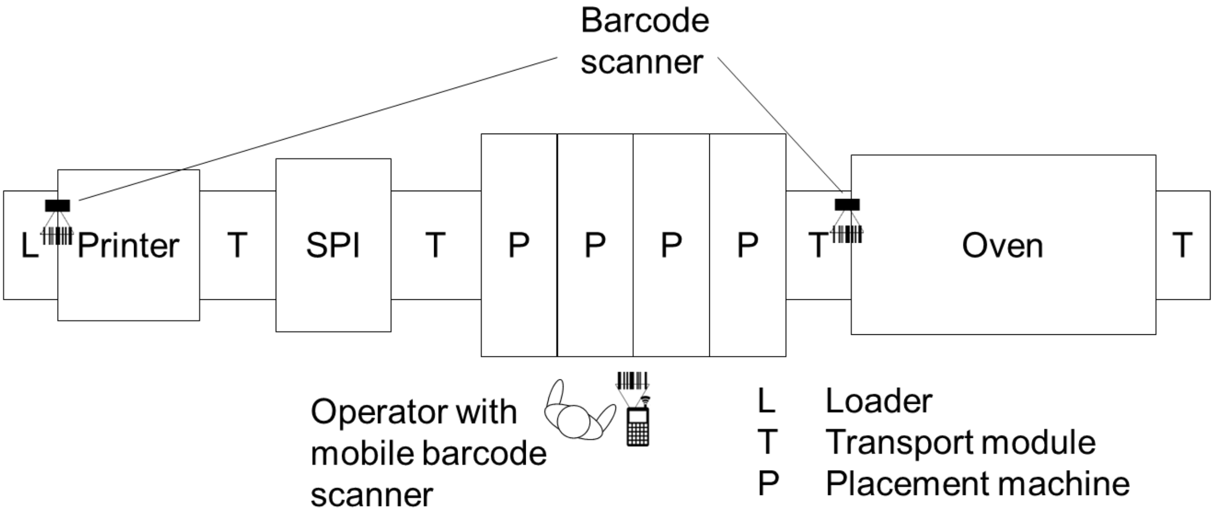

Fig. 22 Line setup with fixed and mobile barcode readers

In this scenario, the operator removes a PCB for inspection from one of the placement machines. The line continues producing PCBs. At some later point in time, the PCB is re-inserted to complete its production.

By removing the PCB from the line, the link between the PCB and the barcode respectively the BoardId is lost. As in the scenario above, different approaches are possible to re-establish the tracking of the PCB:

a) The machine blocks the production of the re-inserted PCB until the operator scans the barcode using a mobile barcode scanner or enters it manually. Then either the original Hermes BoardId is requested from an external system (e.g. MES) using the barcode or a new Hermes BoardId is created and the tracking information is merged by the external system.
b) The machine blocks the production of the re-inserted PCB until the operator scans the barcode using a mobile barcode scanner or enters it manually and specifies which board side is currently up. Then the original Hermes BoardId and all the needed information is requested from the upstream machine via the QueryBoardInfo message. The downstream machine sends the QueryBoardInfo with the top or bottom barcode and gets back a SendBoardInfo message from the upstream machine including BoardId. If information for that barcode was not available then the attribute BoardId will be omitted.

c) A new Hermes BoardId is created and production is continued without barcode. At the next barcode reader in the line, the barcode information is complemented to the Hermes BoardId. An external system can later merge all the collected tracking information.

Option a and c are realized with an MES system. Option b enables the re-insertion of boards directly at the machine without having an MES system for that line (relying only on functions of The Hermes Standard).

4.1.3 Board tracking when board was transferred without data

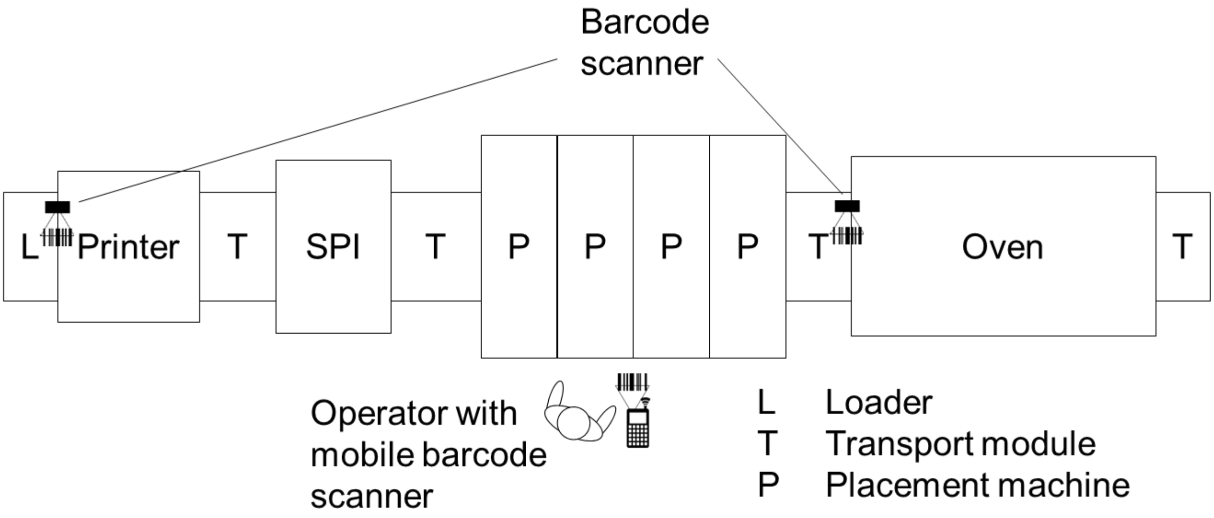

Fig. 23 Line setup with fixed and mobile barcode readers

In this scenario, one of the machines (e.g. a soldering reflow machine) cannot physically stop the transport of the PCB at the end of the machine. So boards may pile up if the next machine is not able to process the boards.
In that case the operator will temporaly remove the boards from the line and try to reinsert those at the same place a bit later on.

By removing a PCB from the line, the link between the PCB, the BoardId and other information (width, length, …) is lost. As in the scenario above, different approaches are possible to re-establish the tracking of the PCB:

a) The machine blocks the production of the re-inserted PCB until the operator scans the barcode using a mobile barcode scanner or enters it manually. Then either the original Hermes BoardId is requested from an external system (e.g. MES) using the barcode or a new Hermes BoardId is created and the tracking information is merged by the external system.
b) The machine blocks the production of the re-inserted PCB until the operator scans the barcode using a mobile barcode scanner or enters it manually and specifies which board side is currently up.

Then the original Hermes BoardId and all the needed information is requested from the upstream machine, that could not stop the PCB, via the QueryBoardInfo message. The downstream machine sends the QueryBoardInfo with the top or bottom barcode and gets back a SendBoardInfo message from the upstream machine including BoardId. If information for that barcode was not available then the attribute BoardId will be omitted.

c) A new Hermes BoardId is created and production is continued without barcode. Information will not be available for the next machine. At the next barcode reader in the line, the barcode information is added to the Hermes data. An external system can later merge all the collected tracking information (if needed).

Option a and c are realized with an MES system. Option b enables the re-insertion of boards directly at the next machine without having an MES system for that line (relying only on functions of The Hermes Standard).

4.2 Glossary, abbreviations

GUID	Globally Unique Identifier
ID	Identifier
IP	Internet Protocol
ISO/OSI	International Organization for Standardization/Open System Interconnection
M2M	Machine-to-Machine
MES	Manufacturing Execution System
PCB	Printed Circuit Board
SMEMA	Surface Mount Equipment Manufacturers Association
SMT	Surface-Mount Technology
SPI	Solder Paste Inspection
TCP	Transmission Control Protocol
XML	Extensible Markup Language

4.3 References

[IPC_SMEMA_9851] IPC-SMEMA-9851 Mechanical Equipment Interface Standard

[ISO_7498-1] ISO/IEC IS 7498-1: Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. 1996

[IETF_RFC_791] Internet Engineering Task Force: RFC791: Internet Protocol. September 1981

[IETF_RFC_2460] Internet Engineering Task Force: RFC791: Internet Protocol, Version 6 (IPv6). September 1998

[IETF_RFC_793] Internet Engineering Task Force: RFC793: Transmission Control Protocol. September 1981

[ITU-T_REC_X.667] International Standard "Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components

[SemVer_2.0.0] Tom Preston-Werner: Semantic Versioning 2.0.0. (Internet: https://semver.org/spec/v2.0.0.html, last access: 23. April 2018)

[W3C_XML_1.1] Extensible Markup Language (XML) 1.1 (Second Edition) - W3C Recommendation 16. August 2006, edited in place 29. September 2006

[W3C_DATE_TIME] Date and Time Formats - W3C Recommendation 15. September 1997

[W3C_XML_Schema] XML Schema Part 2: Datatypes Second Edition - W3C Recommendation 28. October 2004

4.4 History

Version	Date	Author	Change
1.0	03/23/17	The Hermes Standard Initiative	Initial Version
1.0, Rev 1	11/13/17	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Add Top and Bottom clearance height attribute to Board Available message • When already connected to a downstream machine, reject new connection attempts • Specify the BoardId to be a true globally unique identifier (GUID/UUID) • Remove BoardIdCreatedBy from Start-Transport, StopTransport, TransportFinished
1.0.2	04/23/18	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Application of Semantic Versioning • Define minimum requirements for strings
1.1	04/23/18	The Hermes Standard Initiative	Incorporation of changes agreed in initiative meeting • Adding InterfaceId to the configuration • Add weight attribute to BoardAvailable message • CheckAlive Response • BoardForecast • Reinsert Board