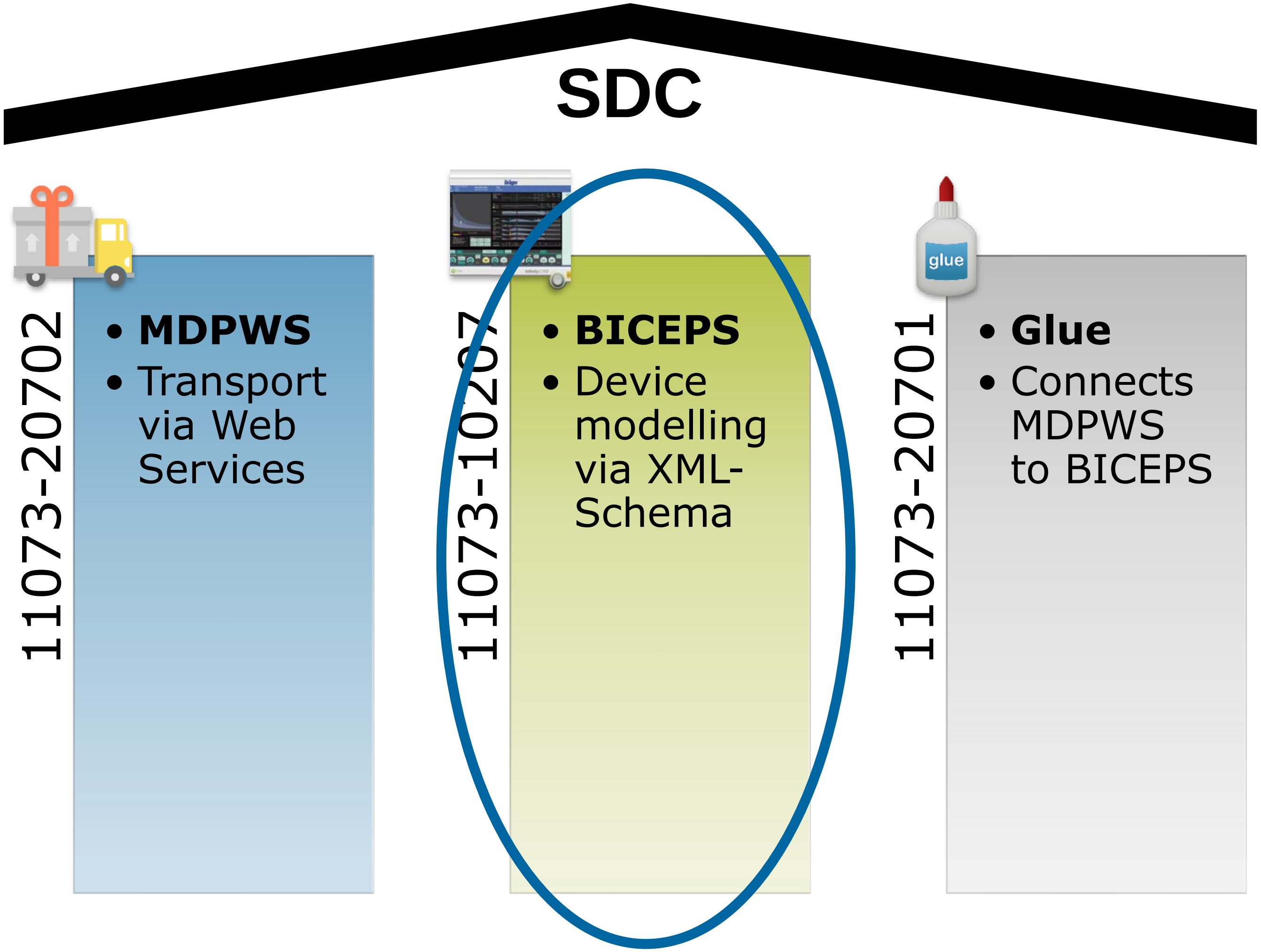
BICEPS
Remote Control

Revision 1,
2018-10-11

IEEE

Advancing Technology
for Humanity

Orientation



SDC

11073-20702

- **MDPWS**
- Transport via Web Services

11073-10207

- **BICEPS**
- Device modelling via XML-Schema

11073-20701

- **Glue**
- Connects MDPWS to BICEPS

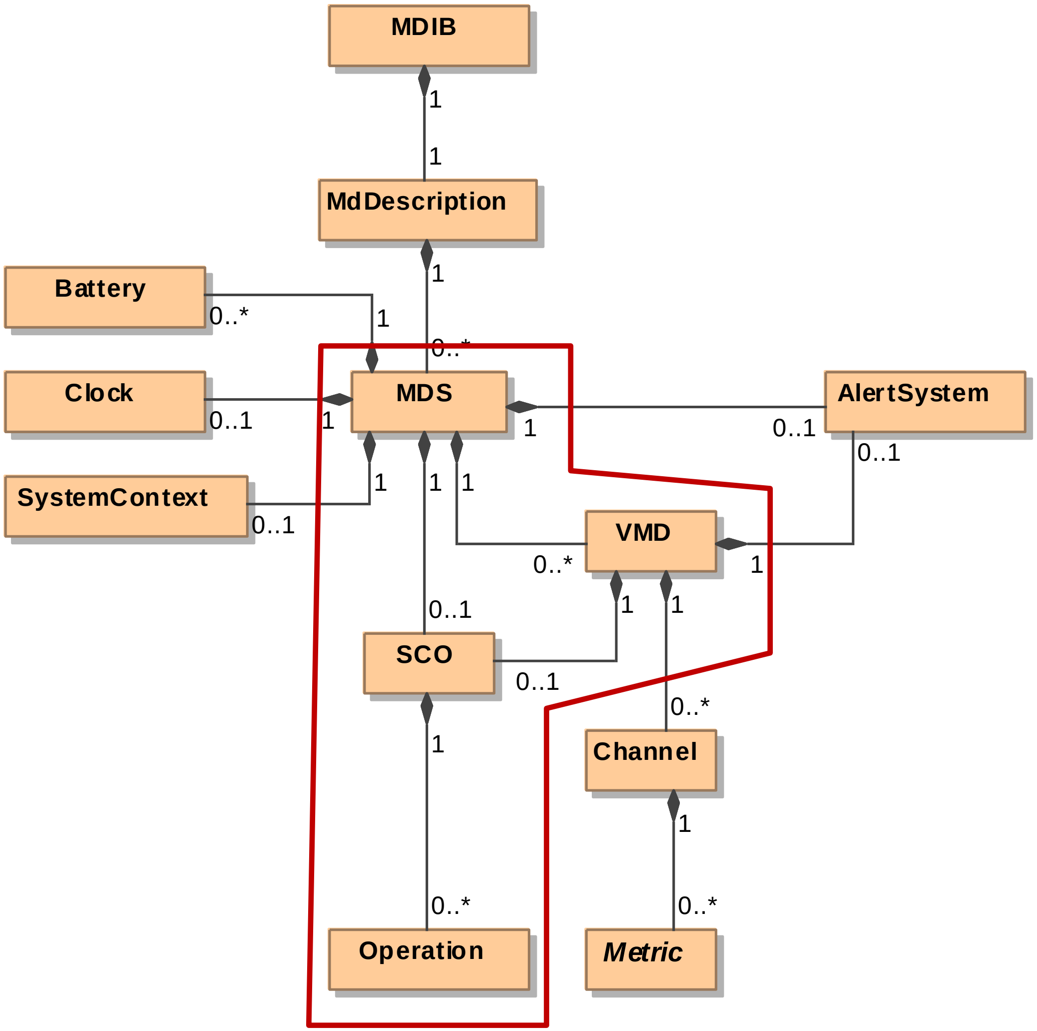
DEFINITION

Definition
Service Control Object

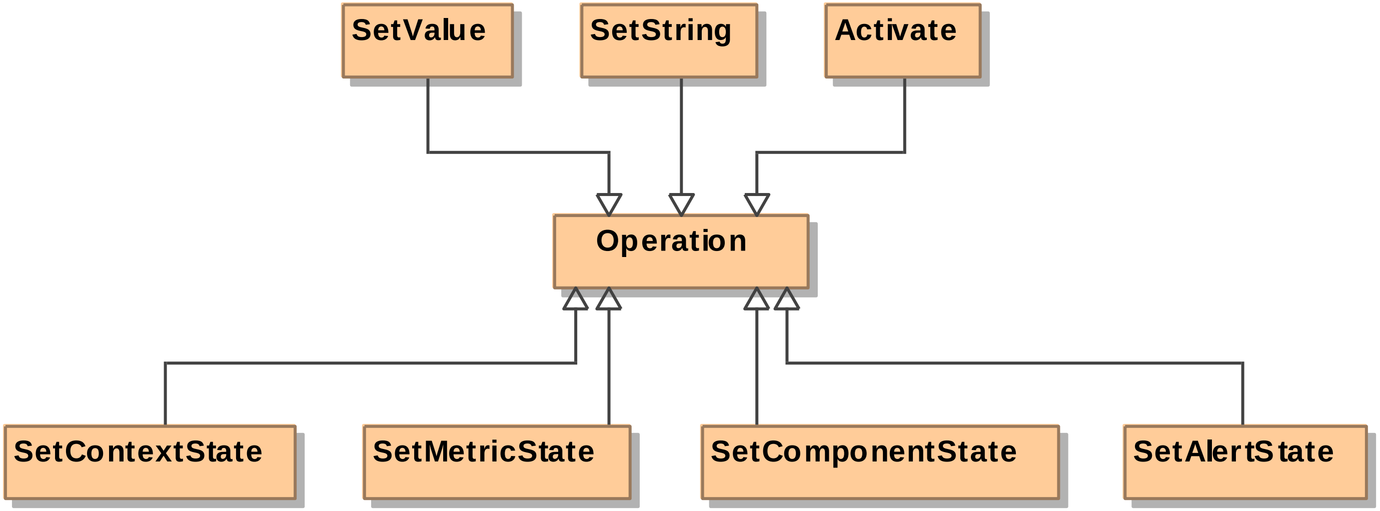
The Service Control Object (SCO) defines a means to invoke procedures in order to allow for remote control of medical devices.

CLASS DIAGRAM

Class diagram
Placement in MDIB



Class diagram
Derived types



Class diagram
Description of operations I

- SetValue: accepts a number to set *any* numeric value of a state
- SetString: accepts a string to set *any* value of a state
- Activate: invokes arbitrary parametrized action
- SetContextState: insert of modify a context state
- SetMetricState: modify a metric state
- SetComponentState: modify a component state
- SetAlertState: modify an alert state

Class diagram

Description of operations II (notes)

The *any* from SetValue and SetString refers to a predefined value that is derived from the target handle and type definition.

BICEPS allows modification of all state attributes, though it might not be meaningful in every case.

SetValue and SetString are intended to be preferred over any set-state operation if technically possible, i.e., only one value is going to be changed.

PROPERTIES

Properties

pm:ScoDescriptor

- The SCO provides a list of operations, where the operation element with a lower list index has a higher clinical relevance than any entry with a higher list index.
- The clinical relevance is defined by the service provider, which may reorder the list at any time.

Properties

pm:ScoState

OperationGroup [list]

- Groups of operations in order to allow clinical grouping and prioritization of operations
  - Each group has an operation mode and a list of operations

InvocationRequested [list]

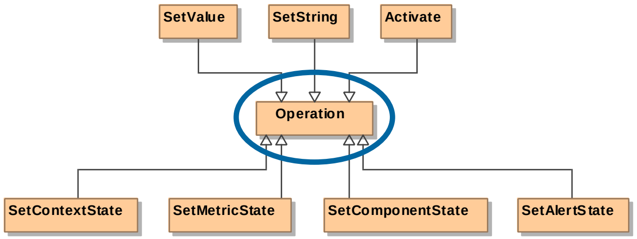
- A prioritized list of operations that are requested to be invoked by a service consumer

InvocationRequired [list]

- A prioritized list of operations that are required to be invoked by a service consumer

Properties

pm:AbstractOperationDescriptor

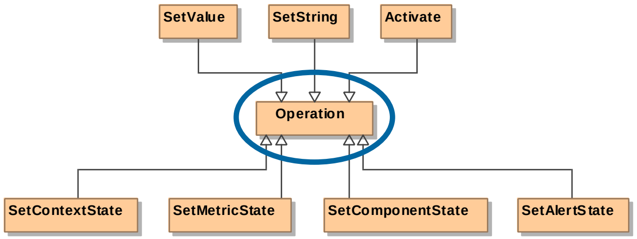


OperationTarget

- Handle reference of the MDIB object an operation is targeted to. In case of a single state this is the handle of the descriptor. In case of a multi state this is the handle of one of the state instances. If the operation is not directly connected to a specific object, the operation target shall be the MDS descriptor handle under which the operation is located.

Properties

pm:AbstractOperationState

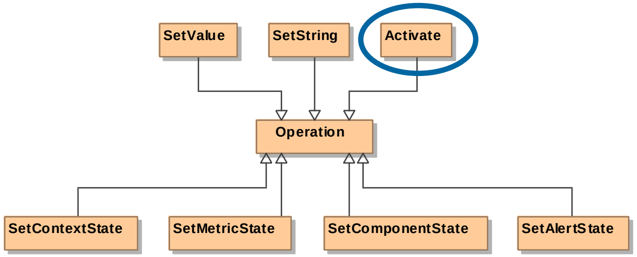


OperatingMode

- Defines if the operation is accessible
- Set of valid values: enabled, disabled or not available (i.e., generally defined but not in a mode where interaction is possible)

Properties

pm:ActivateOperation



Argument [list]

- Arguments that can be consumed by the operation. Every argument is identifiable by a pm:CodedValue (=nomenclature) and Qname (=XML Schema type)

ActivationDuration [optional]

- Defines the time period an activation is valid.

Retriggerable [optional]

- Only needed if an activation duration is defined. Designates if a call to the activate operation resets the current activation duration (=true) or toggles a value (=false).

Properties
Remaining data types

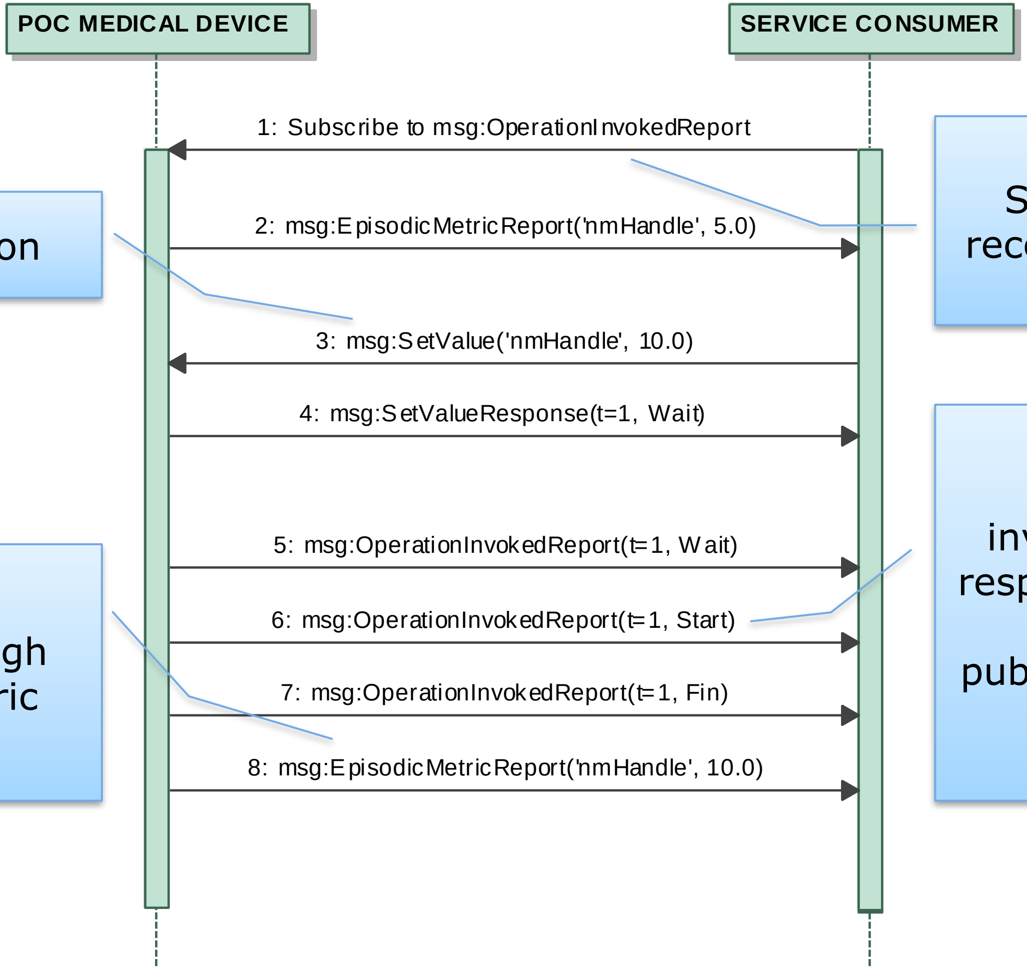
- Other types including the ActivateOperationState do not possess any further properties

TRANSACTION INVOCATION

Transaction invocation
Introduction

- BICEPS defines a transactional remote procedure call model in order to allow long-running actions without blocking, and with the ability to keep other participants informed
- Example: imagine a setting that is smoothly adapted till it finally reaches an expected value. The larger the difference between the actual and expected values are, the longer time it takes to reach the expected value.
- To let any client know about the actual processing status of a remote procedure call, BICEPS provides the msg:OperationInvokedReport notification
  - Should be subscribed before invoking an operation

Transactional invocation
Exemplary operation invocation


POC MEDICAL DEVICE
SERVICE CONSUMER
1: Subscribe to msg:OperationInvokedReport
2: msg:EpisodicMetricReport('nmHandle', 5.0)
3: msg:SetValue('nmHandle', 10.0)
4: msg:SetValueResponse(t=1, Wait)
5: msg:OperationInvokedReport(t=1, Wait)
6: msg:OperationInvokedReport(t=1, Start)
7: msg:OperationInvokedReport(t=1, Fin)
8: msg:EpisodicMetricReport('nmHandle', 10.0)

Invoke operation

New value is delivered through dedicated metric report

Subscribe in order to receive operation reports

Receive several invocation reports; the response status of the set request has to be published via notification, too

Transactional invocation

Invocation report attributes I

TransactionId

- Identifier that connects several operation-invoked messages. The transaction identifier is created by the device and returned as a response to the operation call

InvocationState

- Current status delivered with an operation-invoked report: Wait, Start, Cnclld, CnclldMan, Fin, FinMod, and Fail (see also next slide)

OperationHandleRef

- Handle reference that provides a link to the set operation processed by the transaction.

OperationTarget [optional]

- Handle reference that provides a link to the entity affected by the transaction

Transactional invocation

Invocation report attributes II

InvocationError [optional]

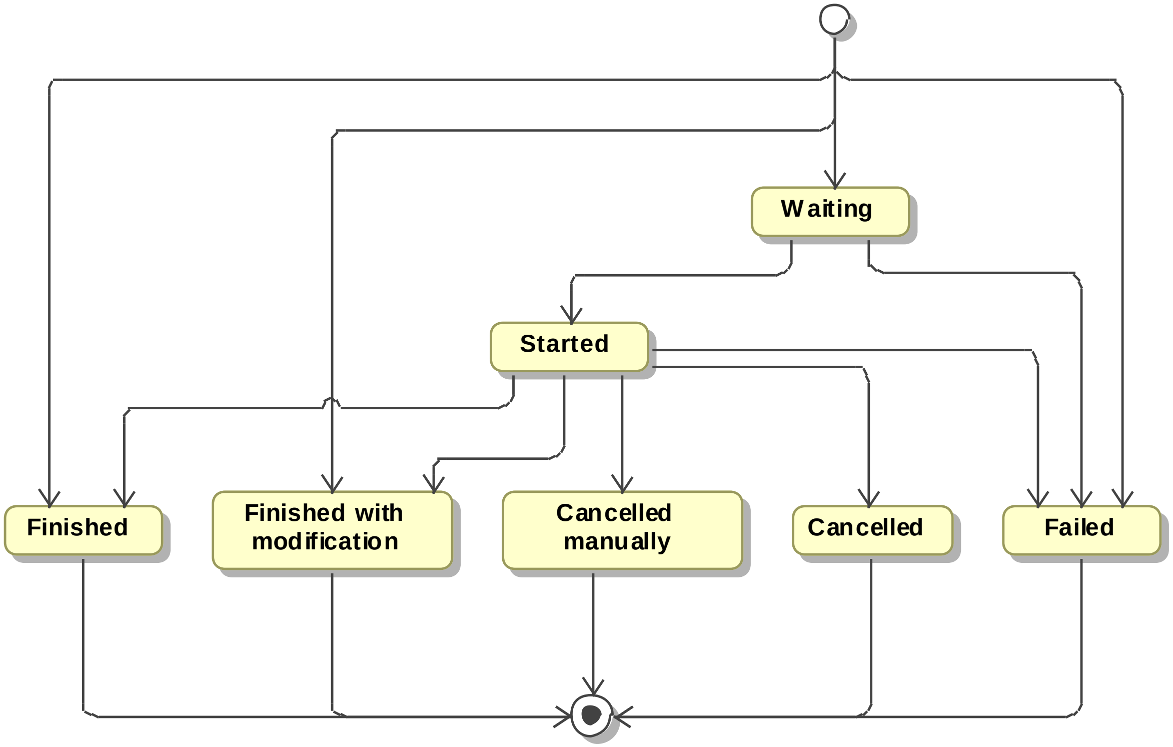
- Enumeration that indicates
  - *Unspec* if there is no specific operation error
  - *Unkn* if the operation handle is unknown
  - *Inv* if the operation handle points to an object that is not allowed for the operation type
  - *Oth* if the error is indicated by other means (e.g., an extension)

InvocationErrorMessage [optional]

- Human-readable description of the error

Properties

pm:ScoState



- **Waiting**: operation has been queued and waits for execution
- **Start**: execution of the operation has been started
- **Cnclld**: execution has been cancelled
- **CnclldMan**: execution has been cancelled by the operator
- **Fin**: execution has been finished
- **FinMod**: w/modification → next best value has been chosen as target value
- **Fail**: execution failed

Thank you for your attention!

Contact information

David Gregorczyk

david.gregorczyk@draeger.com