

**DIN EN 62541-10**

ICS 35.240.50; 35.200

Einsprüche bis 2014-12-24  
Vorgesehen als Ersatz für  
DIN EN 62541-10:2013-06**Entwurf****OPC Unified Architecture –  
Teil 10: Programme  
(IEC 65E/383/CDV:2014);  
Englische Fassung FprEN 62541-10:2014**OPC Unified Architecture –  
Part 10: Programs  
(IEC 65E/383/CDV:2014);  
English version FprEN 62541-10:2014Architecture unifiée OPC –  
Partie 10: Programmes  
(CEI 65E/383/CDV:2014);  
Version anglaise FprEN 62541-10:2014**Anwendungswarnvermerk**

Dieser Norm-Entwurf mit Erscheinungsdatum 2014-10-24 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter [www.entwuerfe.din.de](http://www.entwuerfe.din.de) bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter [www.entwuerfe.normenbibliothek.de](http://www.entwuerfe.normenbibliothek.de), sofern dort wiedergegeben;
- oder als Datei per E-Mail an [dke@vde.com](mailto:dke@vde.com) möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter [www.din.de/stellungnahme](http://www.din.de/stellungnahme) oder für Stellungnahmen zu Norm-Entwürfen der DKE unter [www.dke.de/stellungnahme](http://www.dke.de/stellungnahme) abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 49 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

## Nationales Vorwort

Die Englische Fassung des europäischen Dokuments FprEN 62541-10:2014 „OPC Unified Architecture – Teil 10: Programme“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 65E/383/CDV:2014 „OPC Unified Architecture – Part 10: Programs“ unverändert in den Entwurf FprEN 62541-10:2014 übernommen worden.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 931 „Systemaspekte“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE ([www.dke.de](http://www.dke.de)) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für den vorliegenden Normentwurf erfüllt.

Da sich die Benutzer des vorliegenden Normentwurfs der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 62541-10 veröffentlicht. Zu deren Abschnitt 3, Begriffe, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Normentwurfs hinzugefügt. Dieser Normentwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. ISO [und/oder] IEC sind nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

## Änderungen

Gegenüber DIN EN 62541-10:2013-06 wurden folgende Änderungen vorgenommen:

- a) In 4.2.2 Aufnahme von IT-Sicherheit, auf Basis von Anregungen der NIST;
- b) die Festlegung von Program Diagnostic Type in einen Datentyp umgewandelt und in Tabelle 5 zu Programmtypen den fehlenden Datentyp für Program Diagnostic Variable aufgenommen;
- c) in Tabelle 7 den Browser-Namen des Audit event für Program Transition berichtigt.

## Nationaler Anhang NA (informativ)

### 3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC/TR 62541-1 und IEC 62541-3 und die folgenden Begriffe.

#### 3.1.1

##### **Funktion**

(en: function)

programmtechnische Aufgabenstellung, ausgeführt auf einem Server oder Gerät, gewöhnlich durch Maschinencode

#### 3.1.2

##### **endlicher Zustandsautomat**

(en: finite State Machine)

Sequenz von Zuständen und gültigen Zustandsübergängen zusammen mit den Auslösern und Wirkungen dieser Zustandsübergänge, die die Aktionen eines Programms im Sinne von diskreten Phasen definieren

#### 3.1.3

##### **ProgramType**

(en: ProgramType)

Objektypknoten, der die Typdefinition eines Programms repräsentiert und ein Subtyp des FinitState-MachineType ist

#### 3.1.4

##### **Programmsteuerungsmethode**

(en: Program Control Method)

Methode, definiert in dieser Spezifikation mit spezifischer Semantik und angelegt zur Kontrolle eines Programms durch Auslösen eines Zustandsübergangs

#### 3.1.5

##### **Programmaufruf**

(en: Program Invocation)

eindeutige Objektinstanz eines Programms, die auf einem Server existiert

ANMERKUNG 1 zum Begriff: Ein Programmaufruf unterscheidet sich von anderen Objektinstanzen desselben ProgramType durch den eindeutigen Suchpfad (en: browse path) des Knotens.

– Leerseite –

## CONTENTS

Page

|    |                                                |    |
|----|------------------------------------------------|----|
| 1  | Scope .....                                    | 1  |
| 2  | Reference documents .....                      | 1  |
| 3  | Terms, definitions and conventions .....       | 1  |
| 4  | 3.1 Terms and definitions .....                | 1  |
| 5  | 3.2 Abbreviations .....                        | 2  |
| 6  | Concepts .....                                 | 3  |
| 7  | 4.1 General .....                              | 3  |
| 8  | 4.2 Programs .....                             | 4  |
| 9  | 4.2.1 Overview .....                           | 4  |
| 10 | 4.2.2 Security Considerations .....            | 4  |
| 11 | 4.2.3 Program Finite State Machine .....       | 4  |
| 12 | 4.2.4 Program States .....                     | 5  |
| 13 | 4.2.5 State Transitions .....                  | 6  |
| 14 | 4.2.6 Program State Transition Stimuli .....   | 6  |
| 15 | 4.2.7 Program Control Methods .....            | 6  |
| 16 | 4.2.8 Program State Transition Effects .....   | 7  |
| 17 | 4.2.9 Program Result Data .....                | 7  |
| 18 | 4.2.10 Program Lifetime .....                  | 8  |
| 19 | Model .....                                    | 8  |
| 20 | 5.1 General .....                              | 8  |
| 21 | 5.2 ProgramType .....                          | 9  |
| 22 | 5.2.1 Overview .....                           | 9  |
| 23 | 5.2.2 ProgramType Properties .....             | 10 |
| 24 | 5.2.3 ProgramType Components .....             | 11 |
| 25 | 5.2.4 ProgramType Causes (Methods) .....       | 15 |
| 26 | 5.2.5 ProgramType Effects (Events) .....       | 17 |
| 27 | 5.2.6 AuditProgramTransitionEventType .....    | 19 |
| 28 | 5.2.7 FinalResultData .....                    | 20 |
| 29 | 5.2.8 ProgramDiagnostic DataType .....         | 20 |
| 30 | 5.2.9 ProgramDiagnosticType VariableType ..... | 20 |
| 31 | Annex A (informative) Program Example .....    | 22 |
| 32 | A.1 Overview .....                             | 22 |
| 33 | A.2 DomainDownload Program .....               | 22 |
| 34 | A.2.1 General .....                            | 22 |
| 35 | A.2.2 DomainDownload States .....              | 23 |
| 36 | A.2.3 DomainDownload Transitions .....         | 24 |
| 37 | A.2.4 DomainDownload Methods .....             | 24 |
| 38 | A.2.5 DomainDownload Events .....              | 25 |
| 39 | A.2.6 DomainDownload Model .....               | 25 |

45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55

FIGURES

|                                                |    |
|------------------------------------------------|----|
| Figure 1 – Automation facility control.....    | 3  |
| Figure 2 – Program Illustration .....          | 4  |
| Figure 3 – Program States and Transitions..... | 5  |
| Figure 4 – Program Type.....                   | 9  |
| Figure 5 – Program FSM References .....        | 11 |
| Figure 6 – ProgramType Causes and Effects..... | 15 |

TABLES

|    |                                                       |    |
|----|-------------------------------------------------------|----|
| 56 |                                                       |    |
| 57 | Table 1 – Program Finite State Machine .....          | 4  |
| 58 | Table 2 – Program States.....                         | 5  |
| 59 | Table 3 – Program State Transitions .....             | 6  |
| 60 | Table 4 – Program Control Methods .....               | 7  |
| 61 | Table 5 – ProgramType .....                           | 9  |
| 62 | Table 6 – Program States.....                         | 11 |
| 63 | Table 7 – Program Transitions .....                   | 13 |
| 64 | Table 8 – ProgramType Causes .....                    | 16 |
| 65 | Table 9 – ProgramTransitionEventType.....             | 17 |
| 66 | Table 10 – ProgramTransitionEvents.....               | 18 |
| 67 | Table 11 – AuditProgramTransitionEventType.....       | 19 |
| 68 | Table 12 – ProgramDiagnosticDataType Structure .....  | 20 |
| 69 | Table 13 – ProgramDiagnosticDataType Definition ..... | 20 |
| 70 | Table 14 – ProgramDiagnosticType.....                 | 20 |

71

72

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

## OPC Unified Architecture –

## Part 10: Programs

## FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62541-10 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this standard is based on the following documents:

| FDIS        | Report on voting |
|-------------|------------------|
| 65E/XX/FDIS | 65E/XX/RVD       |

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 62541 series, published under the general title *OPC Unified Architecture*, can be found on the IEC website.

122 The committee has decided that the contents of this publication will remain unchanged until the  
123 stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the  
124 specific publication. At this date, the publication will be

- 125 • reconfirmed,
- 126 • withdrawn,
- 127 • replaced by a revised edition, or
- 128 • amended.

129

130 The National Committees are requested to note that for this publication the stability date is 2017.

131 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT  
132 THE PUBLICATION STAGE.

133

**IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.**

134



# OPC Unified Architecture Specification

## Part 10: Programs

### 1 Scope

This specification is part of the overall OPC Unified Architecture (OPC UA) standard series and defines the information model associated with *Programs*. This includes the description of the *NodeClasses*, standard *Properties*, *Methods* and *Events* and associated behaviour and information for *Programs*.

The complete address space model including all *NodeClasses* and *Attributes* is specified in IEC 62541-3. The services such as those used to invoke the *Methods* used to manage *Programs* are specified in IEC 62541-4.

### 2 Reference documents

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC/TR 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts*

IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*

IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

IEC 62541-5, *OPC Unified Architecture – Part 5: Information Model*

IEC 62541-7, *OPC Unified Architecture – Part 7: Profiles*

### 3 Terms, definitions and conventions

#### 3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC/TR 62541-1, and IEC 62541-3 as well as the following apply.

##### 3.1.1

##### Function

programmatic task performed by a server or device, usually accomplished by computer code execution

##### 3.1.2

##### Finite State Machine

sequence of states and valid state transitions along with the causes and effects of those state transitions that define the actions of a *Program* in terms of discrete stages

171 **3.1.3**172 **ProgramType**173 type definition of a *Program* and is a subtype of the *FiniteStateMachineType*

174

175 **3.1.4**176 **Program Control Method**177 *Method* specified by this specification having specific semantics designed for the control of a178 *Program* by causing a state transition179 **3.1.5**180 **Program Invocation**181 unique *Object* instance of a *Program* existing on a *Server*182 NOTE A Program Invocation is distinguished from other Object instances of the same ProgramType by the object node's  
183 unique browse path.184 **3.2 Abbreviations**

185 API Application Programming Interface

186 DA Data Access

187 FSM Finite State Machine

188 HMI Human Machine Interfaces

189 PCM Program Control Method

190 PGM Program

191 PI Program Invocation

192 PLC Programmable Logic Controller

193 UA Unified Architecture

194 UML Unified Modelling Language

## 4 Concepts

### 4.1 General

Integrated automation facilities manage their operations through the exchange of data and coordinated invocation of system functions like illustrated in Figure 1. *Services* are required to perform the data exchanges and to invoke the functions that constitute system operation. These functions may be invoked through human machine interfaces, cell controllers, or other supervisory control and data acquisition type systems. OPC UA defines *Methods* and *Programs* as an interoperable way to advertise, discover, and request these functions. They provide a normalizing mechanism for the semantic description, invocation of, and result reporting of these functions. Together *Methods* and *Programs* complement the other OPC UA *Services* and *ObjectTypes* to facilitate the operation of an automation environment using a client server hierarchy.

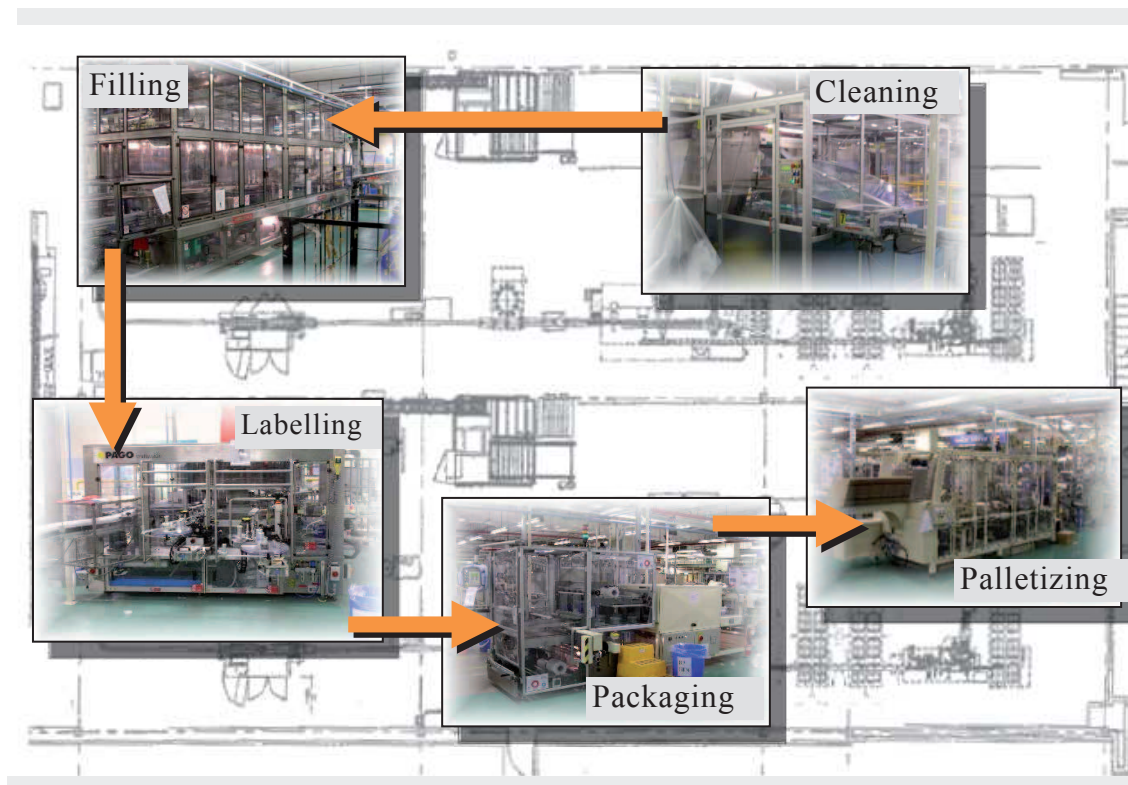


Figure 1 – Automation facility control

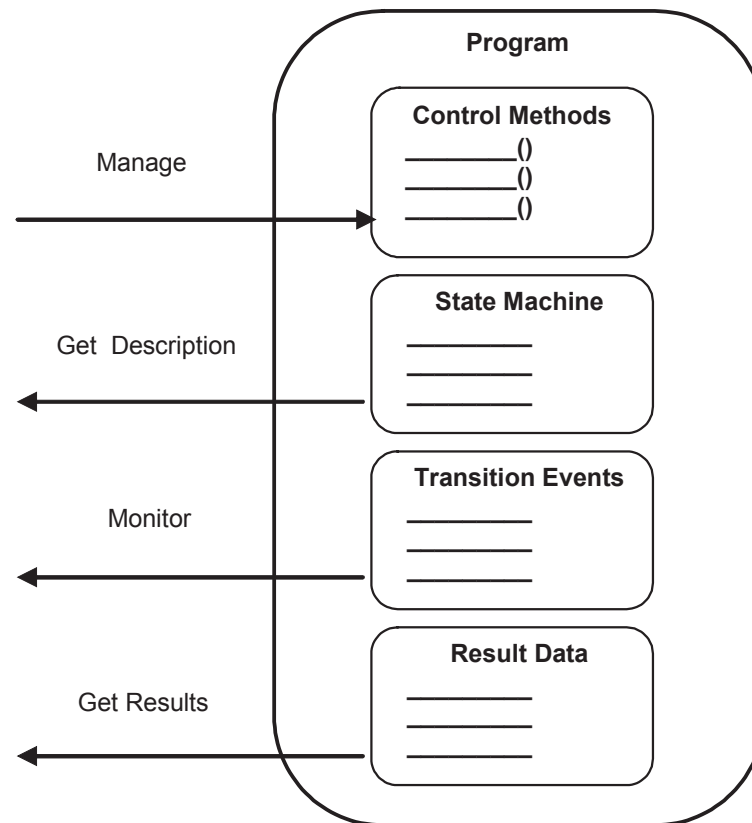
*Methods* and *Programs* model functions typically having different scopes, behaviours, lifetimes, and complexities in *OPC Servers* and the underlying systems. These functions are **not** normally characterized by the reading or writing of data which is accomplished with the OPC UA *Attribute* service set.

*Methods* represent basic functions in the server that can be invoked by a client. *Programs* by contrast, model more complex and stateful functionality in the system. For example, a method call may be used to perform a calculation or reset a counter. A *Program* is used to run and control a batch process, execute a machine tool part program, or manage a domain download. *Methods* and their invocation mechanism are described in IEC 62541-3 and IEC 62541-4.

This specification describes the extensions to, or specific use of the core capabilities defined in IEC 62541-5 as required for *Programs*.

220 **4.2 Programs**221 **4.2.1 Overview**

222 *Programs* are complex functions in a server or underlying system that can be invoked and managed  
 223 by a *Client*. *Programs* can represent any level of functionality within a system or process in which  
 224 client control or intervention is required and progress monitoring is desired. Figure 2 illustrates the  
 225 model.



226  
227 **Figure 2 – Program Illustration**

228 *Programs* are state full and transition through a prescribed sequence of states as they execute. Their  
 229 behaviour is defined by a *Program Finite State Machine (PFSM)*. The elements of the PFSM describe  
 230 the phases of a *Program's* execution in terms of valid transitions between a set of states, the stimuli  
 231 or causes of those transitions, and the resultant effects of the transitions.

232 **4.2.2 Security Considerations**

233 Since *Programs* can be used to perform advanced control algorithms or other actions, their use  
 234 should be restricted to personnel with appropriate access rights. It is recommended that  
 235 AuditUpdateMethodEvents are generated to allow monitoring the number of running programs in  
 236 addition to their execution frequency.

237 **4.2.3 Program Finite State Machine**

238 The states, transitions, causes and effects that compose the Program Finite State Machine are listed  
 239 in Table 1 and illustrated in Figure 3.

240 **Table 1 – Program Finite State Machine**

| No. | Transition Name | Cause        | From State | To State | Effect              |
|-----|-----------------|--------------|------------|----------|---------------------|
| 1   | HaltedToReady   | Reset Method | Halted     | Ready    | Report Transition 1 |

|   |                    |                                    |           |           | Event/Result                        |
|---|--------------------|------------------------------------|-----------|-----------|-------------------------------------|
| 2 | ReadyToRunning     | Start Method                       | Ready     | Running   | Report Transition 2<br>Event/Result |
| 3 | RunningToHalted    | Halt Method or<br>Internal (Error) | Running   | Halted    | Report Transition 3<br>Event/Result |
| 4 | RunningToReady     | Internal                           | Running   | Ready     | Report Transition 4<br>Event/Result |
| 5 | RunningToSuspended | Suspend Method                     | Running   | Suspended | Report Transition 5<br>Event/Result |
| 6 | SuspendedToRunning | Resume Method                      | Suspended | Running   | Report Transition 6<br>Event/Result |
| 7 | SuspendedToHalted  | Halt Method                        | Suspended | Halted    | Report Transition 7<br>Event/Result |
| 8 | SuspendedToReady   | Internal                           | Suspended | Ready     | Report Transition 8<br>Event/Result |
| 9 | ReadyToHalted      | Halt Method                        | Ready     | Halted    | Report Transition 9<br>Event/Result |

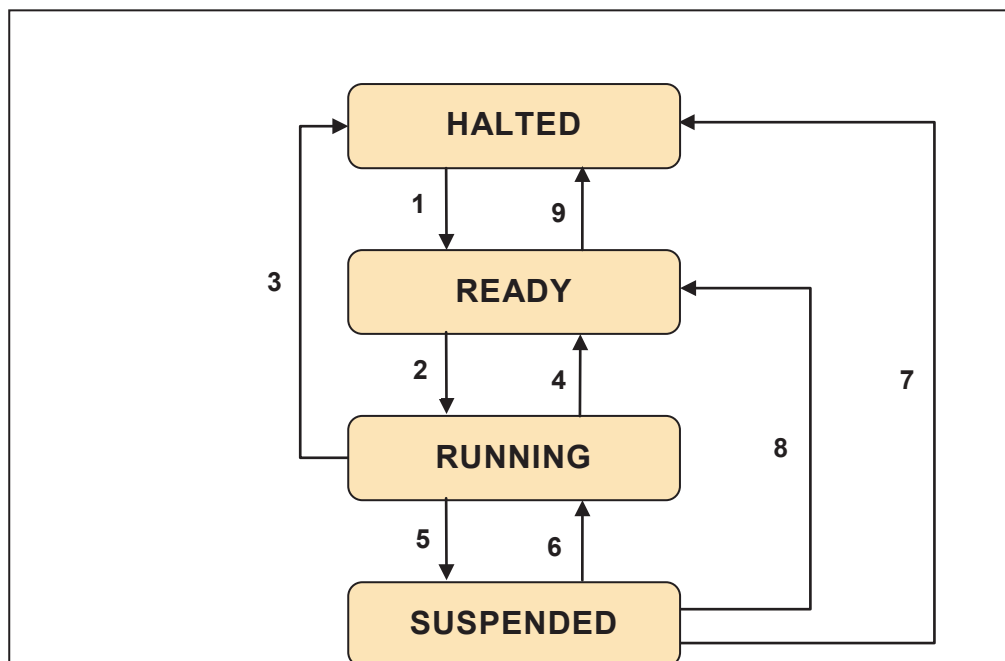


**Figure 3 – Program States and Transitions**

#### 4.2.4 Program States

A standard set of base states are defined for *Programs* as part of the *Program Finite State Machine*. These states represent the stages in which a *Program* can exist at an instance in time as viewed by a client. This state is the *Program's Current State*. All *Programs* shall support this base set. A *Program* may or may not require a client action to cause the state to change. The states are formally defined in Table 2.

**Table 2 – Program States**

| State            | Description                                                                                |
|------------------|--------------------------------------------------------------------------------------------|
| <b>Ready</b>     | The <i>Program</i> is properly initialized and may be started.                             |
| <b>Running</b>   | The <i>Program</i> is executing making progress towards completion.                        |
| <b>Suspended</b> | The <i>Program</i> has been stopped prior to reaching a terminal state but may be resumed. |