behavior\_annex ::=

[ **variables** { behavior\_variable }+ ]

[ **states** { behavior\_state }+ ]

[ **transitions** { behavior\_transition }+ ]

H

behavior\_variable ::=

*local\_variable\_*declarator { **,** *local\_variable\_*declarator }\*

**:** *data\_*unique\_component\_classifier\_reference [ **{** { *data\_classifier\_*property\_association }+ **}** ] **;**

unique\_component\_classifier\_reference ::=

{ *package\_*identifier **::** }\* *component\_type\_*identifier

[ **.** *component\_implementation\_*identifier ]

behavior\_state ::=

*behavior\_state\_*identifier { **,** *behavior\_state\_*identifier }\*

**:** behavior\_state\_kind **state ;**

behavior\_state\_kind ::=

[ **initial** ][ **complete** ][ **final** ]

behavior\_transition ::=

[ *transition\_*identifier [ **[** behavior\_transition\_priority **]** ] **:** ]

*source\_state\_*identifier { , *source\_state\_*identifier }\*

**-[** behavior\_condition **]->**

*destination\_state\_*identifier [ behavior\_action\_block ] **;**

behavior\_transition\_priority ::=

numeral

behavior\_condition ::=

dispatch\_condition

| execute\_condition

execute\_condition ::=

[ *logical\_*value\_expression | *behavior\_action\_block\_*timeout\_catch | **otherwise** ]

timeout\_catch ::=

**timeout**

declarator ::= identifier { array\_size }\*

array\_size :: **[** *integer\_*value\_constant **]**

dispatch\_condition ::=

**on dispatch** [ dispatch\_trigger\_condition ] [ **frozen** frozen\_ports ]

dispatch\_trigger\_condition ::=

dispatch\_trigger\_logical\_expression

~~|~~ *~~provides\_subprogram\_access\_~~*~~identifier~~

| *provides\_subprogram\_access\_*name

| **stop**

| completion\_relative\_timeout\_condition\_and\_catch

| dispatch\_relative\_timeout\_catch

dispatch\_trigger\_logical\_expression ::=

dispatch\_conjunction { **or** dispatch\_conjunction }\*

dispatch\_conjunction ::=

dispatch\_trigger { **and** dispatch\_trigger }\*

completion\_relative\_timeout\_condition\_and\_catch ::=

**timeout** behavior\_time

dispatch\_relative\_timeout\_catch ::=

**timeout**

~~dispatch\_trigger ::=~~

*~~in\_event\_port\_~~*~~identifier~~

~~|~~ *~~in\_event\_data\_port\_~~*~~identifier~~

dispatch\_trigger ::=

*in\_event\_port\_*name

| *in\_event\_data\_port\_*name

~~frozen\_ports ::=~~

*~~in\_port\_~~*~~identifier {~~ **~~,~~***~~in\_port\_~~*~~identifier }\*~~

frozen\_ports ::=

*in\_port\_*name { **,** *in\_port\_*name }\*

behavior\_action\_block ::=

**{** behavior\_actions **}** [ **timeout** behavior\_time ]

behavior\_actions ::=

behavior\_action

| behavior\_action\_sequence

| behavior\_action\_set

behavior\_action\_sequence ::=

behavior\_action { **;** behavior\_action }+

behavior\_action\_set ::=

behavior\_action { **&** behavior\_action }+

behavior\_action ::=

basic\_action

| behavior\_action\_block

| **if** **(** *logical\_*value\_expression **)** behavior\_actions

{ **elsif** **(** *logical\_*value\_expression **)** behavior\_actions }\*

[ **else** behavior\_actions ]

**end if**

| **for** **(**

*element\_*identifier **:** *data\_*unique\_component\_classifier\_reference

**in** element\_values **)** **{** behavior\_actions **}**

| **forall** **(**

*element\_*identifier **:** *data\_*unique\_component\_classifier\_reference

**in** element\_values **)** **{** behavior\_actions **}**

| **while** **(** *logical\_*value\_expression **)** **{** behavior\_actions **}**

| **do** behavior\_actions **until** **(** *logical\_*value\_expression **)**

element\_values ::=

integer\_range

| *event\_data\_port\_*name

| *array\_*data\_component\_reference

basic\_action ::=

assignment\_action

| communication\_action

| timed\_action

assignment\_action ::=

target **:=** ( value\_expression | **any** )

A

target ::=

*~~local\_variable\_~~*~~name~~

| *outgoing\_port\_*name

~~|~~ *~~outgoing\_subprogram\_parameter\_~~*~~name~~

| *outgoing\_*subprogram\_parameter

| data\_component\_reference

| outgoing\_port\_prototype\_name

communication\_action ::=

*subprogram\_prototype\_*name **!** [ **(** subprogram\_parameter\_list **)** ]

| *required\_subprogram\_access\_*name **!** [ **(** subprogram\_parameter\_list **)** ]

| *subprogram\_subcomponent\_*name **!** [ **(** subprogram\_parameter\_list **)** ]

| *subprogram\_*unique\_component\_classifier\_reference **!** [ **(** subprogram\_parameter\_list **)** ]

| *output\_port\_*name **!** [ **(** value\_expression **)** ]

| *input\_port\_*name **>>**

| *input\_port\_*name **?** [ **(** target **)** ]

| *required\_data\_access\_*name **!<**

| *required\_data\_access\_*name **!>**

| *required\_data\_access\_*name . *provided\_subprogram\_access\_*name **!** [ **(** subprogram\_parameter\_list **)** ]

| *data\_subcomponent\_*name . *provided\_subprogram\_access\_*name **!** [ **(** subprogram\_parameter\_list **)** ]

| *local\_variable\_*name . *provided\_subprogram\_access\_*name **!** [ **(** subprogram\_parameter\_list **)** ]

| **\*!<**

| **\*!>**

timed\_action ::=

**computation** **(** behavior\_time [ **..** behavior\_time ] **)**

subprogram\_parameter\_list ::=

parameter\_label { **,** parameter\_label }\*

parameter\_label ::=

*in\_*subprogram\_parameter | *out\_*subprogram\_parameter

B

data\_component\_reference ::=

*~~data\_subcomponent\_~~*~~name {~~ **~~.~~** ~~data\_subcomponent\_name }\*~~

~~|~~ *~~data\_access\_feature\_~~*~~name {~~ **~~.~~** ~~data\_subcomponent\_name }\*~~

*data\_subcomponent\_*name { **.** data\_element }\*

| *data\_access\_feature\_*name { **.** data\_element }\*

| *local\_variable\_*identifier { . data\_element }\*   
| *data\_access\_feature\_prototype*\_name { **.** data\_element }\*

| **self** { **.** data\_element }+

C

data\_element ::=

*data\_subcomponent\_*name

| *data\_access\_*name

| *data\_access\_prototype*\_name

D

subprogram\_parameter ::=

*subprogram\_parameter\_*name {**.** data\_element }\*

E

port\_component\_reference ::=

*subcomponent\_*name **.** port\_element

| **self** **.** port\_element

port\_element ::=

*port*\_name

| *port\_prototype*\_name

~~name ::=~~

~~identifier { array\_index }\*~~

name ::=

{ *thread\_group\_*identifier **.** }\*

{

[ *subprogram\_group\_access\_*identifier **.** ]

[ *subprogram\_group\_*identifier**.** ]

[ *feature\_group\_*identifier **.** ]

}\*

identifier { array\_index }\*

~~array\_index ::=~~

**~~[~~***~~integer\_~~*~~value\_variable~~ **~~]~~**

array\_index ::=

**[** *integer\_*value **]**

value ::=

value\_variable

| value\_constant

| **(** value\_expression **)**

F

value\_variable ::=

*incoming\_port\_*name

| *incoming\_port\_*name **?**

~~|~~ *~~incoming\_subprogram\_parameter\_~~*~~name~~

| *incoming\_*subprogram\_parameter

~~|~~ *~~local\_variable\_~~*~~name~~

| *incoming\_port\_prototype\_*name

| data\_component\_reference

| *port\_*name **’ count**

| *port\_*name ’ **fresh**

value\_constant ::=

boolean\_literal

| numeric\_literal

| string\_literal

| property\_constant

| property\_value

value\_expression ::=

relation { logical\_operator relation}\*

relation ::=

simple\_expression [ relational\_operator simple\_expression ]

simple\_expression ::=

[ unary\_adding\_operator ] term { binary\_adding\_operator term }\*

term ::=

factor { multiplying\_operator factor }\*

factor ::=

value [ binary\_numeric\_operator value ]

| unary\_numeric\_operator value

| unary\_boolean\_operator value

logical\_operator ::=

**and** | **or** | **xor**

relational\_operator ::=

**=** | **!=** | **<** | **<=** | **>** | **>=**

binary\_adding\_operator ::=

**+** | **-**

unary\_adding\_operator ::=

**+** | **-**

multiplying\_operator ::=

**\*** | **/** | **mod** | **rem**

binary\_numeric\_operator ::=

**\*\***

unary\_numeric\_operator ::=

**abs**

unary\_boolean\_operator ::=

**not**

boolean\_literal ::=

**true** | **false**

integer\_range ::=

integer\_value **..** integer\_value

integer\_value ::=

*integer\_*value\_variable

| *integer\_*value\_constant

behavior\_time ::=

integer\_value *unit\_*identifier

property\_constant ::=

[ *property\_set\_*identifier **::** ] *property\_constant\_*identifier

property\_value ::=

[ *property\_set\_*identifier **::** ] *property\_value\_*identifier

G

property\_reference ::=

property\_value

| *own\_*component\_element **#** property\_name { **.** *record\_ field\_*property\_name }\*

| unique\_component\_classifier\_reference **#** property\_name { **.** *record\_ field\_*property\_name }\*

property\_name ::=

property\_value { property\_field }\*

property\_field ::=

**[** *integer\_*value **]**

| **.** *item\_list\_*identifier

| **.** **upper\_bound**

| **.** **lower\_bound**

component\_element ::=

*subcomponent\_*name

| *local\_variable\_*name

| *binded\_prototype\_*name

| *feature\_*name

| **self**

numeric\_literal ::= <refer to [AS5506A 15.4]>

string\_literal ::= <refer to [AS5506A 15.5]>

numeral ::= <refer to [AS5506A 15.4.1]>