

Learning the Portable Stimulus Standard

Tom Fitzpatrick

CEO Big Fish EDA Consulting

On behalf of the Accellera Systems Initiative

and the Accellera Portable Stimulus Working Group



Hello, World!

 Session 02

What You Will Learn

- ▶ How to Define a Simple PSS Model
- ▶ Critical Parts of a PSS Model
- ▶ How a Simple PSS Model Can Define Multiple Scenarios

Looking Back at a Classic

Include the standard input/output library for printing

Main function - where the program starts

Print the text "Hello, World!" followed by a newline

Return 0 to indicate the program ran successfully

```
#include <stdio.h>
int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Hello, World in PSS

Hello
Hola
Guten Tag

hello_a

**World
Mundo
Welt**

world a

```
do hello_a;
```

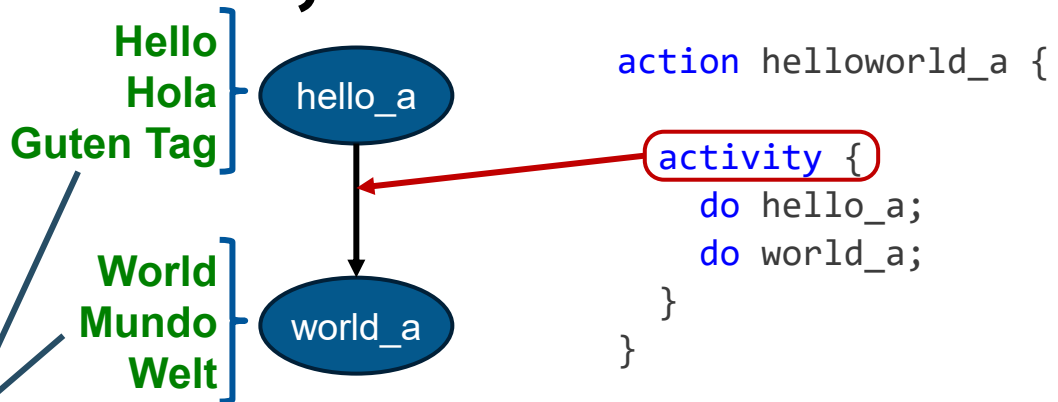
```
do world_a;
```

```
enum lang_e {English, Spanish, German};
```

```
action hello_a {  
    rand lang_e lang;  
}
```

```
action world_a {
```

::: Hello, World in PSS



```
enum lang_e {English, Spanish, German};
```

```
action hello_a {  
    rand lang_e lang;  
}
```

```
action world_a {  
  
}
```

Hello, World in PSS

Hello
Hola
Guten Tag

World
Mundo
Welt

Languages
must match

hello_a

world_a

```
action helloworld_a {
```

```
  activity {  
    do hello_a;  
    do world_a;  
  }  
}
```

```
buffer hello_b {  
  rand lang_e lang;  
  rand hello_e hello;  
}
```

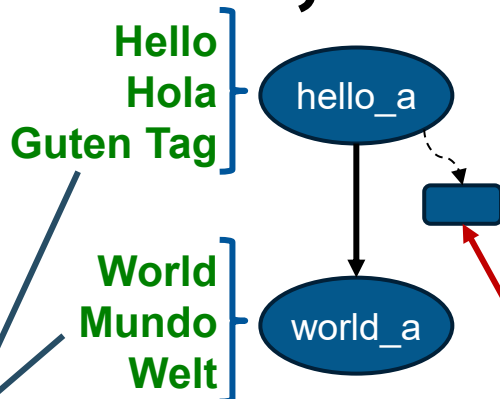
```
enum lang_e {English, Spanish, German};  
enum hello_e {Hello, Hola, Guten_Tag};
```

```
action hello_a {  
  output hello_b buf_o;  
  rand lang_e lang;  
}
```

```
action world_a {
```

```
}
```

Hello, World in PSS



Languages must match

```
action helloworld_a {  
    activity {  
        do hello_a;  
        do world_a;  
    }  
}
```

```
buffer hello_b {  
    rand lang_e lang;  
    rand hello_e hello;  
    constraint c_hello {  
        lang == English -> hello == Hello;  
        lang == Spanish -> hello == Hola;  
        lang == German -> hello == Guten_Tag;  
    }  
}
```

```
enum lang_e {English, Spanish, German};  
enum hello_e {Hello, Hola, Guten_Tag};
```

```
action hello_a {  
    output hello_b buf_o;  
    rand lang_e lang;  
    constraint buf_o.lang == lang;  
}
```

```
action world_a {
```

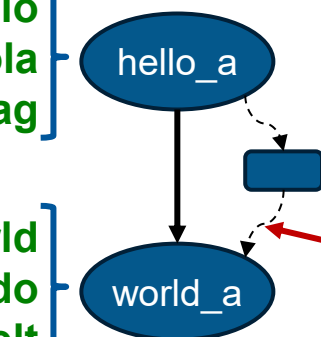
```
}
```

Hello, World in PSS

Hello
Hola
Guten Tag

World
Mundo
Welt

Languages
must match



```
action helloworld_a {
```

```
  activity {  
    do hello_a;  
    do world_a;  
  }
```

```
}
```

```
enum lang_e {English, Spanish, German};  
enum hello_e {Hello, Hola, Guten_Tag};  
enum world_e {World, Mundo, Welt};
```

```
action hello_a {  
  output hello_b buf_o;  
  rand lang_e lang;  
  constraint buf_o.lang == lang;  
}
```

```
buffer hello_b {  
  rand lang_e lang;  
  rand hello_e hello;  
  constraint c_hello {  
    lang == English -> hello == Hello;  
    lang == Spanish -> hello == Hola;  
    lang == German -> hello == Guten_Tag;  
  }  
}
```

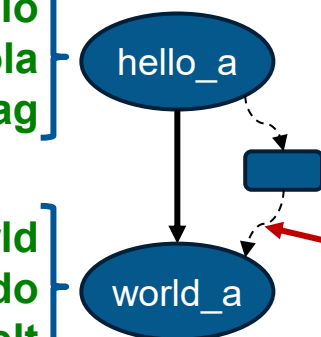
```
action world_a {  
  input hello_b buf_i;  
  rand world_e world;  
  constraint c_world {  
    buf_i.lang == English -> world == World;  
    buf_i.lang == Spanish -> world == Mundo;  
    buf_i.lang == German -> world == Welt;  
  }  
}
```

::: Hello, World in PSS

Hello
Hola
Guten Tag

World
Mundo
Welt

Languages
must match



```
action helloworld_a {
```

```
  activity {  
    do hello_a;  
    do world_a;  
  }
```

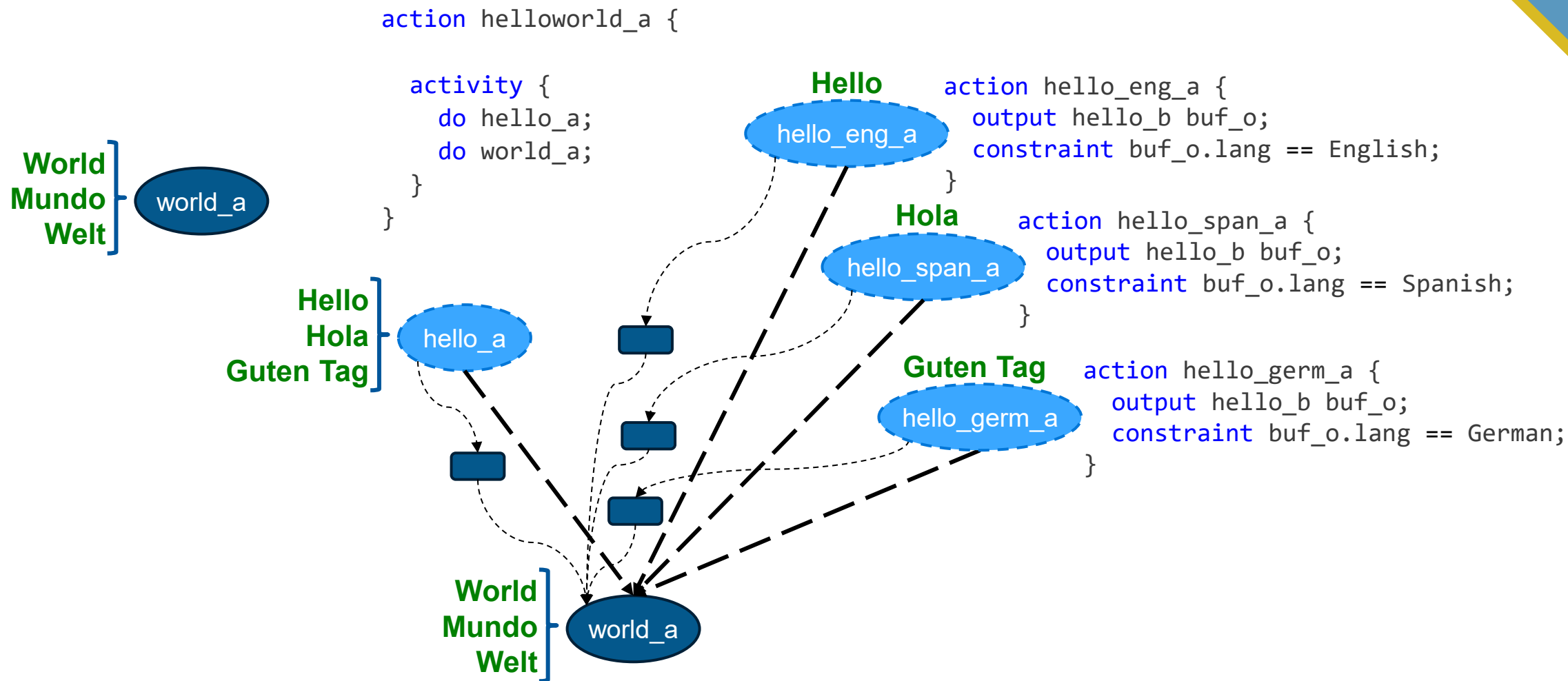
```
buffer hello_b {  
  rand lang_e lang;  
  rand hello_e hello;  
  constraint c_hello {  
    lang == English -> hello == Hello;  
    lang == Spanish -> hello == Hola;  
    lang == German -> hello == Guten_Tag;  
  }  
}
```

```
enum lang_e {English, Spanish, German};  
enum hello_e {Hello, Hola, Guten_Tag};  
enum world_e {World, Mundo, Welt};
```

```
action hello_a {  
  output hello_b buf_o;  
  rand lang_e lang;  
  constraint buf_o.lang == lang;  
}
```

```
action world_a {  
  input hello_b buf_i;  
  rand world_e world;  
  constraint c_world {  
    buf_i.lang == English -> world == World;  
    buf_i.lang == Spanish -> world == Mundo;  
    buf_i.lang == German -> world == Welt;  
  }  
  
  exec body {  
    message(NONE, "%n, %n!", buf_i.hello, world);  
  }  
}
```

::: Hello, World in PSS





Thank You!

