



Week 2: Development Tasks

Unit 1: Domain Modeling

What is Domain Modeling

1. **Keep your domain models clean, concise, comprehensible by**
Factoring out technical aspects
Separating Concerns, e.g.
 - Fiori Markup
 - Authorization
 - Persistence... in same or different files or projects

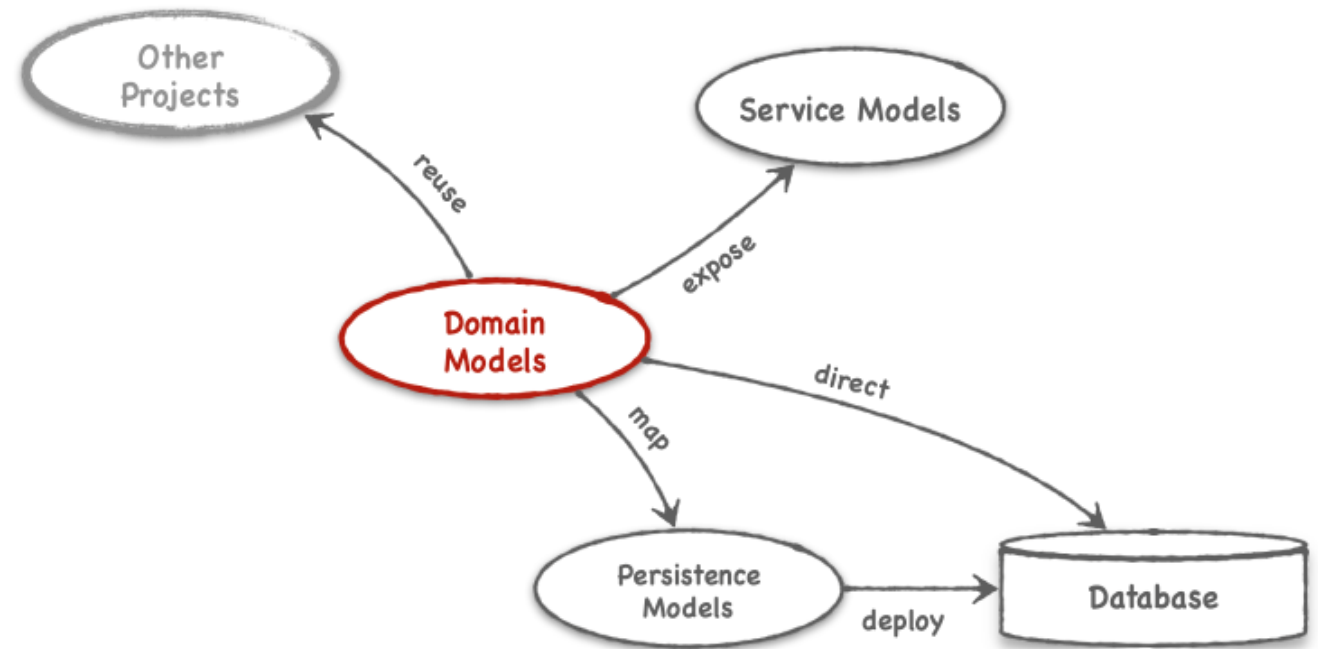
2. **Intrinsic Extensibility**

Project-level

→ Reuse and Extend

SaaS-level

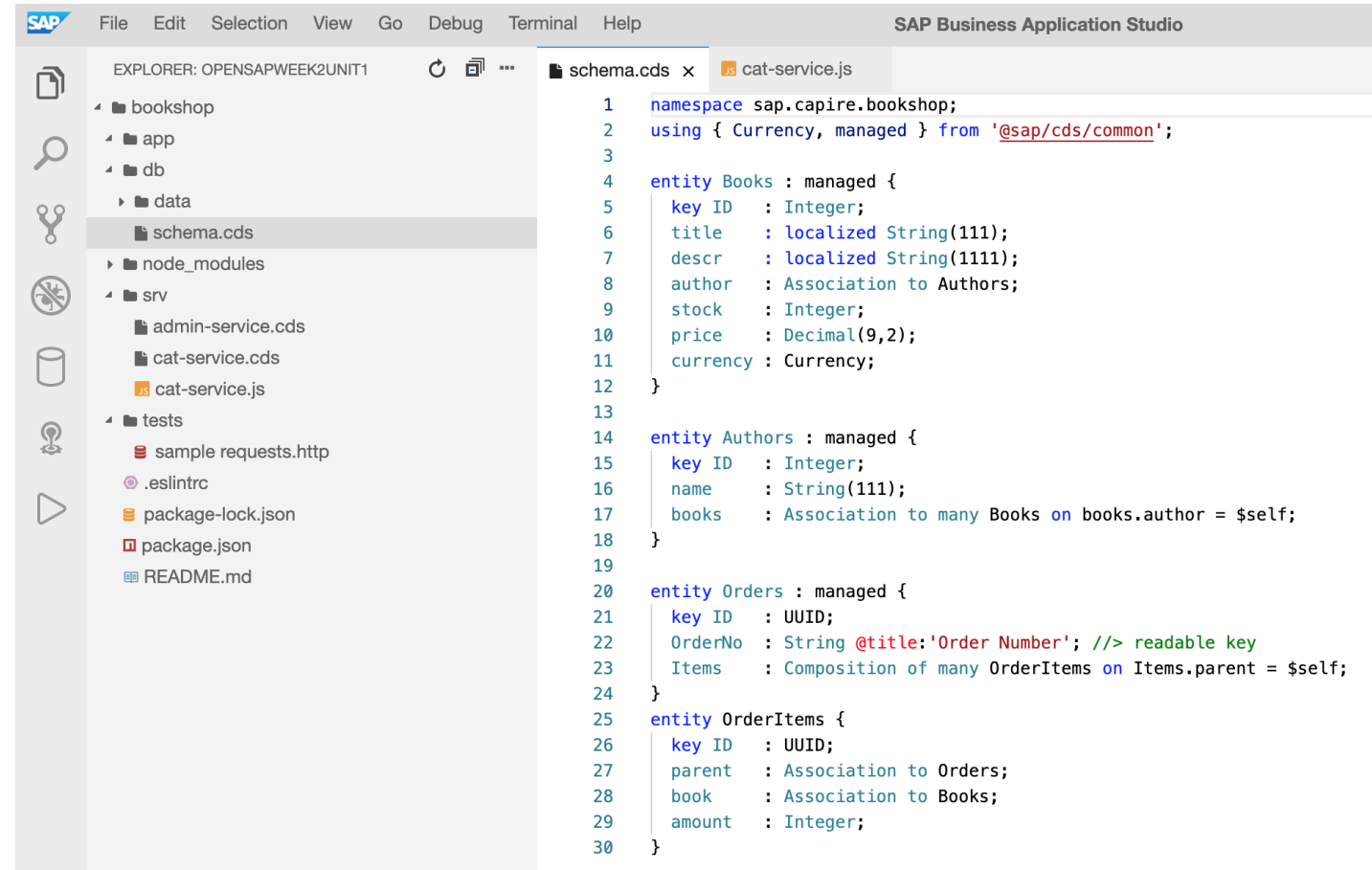
→ Dynamic Extensibility



Domain Modeling

Domain models

- Types, Entities and Associations
- Aspects
- Best Practices in Domain Modeling:
 - Naming Conventions
 - Namespaces
 - Avoid over Normalization
 - Simple, flat structures
 - Enterprise features
 - e.g. localized data



The screenshot shows the SAP Business Application Studio interface. On the left is the Explorer view showing a project structure for 'OPENSAPWEEK2UNIT1'. The project contains folders for 'bookshop', 'app', 'db', 'data', 'node_modules', 'srv', and 'tests'. The 'srv' folder is expanded, showing files like 'admin-service.cds', 'cat-service.cds', 'cat-service.js', 'sample requests.http', '.eslintrc', 'package-lock.json', 'package.json', and 'README.md'. The 'cat-service.js' file is selected. On the right is the Code Editor view showing the content of 'cat-service.js'. The code defines three entities: 'Books', 'Authors', and 'Orders', each with various attributes and associations.

```
1 namespace sap.capire.bookshop;
2 using { Currency, managed } from '@sap/cds/common';
3
4 entity Books : managed {
5   key ID : Integer;
6   title : localized String(111);
7   descr : localized String(1111);
8   author : Association to Authors;
9   stock : Integer;
10  price : Decimal(9,2);
11  currency : Currency;
12 }
13
14 entity Authors : managed {
15   key ID : Integer;
16   name : String(111);
17   books : Association to many Books on books.author = $self;
18 }
19
20 entity Orders : managed {
21   key ID : UUID;
22   OrderNo : String @title:'Order Number'; //> readable key
23   Items : Composition of many OrderItems on Items.parent = $self;
24 }
25 entity OrderItems {
26   key ID : UUID;
27   parent : Association to Orders;
28   book : Association to Books;
29   amount : Integer;
30 }
```

Entities, Types and Associations (recap)

```
namespace 'our.library';

entity Books {
  key ID : UUID;
  title : String;
  descr : String;
  author : Association to Authors;
}

entity Authors {
  key ID : UUID;
  name : String;
  books : Association to many Books on books.author=$self;
  birth : Date;
  death : Date;
}
```

1. **Entities** represent data that can be read and written by consumers, uniquely identified by their primary keys.
2. **Types** describe the types of elements within entities.
3. **Associations** capture relationships between entities.

```
entity Books {
  key ID : UUID;
  title : String;
  genre : Genre;
  author : Association to Authors;
}

type Genre : String enum {
  Mystery; Fiction; Drama;
}
```

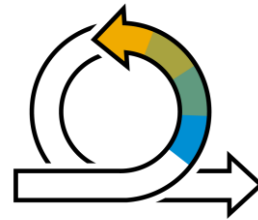
Domain Modeling

Aspects

The aspect '*additionalInfo*' can be reused in other entities as well:

```
aspect additionalInfo {  
    genre : Genre;  
    language : String(1000);  
}
```

```
entity Books : additionalInfo {  
    key ID : String(3);  
    name: String(3);  
}
```



```
entity Books {  
    key ID : String(3);  
    name : String(3);  
    genre : Genre;  
    language : String(1000);  
}
```

Best practices: Naming Conventions

1. Naming Conventions

- Start *entity* and *type* names with *uppercase letters* – for example, *Books*
- Start *elements* with a *lowercase letter* – for example, *title*
- Use *plural* form for *entities* – for example, *Authors*
- Use *singular* form for *types* – for example, *Genre*

```
entity Books {  
    key ID : UUID;  
    title : String;  
    genre : Genre;  
    author : Association to Authors;  
}  
  
type Genre : String enum {  
    Mystery; Fiction; Drama;  
}
```

Best practices: Using Namespaces & Enterprise features

1. Using Namespaces

- Use namespaces if reusing models
- Services are rarely reused, so rarely need namespaces

```
namespace foo.bar;  
entity Boo {}  
entity Moo : Boo {}
```

..... is equivalent to:

```
entity foo.bar.Boo {}  
entity foo.bar.Moo : foo.bar.Boo {}
```

2. Namespaces vs Contexts

- Use namespaces over top-level contexts

3. Using Enterprise Features

- Localized data
- Authentication & Authorization

Do:

```
namespace foo.bar;  
entity Boo {}  
entity Car {}
```

Don't:

```
context foo.bar {  
    entity Boo {}  
    entity Car {}  
}
```

```
...  
    title : localized String;  
    descr : localized String;  
...
```

Best practices: Avoid over-normalization

Do

```
entity Contacts {  
  key ID : UUID;  
  name : String;  
  emails : array of {  
    kind : String;  
    address : String;  
    primary : Boolean;  
  }  
  phones : array of {...}  
}
```

Don't

```
entity Contacts {  
  key ID : UUID;  
  name : String;  
  emails : Composition of many EmailAddresses on emails.contact=$self;  
  phones : Composition of many PhoneNumbers on phones.contact=$self;  
}  
entity EmailAddresses {  
  contact : Association to Contacts;  
  key ID : UUID;  
  kind : String;  
  address : String;  
  primary : Boolean;  
}  
entity PhoneNumbers {...}
```


Best practices: Favor simple, flat structures

Do

```
entity Contacts {  
    isCompany : Boolean;  
    company : String;  
    title : String;  
    firstname : String;  
    lastname : String;  
    ...  
}
```



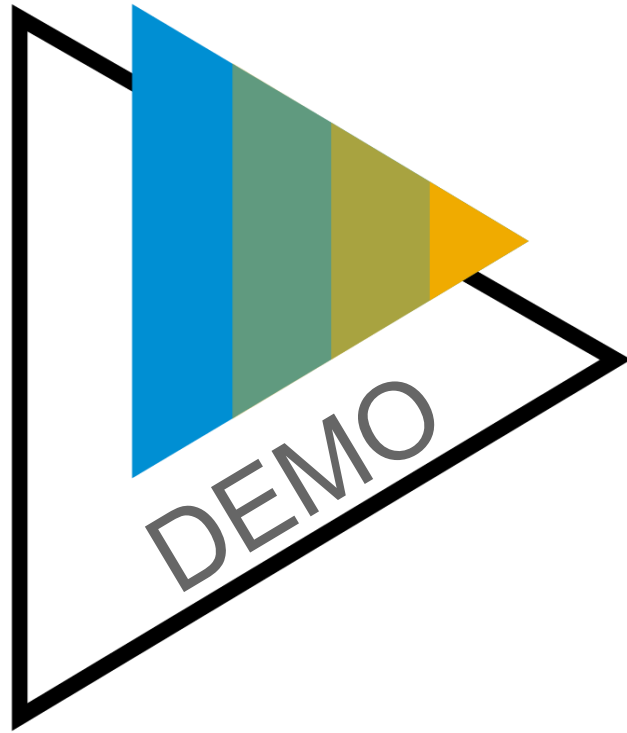
Don't

```
entity Contacts {  
    isCompany : Boolean;  
    companyData : {  
        name : String;  
    }  
    personData : {  
        title : {  
            primary : String;  
            secondary : String;  
        }  
        name : {  
            firstname : String;  
            lastname : String;  
            ...  
        }  
    }  
    ...  
}
```



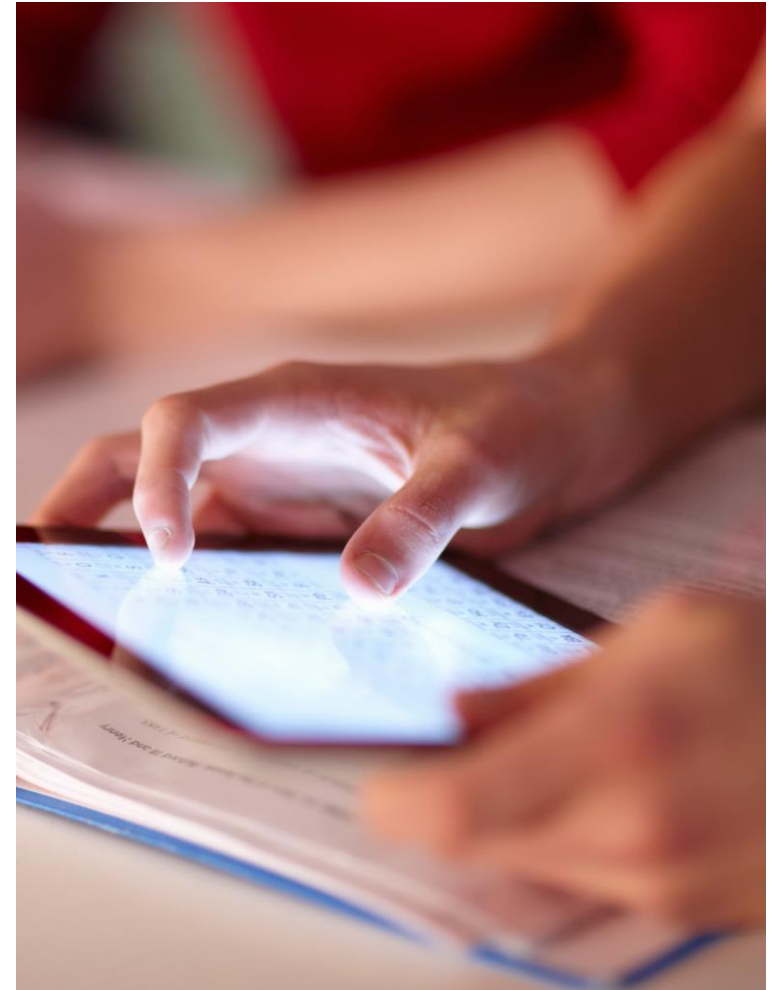
Domain Modeling

Domain models



What you've learned in this unit

- Domain modeling with Core Data Services
- Types, Entities and Associations
- Aspects
- Best Practices in Domain Modeling:
 - ✓ Naming Conventions
 - ✓ Namespaces
 - ✓ Avoid over Normalization
 - ✓ Simple, flat structures
 - ✓ Enterprise features e.g. localized data



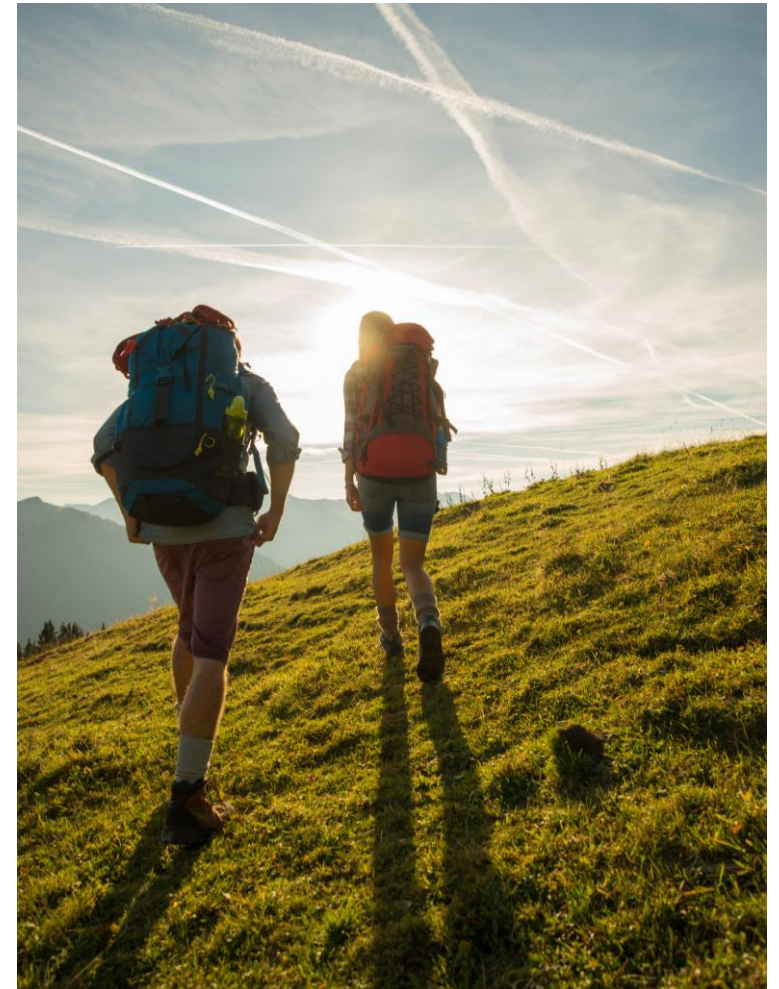
Domain Modeling

Further reading



Additional Material

- [Official documentation](#)
- [Blogs: SAP Cloud Application Programming Model](#)
- [Starter Scenario](#)
- [TechEd Replay](#)
- [Hello World Tutorial](#)



Thank you.

Contact information:

open@sap.com

Follow all of SAP



www.sap.com/contactsap

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See www.sap.com/copyright for additional trademark information and notices.