

connascence

Jan Wedekind

Tuesday, April 29th 2025

motivation

- ▶ metric for software design
- ▶ theoretical foundation for software engineering

talk(s) by Jim Weirich

The Building Blocks of Modularity

Jim Weirich
Chief Scientist
EdgeCase LLC
@jimweirich

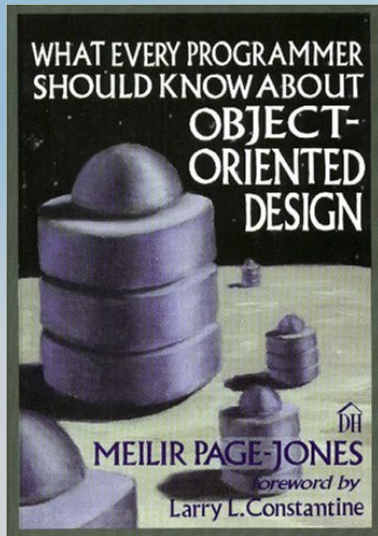
Copyright 2009, Jim Weirich
All Rights Reserved

EdgeCase

software solutions



book by Meilir Page-Jones



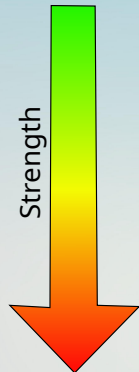
connascence definition

Jim Weirich:

*Two pieces of software share **connascence** when a change in one requires a corresponding change in the other.*

kinds of connascence ordered by increasing strength

	abbreviation	kind
static	CoN	Connascence of Name
	CoT	Connascence of Type
	CoM	Connascence of Meaning
	CoP	Connascence of Position
	CoA	Connascence of Algorithm
dynamic	CoE	Connascence of Execution
	CoTi	Connascence of Timing
	CoV	Connascence of Value
	CoI	Connascence of Identity



Also see <https://connascence.io/>

connascence of name (CoN)

```
def sqr(x):  
    return x
```

```
l = sqr(x) + sqr(y)
```

connascence of type (CoT)

```
from datetime import datetime

def date_to_isostring(date: datetime):
    return date.strftime('%Y-%m-%d')

date_to_isostring(datetime(2025, 4, 2))
```


connascence of meaning (CoM)

```
class Order:  
    def ship_order(self):  
        self.status = 2
```

```
if order.status == 2:  
    print('Order was shipped.')
```

connascence of position (CoP)

```
def imwrite(filename, image):  
    # ...
```

```
imwrite('image.jpg', image)
```

connascence of algorithm (CoA)

```
from werkzeug.security import generate_password_hash
```

```
hashed_password = generate_password_hash(password)
```

```
from werkzeug.security import check_password_hash
```

```
is_correct = check_password_hash(hashed_password, user_input)
```

connascence of execution (CoE)

```
email = Email()  
email.set_recipient('foo@example.com')  
email.set_sender('me@mydomain.com')  
email.send()  
email.set_subject('Hello World')
```

other kinds of connascence

connascence of timing (CoTi)	The timing of the execution of multiple components is important.
connascence of value (CoV)	Several values must change together.
connascence of identity (CoI)	Multiple components must reference the same entity.

properties of connascence

strength	Difficulty of refactoring associated with each kind of connascence.
locality	Closeness of the involved entities (same method, same class, same module, ...).
degree	Number of entities involved in an instance of connascence.

Strong connascence is ok if the locality is high as well.

CoM to CoN transformation

```
class Order:
    def ship_order(self):
        self.status = 2
```

```
if order.status == 2:
    print('Order was shipped.')
```

⇒

```
from enum import Enum
```

```
class Status(Enum):
    NEW = 1
    SHIPPED = 2
```

```
class Order:
    def ship_order(self):
        self.status = Status.SHIPPED
```

```
if order.status == Status.SHIPPED:
    print('Order was shipped.')
```

Transforming strong kind of connascence to weaker kind.

Thanks for listening!