



# MSc thesis in Computer Science

Using Property-Based testing to test MCP Instance-Specifications

Subtitle

**Author**

Søren Lund Hess-Petersen

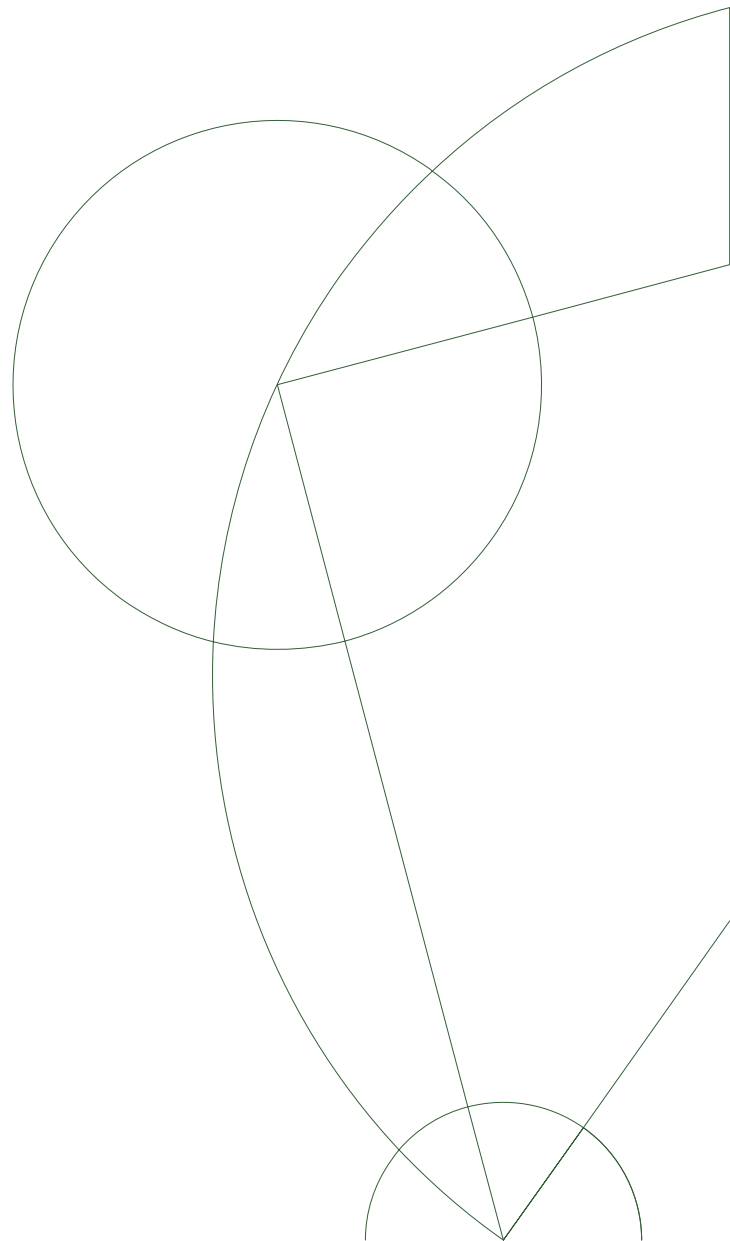
<pws412@alumni.ku.dk>

**Supervisor**

Michael Kirkedal Thomsen

<m.kirkedal@di.ku.dk>

Submitted September 10, 2019



## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
<b>3</b>	<b>Analysis</b>	<b>3</b>
<b>4</b>	<b>Grammar of an Instance Specification</b>	<b>3</b>
<b>5</b>	<b>Implementation</b>	<b>5</b>
<b>6</b>	<b>Results</b>	<b>5</b>
<b>7</b>	<b>Conclusion</b>	<b>5</b>
<b>8</b>	<b>Future Work</b>	<b>5</b>

# 1 Introduction

# 2 Background

# 3 Analysis

# 4 Parser Grammars

## 4.1 ServiceSpecificationScema

ServiceSpecificationSchema ::= specifications

specifications ::= spec specifications  
|  $\epsilon$

spec ::= name  
| status  
| id  
| version  
| description  
| keywords  
| isSpatialExclusive  
| authorInfos  
| requirements  
| serviceDataModel  
| serviceInterfaces

authorInfos ::= authorInfo authorInfos  
|  $\epsilon$

authorInfo ::= spec specifications

requirements ::= requirement requirements  
|  $\epsilon$

requirement ::= spec specifications

serviceDataModel ::= definitionAsXSD

definitionAsXSD ::=  $\epsilon$

serviceInterfaces ::= serviceInterface serviceInterfaces  
|  $\epsilon$

serviceInterface ::= spec  
| dataExchangePattern  
| operations

operations ::= operation operations

```

|  $\epsilon$ 

operation ::= spec
| returnType
| parameterTypes

returnValueType ::= typeReference

parameterType ::= typeReference

4.2 ServiceDesignSchema

ServiceDesignSchema ::= specifications

specifications ::= spec specifications
|  $\epsilon$ 

spec ::= name
| status
| id
| version
| description
| offersTransport
| designServiceSpecifications
| designedBy
| servicePhysicalModel

offersTransport ::= offersTransport offersTransport
| name
| description
| protocol
|  $\epsilon$ 

designsServiceSpecifications ::= designsServiceSpecifications designsServiceSp
| id
| version
|  $\epsilon$ 

designedBy ::= id
| name
| description
| contactInfo
| isCommercial

servicePhysicalDatamodel ::= name
| description
| modelType
| model

```

### 4.3 ServiceInstanceSchema

```

serviceInstanceSchema ::= specifications

specifications ::= spec specifications
                  |  $\epsilon$ 

spec ::= name
      | id
      | version
      | status
      | description
      | keywords
      | URL
      | requiresAutiorization
      | implementsServiceDesign
      | offersServiceLevel
      | coversAreas
      | producedBy
      | providedBy

implementsServiceDesign ::= id
                        | version

offersServiceLevel ::= availability
                  | name
                  | description

coversAreas ::= coversArea coversAreas
            |  $\epsilon$ 

coversArea ::= name
           | description
           | geometryAsWKT

producedBy ::= id
          | name
          | description
          | contactInfo
          | isCommercial

providedBy ::= id
          | name
          | description
          | contactInfo
          | isCommercial

```

**5 Implementation**

**6 Results**

**7 Conclusion**

**8 Future Work**