



Performance Engineering of Service Compositions



THE UNIVERSITY
OF AUCKLAND

NEW ZEALAND

Te Whare Wānanga o Tāmaki Makaurau

John Grundy ^{1,2} John Hosking ¹, Lei Li¹ and Na Liu¹

¹Dept. of Computer Science and

²Dept. of Electrical and Computer Engineering

University of Auckland New Zealand

{john-g, john}@cs.auckland.ac.nz

Outline

- **Background and motivation**
- **Our approach**
- **Example usage**
 - Service composition specification
 - Architectural specification & test bed generation
 - Client load modelling
 - Results generation & analysis
- **Discussion of results**
- **Conclusions**

Background & Motivation

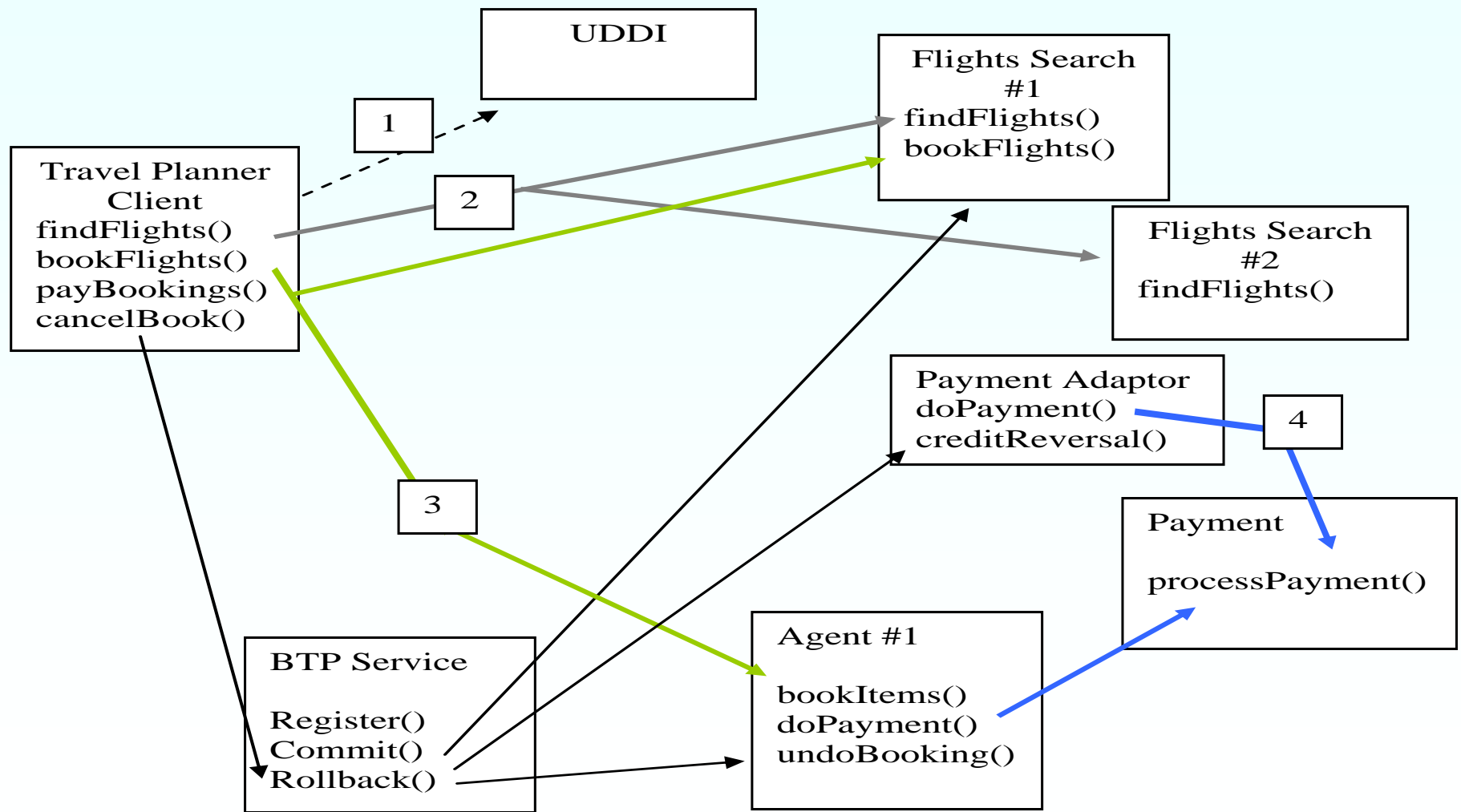
- **Previous work:**

- Specification of test bed generation from architectural specification (SoftArchMTE & ArgoMTE)
- More recent port to Eclipse using our Marama tool set (MaramaMTE)
- Incorporation of client load modelling using Formcharts (CSMR 2006 paper) for web applications

- **This extension**

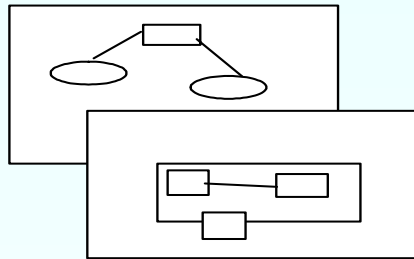
- To permit realistic performance test bed generation of systems based on web service compositions
- Preliminary work – very much proof of concept

Motivating example: travel planner

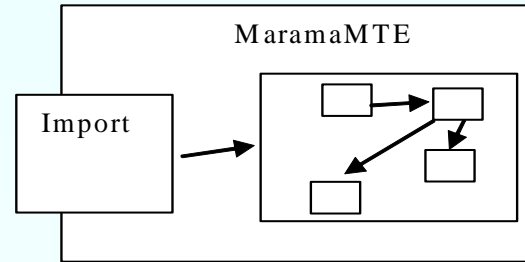


Our approach

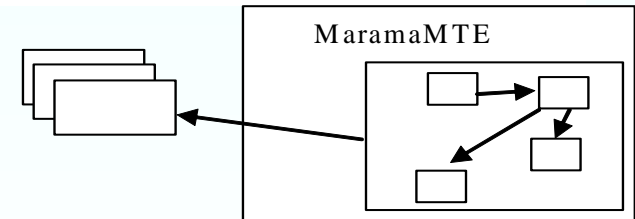
1. Model compositions e.g. in BPMN, ViTABaL-WS



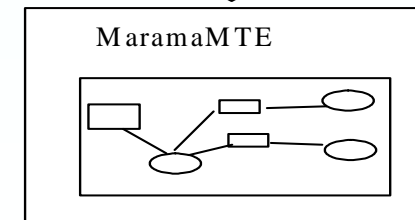
2. Model detailed architecture in MaramaMTE



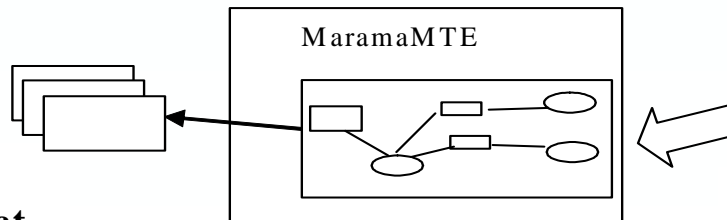
3. Generate service stub code from MaramaMTE



4. Model client loads in MaramaMTE



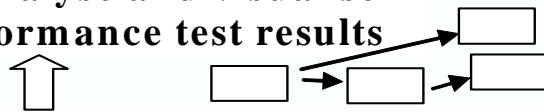
5. Generate client loading tests from MaramaMTE



6. Run generated performance tests & collect results



7. Analyse and visualise performance test results



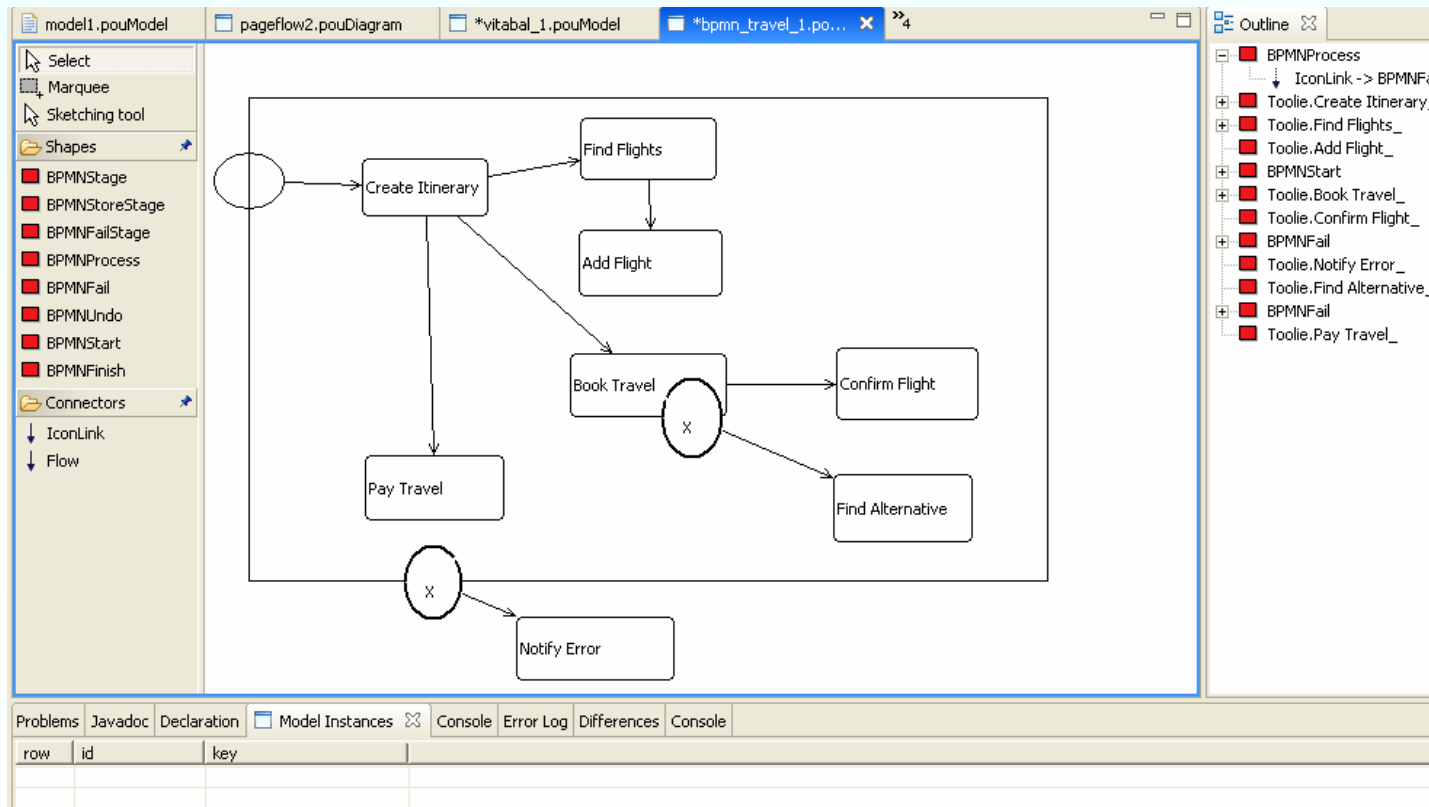
8. Modify compositions and re-test...



Service Composition specification

- **BPMN modeller**

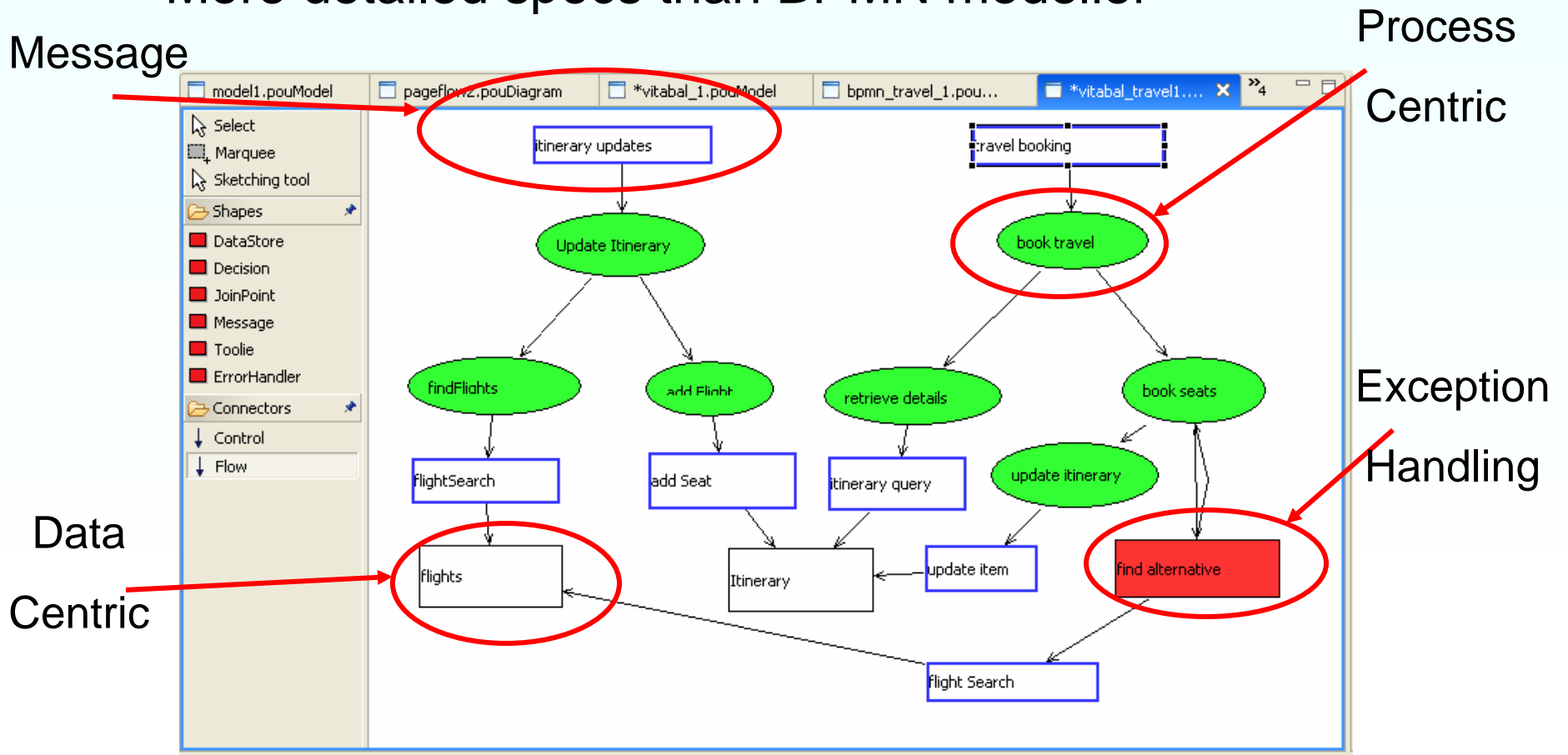
- Conventional process/message flow oriented specification
- Stages = services, composed using process flow
- Protocols & message formats fleshed out in MaramaMTE



Service Composition specification

- **ViTABaL-WS**

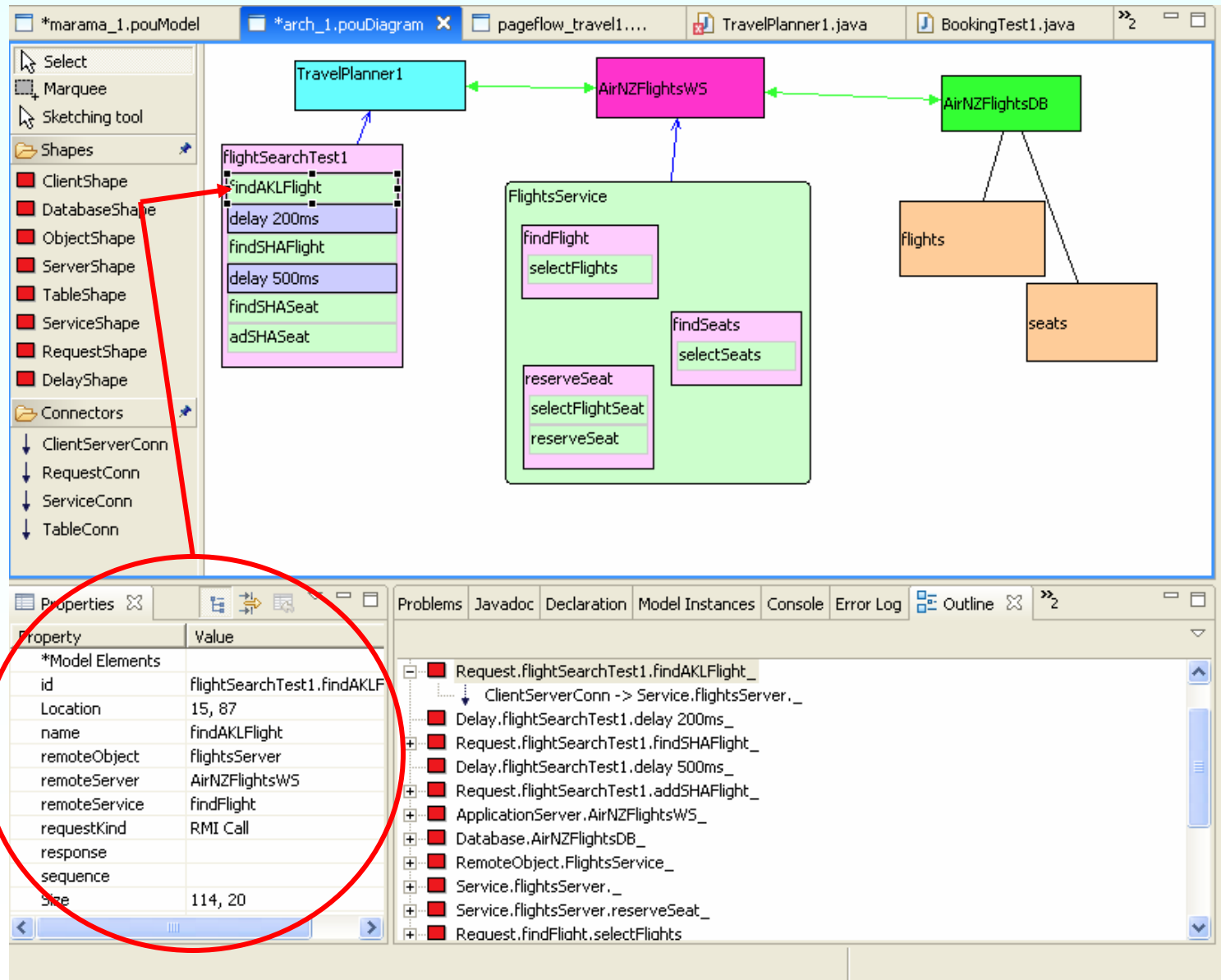
- Uses Tool Abstraction paradigm – event & control oriented
- More detailed specs than BPMN modeller



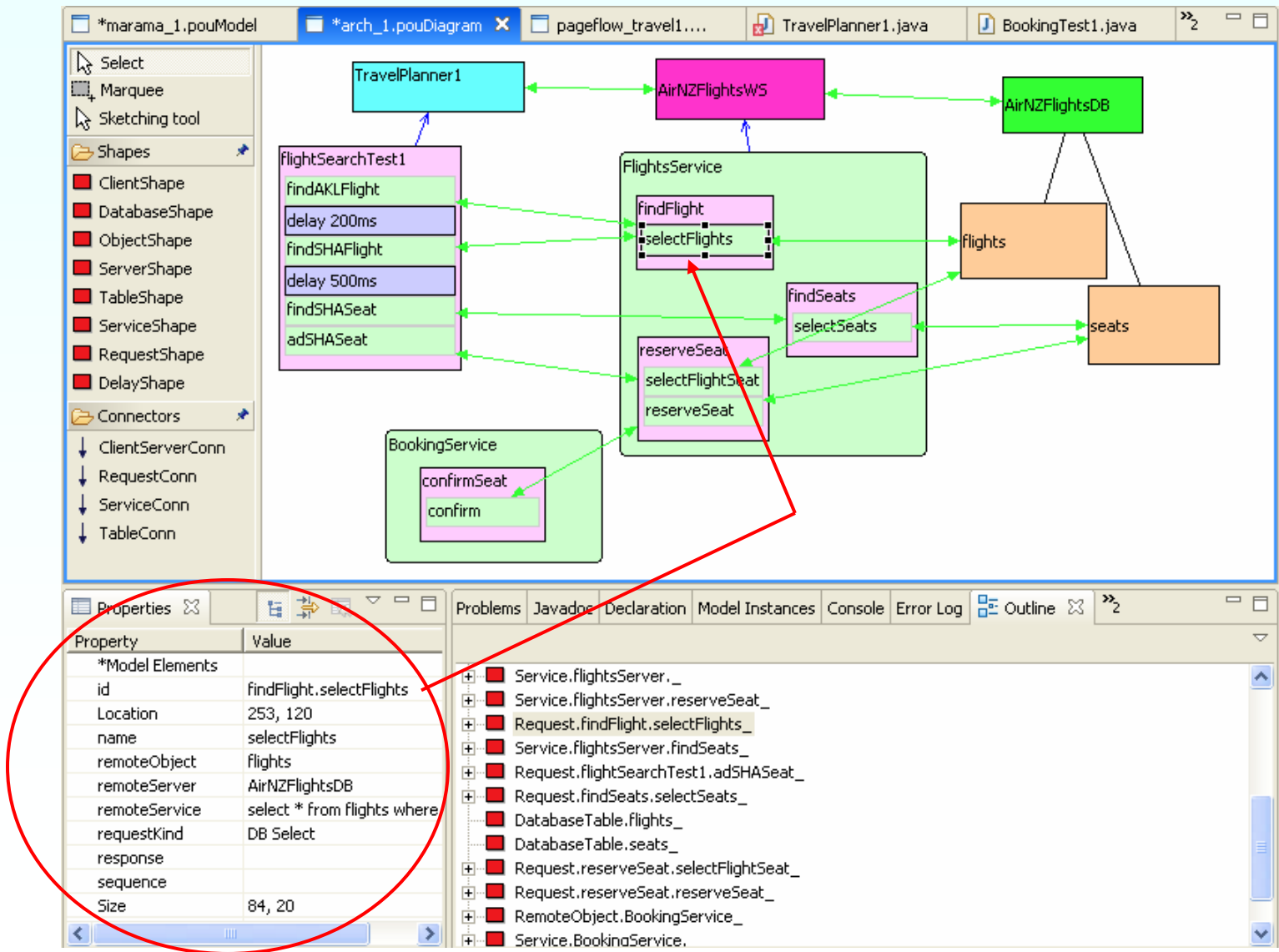
Architectural specification

- **Export models from these two tools**
- **Provides two overlapping architectural models**
- **Assemble a partial architecture model from these**
- **But need to flesh out**
 - Host/port info
 - Message structure details
 - Loading model
- **Get some from available sources eg from WSDL**
- **But need to input rest via MaramaMTE**

Architectural specn: MaramaMTE



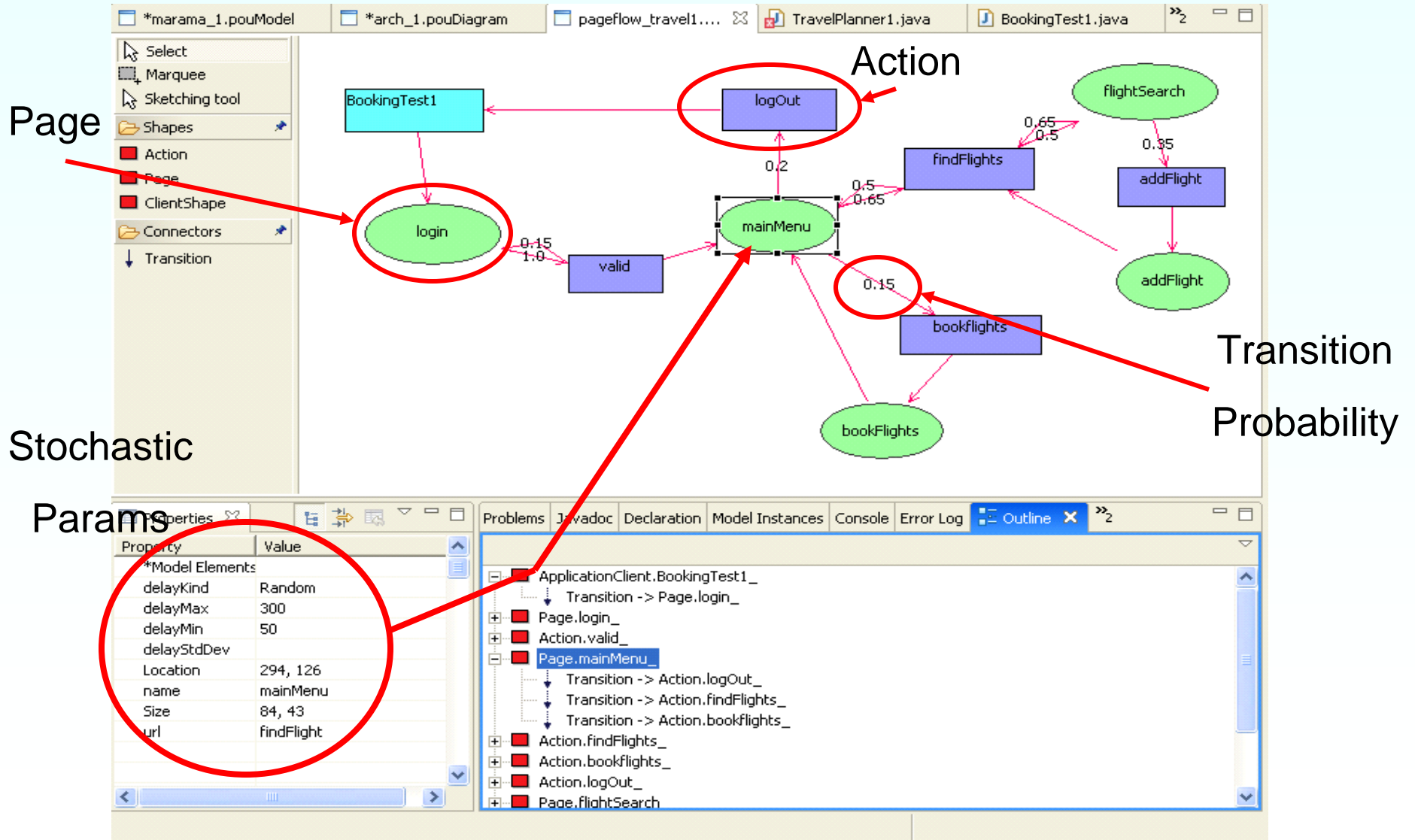
Architectural specn: MaramaMTE



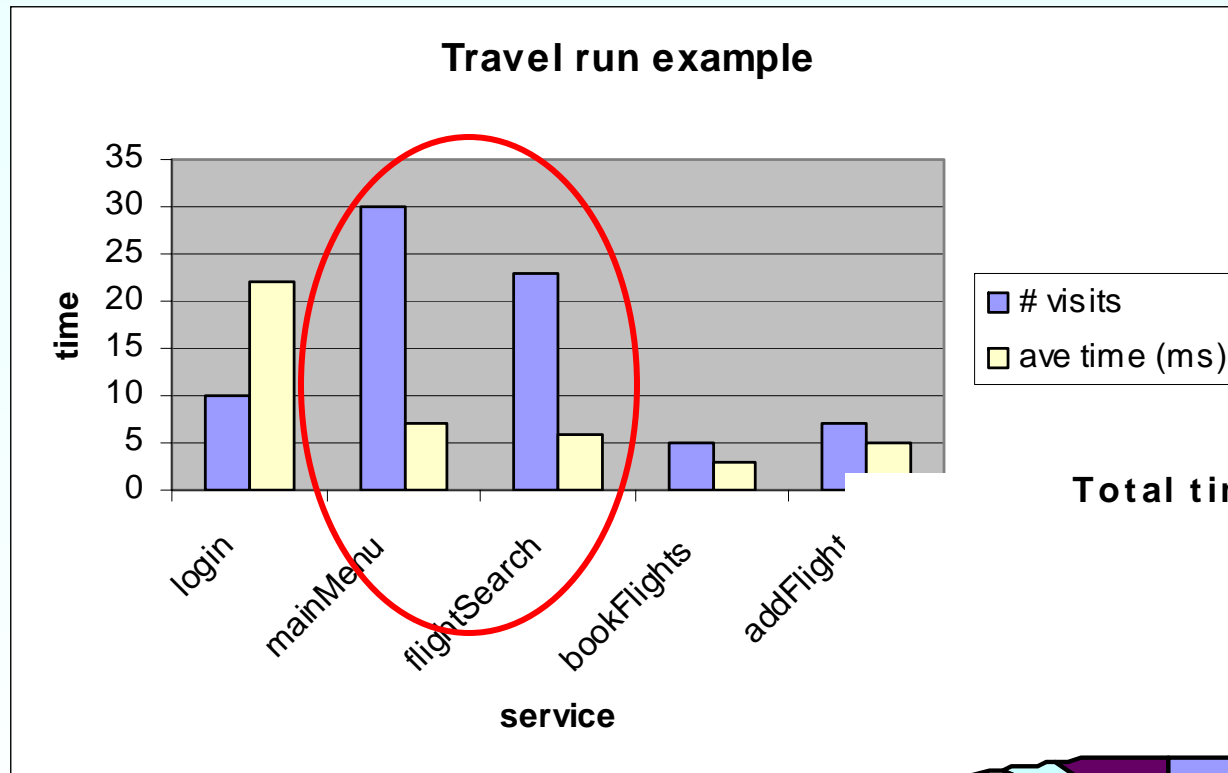
Loading tests

- **MaramaMTE generates testbed plus simple testing schedule**
 - Stubs for services (unless already implemented)
 - Assumes inter-service comms costs much higher than service execution cost
 - Useful for stress testing to obtain maximum throughput performance and concurrent usage
- **But doesn't represent typical service client behaviour**
 - especially if client is a web browser plus user
- **Use Formchart specification to generate more realistic schedules**
 - Can generate multiple concurrent formcharts

Client load modelling via Formcharts

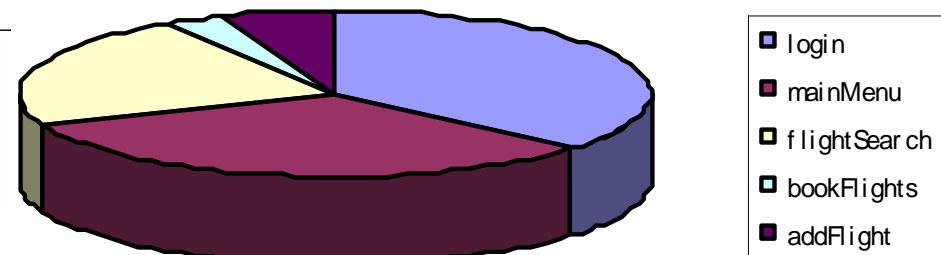


Results visualization



10 concurrent threads
of previous formchart
all services on same host

Total time taken (ms)



Design iteration

- **Can modify process specification & architectural design to generate alternative architectures**
 - Eg merge/split services.
 - Change hosting
 - Caching & transaction policies
- **Can modify modelled service loading to represent different scenarios**
 - Replay historical data
 - Peak versus normal loadings

Conclusions and future work

- **Approach shows promise but very much proof of concept**
- **Only supports static compositions: needs dynamic discovery support**
 - Could model selection stochastically too
- **No use of SLAs/contract specs**
- **Need for “better WSDL” – exploring aspect based repns**
- **Limited set of generated impn technologies**
- **Need for better historical usage support**
- **No support for asynch performance evaln**