Sustainable Software Architecture for NEMo Mobility Platform

Dilshodbek Kuryazov, Andreas Winter, and Alexander Sandau

Abstract

The NEMo project aims at developing a sustainable mobility platform to facilitate *sustainable fulfillment of mobility needs* in rural areas. In order to meet the core objectives of NEMo, the envisioned mobility platform must be sustainable itself by providing flexibility in co-evolution with changing and novel mobility needs, services, and business models. The SENSEI approach proposes a model-driven, service-oriented and component-based means to define mobility services in terms of technology-independent services and their functionalities that are mapped automatically to suitable, reusable software components. This paper describes the application of SENSEI to the NEMo mobility platform, and explains benefits in terms of sustainability by flexible and adaptable software development architecture.

Keywords: Mobility platform, Sustainable software architecture, Sustainable mobility platform, Service-oriented Architecture, Component-based Development

1 Motivation

The interdisciplinary research project NEMo [1] aims at the sustainable fulfillment of mobility needs in rural areas considering social, demographic, accessibility, legal, economic, and ecological conditions and objectives. It intends to facilitate the provision of novel mobility services based on social self-organization, and develop business models that increase utilization of public and private transport, while reducing the overall stream of vehicles on streets. Information and communication technologies (ICT) are viewed as a key enabler to support these objectives by software systems implementing a mobility platform that is accessible through various devices and media. This paper strictly focuses on designing a flexible, adaptable, and sustainable software architecture to provide appropriate ICT support.

Like any software system, the NEMo mobility platform needs to evolve to remain up to date with new or modified requirements, e.g. new business models, mobility services and implementations. Continuously adapting the mobility platform leads to more complex and less maintainable software systems [2]. Due to the innovative nature of the NEMo project, a sustainable and adaptable software architecture plays an essential role in simple and fast development, integration and maintenance of new features.

Independent of the scientific context, the application domain of novel mobility services also addresses to highly flexible software support. The NEMo mobility platform should be able to support all kinds of mobility needs and scenarios, modes of transportation, and business models. It should facilitate the

recombination of existing mobility services to provide enhanced services, as well as completely new, unanticipated usage scenarios [3].

Finally, with the overall goal of NEMo being *sustainability*, it is only appropriate to strive for it in terms of software design. A rigid, monolithic software system would lead to high maintenance costs, and ultimately to its phaseout, close down [4], and forced replacement. To be sustainable, the NEMo mobility platform must make architectural provisions for sustainability, flexibility and adaptability [5].

A major use case, that is expected to play an essential role for the NEMo mobility platform, is *inter-modal routing* combining the different modes of transportation e.g. walk, bike, bus, train, car pooling, etc. The existing infrastructure and functionality of the ICT Platform and ICT Services projects [8] is already able to support the use case of inter-modal routing to a large extent. On top of this platform, the ICT Services project build another software system to combine its basic software services, and offer value-added services to support the designated business processes. These mobility services are reused in developing a sustainable prototypical mobility platform in Section 3.

The current infrastructure has not been designed with a particular focus on flexibility, sustainability and adaptability. The business processes such as for inter-modal routing, are “hard-wired” into the software system of ICT Services, and the system is therefore hard to maintain, adapt and extend. This is not a good fit for NEMo according to its sustainability objective and the need for the mobility platform to sustain the project’s progress, innovative nature, and evolving requirements. The sustainability objectives of the NEMo project, from the ICT point of view, are twofold:

- Facilitate existing, and future changes, by providing architectural foundations that incorporate the existing functionality, but highlights flexibility, adaptability, and long-term sustainability;
- Reuse, enhance, and modify the existing functionality based on the research findings within the NEMo project, to support, for example, novel business models, model-driven and service-oriented mobility services and component-based functionality.

In NEMo, the solution proposed to accomplish these objectives is adopting the SENSEI approach for building the mobility platform. So far, some prototypical results are achieved in providing the existing mobility platform with the flexible, sustainable and adaptable software support based on component-based, service-oriented software development and maintenance. This paper presents the results of applying the SENSEI approach [6] to the design and development of the NEMo mobility platform.

SENSEI *(Software EvolutioN SErvice Integration)* provides *service-oriented* software design facilities (*service orchestration*) on an abstraction level close to the application domain of mobility needs and services. Strictly separated from this layer, concrete implementations of these services are realized in the *component-based* terms, to promote reuse. An automated mapping from services to components bridges the gap between service-oriented specification and component-based realization.

The remainder of this paper is outlined as follows: Section 2 introduces the NEMo taxonomy which is an architectural foundation for sustainable mobility platform development, and gives the example of a

usage scenario for the mobility platform. In Section 3, the central concepts of the SENSEI approach are described and the application of SENSEI to the NEMo mobility platform is sketched. Section 4 explains the current state of the state-based interaction support for the NEMo mobility platform. Section 5 finalizes this paper by drawing some conclusions.

2 NEMo Taxonomy

One of the main objectives of NEMo was initially to develop a sustainable software architecture. Figure 1 depicts the four-layer NEMo taxonomy [7] which is proposed as a sustainable software architecture in the framework of the NEMo project. It is an abstract sustainable architecture which serves as a common blueprint and foundation in developing the sustainable NEMo mobility platform. The taxonomy constitutes clear separation of concerns by distinguishing between the *business models and processes*, *IT services* and *IT components* of *mobility services*.

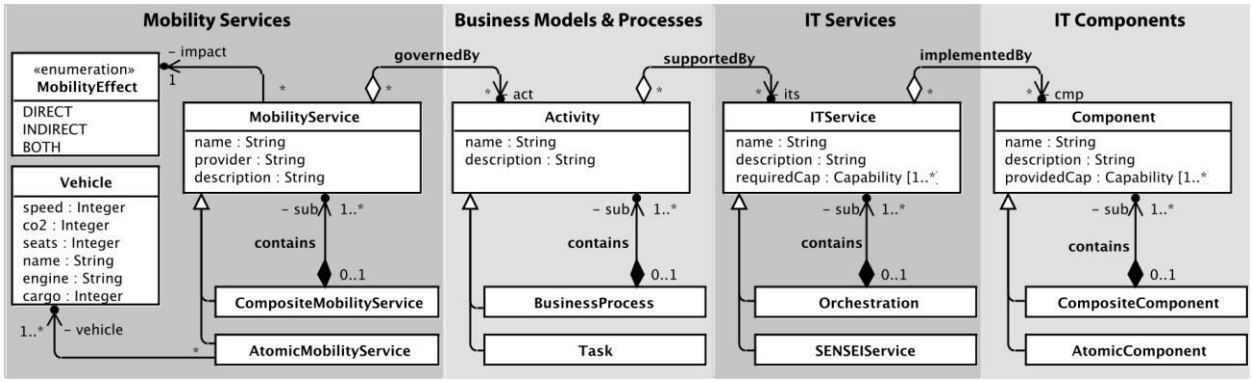


Figure I. NEMo Taxonomy [7]

As a major use case for the mobility platform, the *inter-modal routing* scenario is explained throughout this paper according the four layers of the NEMo taxonomy in Figure 1. In Section 3, this architectural foundation is realized combining the model-driven, service-oriented and component-based flexible approaches. Section 2.1 portrays the simplified example of the inter-modal routing use case of the mobility platform, however it conveys the general idea. Section 2.2 models the business process for that mobility service. Section 2.3 defines the IT-services and Section 2.4 explains the IT-components that are needed for this exemplary inter-modal routing use case.

2.1 Mobility Service

In general, the mobility platform offers *mobility services* by means of transportation (*Vehicle*) and provided by *providers*. A mobility service can also be the combination of *atomic mobility services* which is referred to as *composite mobility service*. This is needed because of the inter-modality nature of mobility services. Any mobility services can be performed directly by transporting people, indirectly by transporting things, or both by transporting people and things at once. Each mobility service may consist of several *business models and processes*.

The inter-modal routing serves to find routes: given a point of origin and a destination, it should provide possible routes. Combining different modes of transport, e.g. walking, riding a bike, taking a bus or train, driving a private car, or joining a car pool, makes this *inter-modal routing*.

Figure 2 depicts the user interface of the simple inter-modal routing use case representing two different states of user interaction with the mobility platform. In the first window, the *origin* and *destination* of a route and *transport modes* can be given and the possible routes can be searched for between these locations. In this example, BUS is selected as the transport mode.

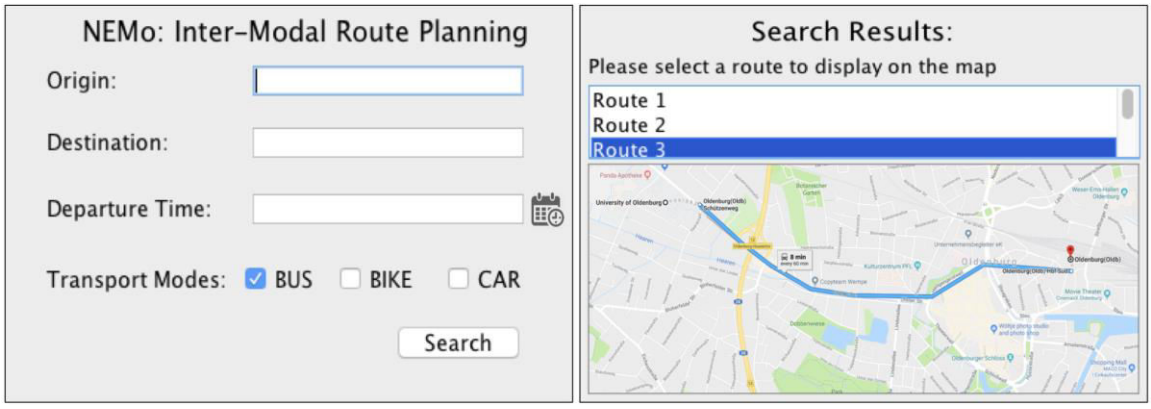


Figure II. Exemplary Mobility Service UI

Once the *search* button is clicked, the second window (*Search Results*) lists all possible routes. When any of these routes is selected, the selected route is drawn on the map right below. Section 2.2 presents the business process for this use case.

2.2 Business Models and Processes

In the second column, the taxonomy describes *business models and processes* that might be related to many mobility services. The business models and processes consist of *activities* performed by the users of mobility platform. The activities can further be defined as *tasks* that the users should perform before, while and/or after using the mobility service. The activities are supported by *IT-services*.

The UML activity diagram depicted in Figure 3 displays the possible breakdown of the use case explained in Section 2.1 into process steps. The *Client swim-lane* consists of the two states of the user interface shown in Figure 2. A client wishing to travel inter-modally requests a route to the destination. The *System swim-lane* represents all activities done by the mobility platform to extract and deliver all necessary route information.

The mobility platform first finds reasonable stopovers to possibly change the mode of transport. Then, available mobility services, providing different transportation means, are investigated to instantiate the individual sub-routes. Eventually, the mobility platform returns route indicators in which they are reused to extract route details by another additional service. The results are integrated into complete traveling itineraries by route concatenation, and returned to the user who chooses the best route from them.

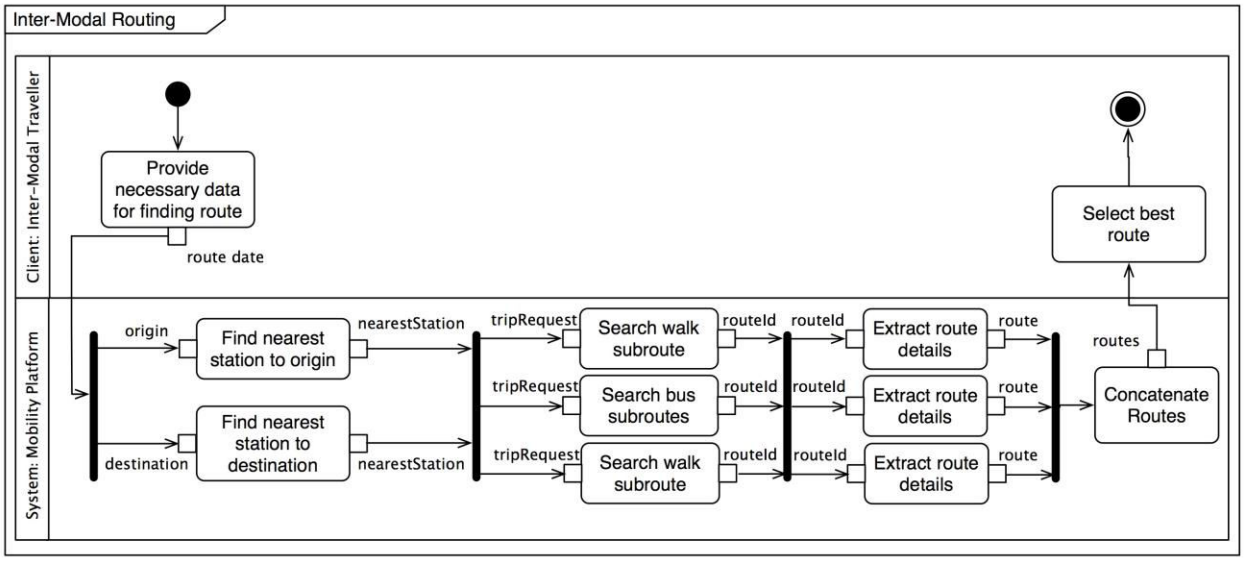


Figure III. Inter-Modal Routing Activities

2.3 IT-Services

An IT-service defines a piece of software functionality. It adds appropriate functionalities to the activities focusing on human behavior [6]. An IT-service is a description of how a software component should work.

According to the business model explained in Section 2.2, the following IT- services are necessary and thus, defined.

- *Find nearest stations.* This service is used to find the list of nearest stations to the given location. The amount of stations can be restricted by providing the radius in kilometers around the given location.
- *Search sub-routes.* This is actual mobility service which is utilized for finding routes for different transport modes. In the exemplary business model, three instances of this service are used with different transport modes.
- *Extract route details.* In the existing ICT mobility platform, mobility services return sub-set of route data. Detailed information including stops between the given two locations is then extracted by an additional service.
- *Concatenate Routes.* After finding all possible sub-routes, they are integrated into coherent instructions for the whole trip by this service.

These are main mobility services that are necessary for fulfilling the business model defined in Section 2.2. Section 2.4 explains how these services are realized by relevant IT-components.

2.4 IT-Components

IT-components are the concrete implementations of the functionality defined by IT- services [6]. An atomic service can be implemented by several components. For instance, a *find route* service can be implemented by several components because of the different transportation means e.g. bus, walk, etc.

As long as some mobility services and functionality are already developed as the outcome of the Electric Mobility Showcase program [8], these existing functionality of ICT Services and ICT Platform are invoked as components in the context of NEMo.

3 SENSEI Applied

The *NEMo taxonomy* explained in Section 2 serves as a common blueprint for model-driven [9], service-oriented and component-based development and maintenance of the NEMo mobility platform. In order to achieve the overall goal of NEMo being *sustainability*, this section applies SENSEI approach to the NEMo mobility platform making the NEMo taxonomy as the central architectural provisions for developing model-driven, service-oriented and component-based mobility platform.

3.1 SENSEI Approach

SENSEI [6] provides a tool-chain support for building a framework that increases flexibility, adaptability and sustainability. SENSEI combines service-oriented, component-based, and model-driven techniques to automatically map high-level, process models (*service orchestrations*) onto reusable and interoperable components possessing the required capabilities, and generate code that combines and coordinates them in the required manner.

In SENSEI, clear separation of concerns is established by distinguishing *services* abstracting from technologies and interoperability issues on the one hand, and concrete implementations in the form of *components* on the other hand. SENSEI consists of the following core concepts:

- The *service catalog* serves as a central repository containing service definitions that are described in a standardized means. Section 3.2 defines a catalog of the mobility services identified in Section 2.3.
- *Service orchestrations* allow to combine services from the service catalog to create more complex functionality, using a process-oriented, graphical modeling language. Section 3.3 explains service orchestrations that are required for supporting the mobility service defined in Section 2.1.
- The *component registry* (Section 3.4) establishes relations between services defined in the service catalog, and components that provide the functionality provided by them.

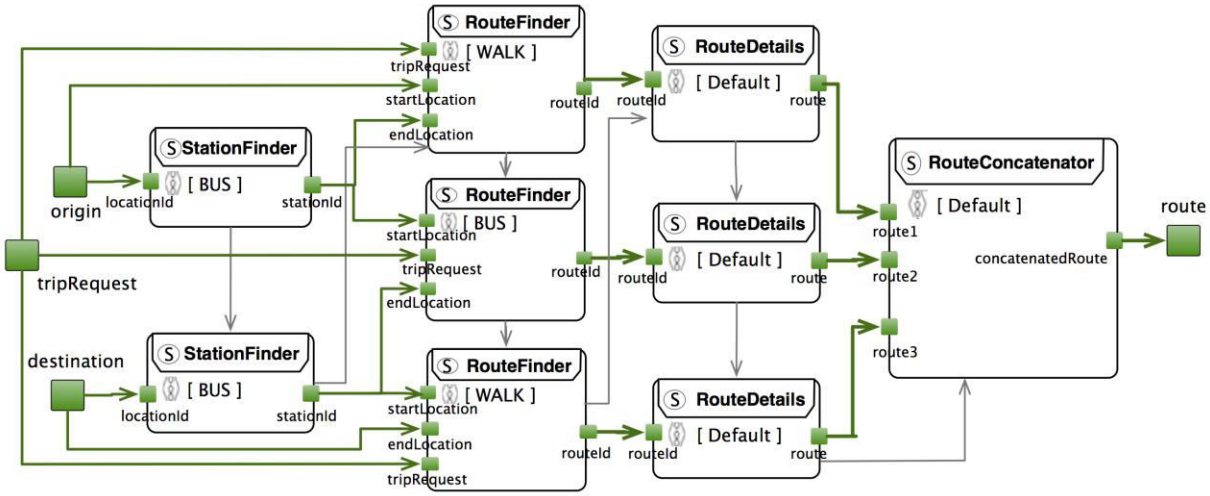
3.2 Service Catalog

In order to define the mobility services in the SENSEI service catalog, the four mobility services explained in Section 2.3 are conditionally named as follows: *StationFinder*, *RouteFinder*, *RouteDetails* and *RouteConcatenator*.

All services defined in the service catalog have names and descriptions, along with input and output parameters, and associated data types, also modeled in the catalog as data structures. The implementations of the *RouteFinder* service cannot usually be expected to be able to provide traveling information for *all* imaginable modes of transport, and the users of the service might only need a subset of them. At the same time, the service catalog would become extremely cluttered if a service was to be defined for every possible variant. SENSEI solves this by introducing *capabilities* to describe service variants concisely. Services define *capability classes* to represent aspects that can vary independently.

3.3 Service Orchestrations

In order to fulfill business models and processes, IT-services are orchestrated using one or many atomic IT-services from the service catalog. For instance, different IT-services provide different routing services by different transport modes and capabilities, whereas they are orchestrated to provide the complete



inter-modal routing.

Figure IV. Service Orchestration

The manual steps such as defining services and registering components are supported by the SENSEI editor, as well as service orchestrations (e.g. Figure 4) are modeled graphically on the SENSEI editor. For this, services are instantiated from the catalog by selecting the *required capabilities*. In orchestrations, the invocation order of services defined by *control flows* (gray arrows) and flow of data among services is defined *data flows* (green arrows) connecting inputs and outputs (green boxes) of services. Services are marked with an encircled "S" ahead their names. The input parameters of the overall orchestration are defined by bigger green boxes, e.g. three boxes on the most left side of Figure 4 with names *origin*, *destination* and *tripRequest*, one green box named *route* on the most right side.

3.4 Component Registry

Services are *implemented* by components. This relationship is persisted in the component registry, which can be thought of as a table. Each entry refers to a component and lists one or more services it

implements. With each service, the *provided capabilities* are specified in exactly the same way as required capabilities for orchestrations. Existing mobility services provided by the ICT Services are wrapped and for SENSEI compatibility and interpreted as implementations behind mobility services in orchestrations. These components are published as REST Web services and made available to reuse in the framework of NEMo.

To conclude the process, the SENSEI orchestration model is mapped to a particular target platform that provides a runtime environment, such as WSO2, the middleware used by the ICT Platform project. This step is performed by a model-driven code generator. It results in a fully auto-generated, new component called composer. The composer itself implements the orchestration logic, i.e. routes data according to the data flows, and invokes components in the order dictated by control flow.

3.5 Sustainability

SENSEI service catalog, orchestrations and component registry (e.g. depicted in Figure 4) are quite flexible and adaptable providing sustainable mobility platform in NEMo. This section presents the extension of the service orchestration depicted Figure 4 how it can be sustained, adapted and extended.

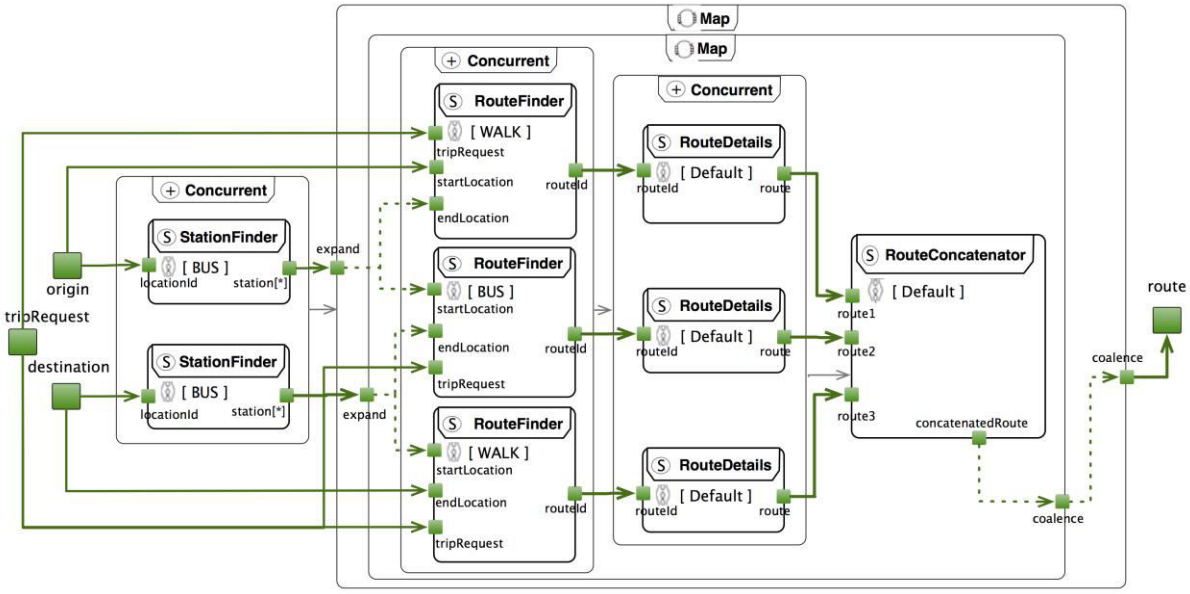


Figure 5 depicts the same service orchestration model with some extensions.

Figure V. Service Orchestration with Concurrency and Map

In this orchestration, the following extensions are done: (1) the service instances *StationFinder*, *RouteFinder* and *RouteDetails* are surrounded by concurrency service containers entitled as *concurrent*, (2) the output parameter of *StationFinder* service is changed to return the list of stations, (3) the part of the orchestration except station finders is surrounded with map iterations for iterating overall list of bus stop (the station finder in Figure 4 return only one nearest stop, whereas the station finder in this orchestration returns many).

These changes and extensions are made on the orchestration editor without affecting underlying implementations and any other artifacts. This allows for developing model-driven, service-oriented, component-based sustainable mobility platform. Modeling service orchestrations allows to remain abstract from technical implementation and does not require programming skills or expert knowledge of the diverse technologies used by components.

All service instances included in concurrency boxes are executed in parallel. The concurrent service invocation is another main contribution of applying the SENSEI approach to the NEMo projects which is not provided by the current implementations. It improves overall performance of service orchestration.

4 Future Work: Interactivity

The NEMo mobility platform requires an interactivity support for performing user interactions with the mobility platform. Since interactivity support is usually only anchored in the source code implicitly, both the implementations of interactive user interfaces and program logic become blurred. Thus, sustainability of interactivity support becomes a challenge to subsequently maintain overall interaction.

The students project group DORI (Do Your Own Reactive Interface) intends to provide tool support for designing state-based interactivity models to describe the overall interaction by GUI *states* and *transitions* between GUI states. It takes advantage of the SENSEI approach for defining abstract GUI widgets and their underlying implementations. For instance, Figure 2 depicts two interaction states with the mobility platform. Its initial state consists of widgets instances such as *input text* fields for providing trip request information and *button* to perform background logic. The second state depicts a *list* and *map* widgets to select from and show the result list of routes. In DORI, all widgets in these states are defined in an abstract widget catalog and all widgets in these two states are the widget instances of abstract widgets defined in that abstract widget catalog.

In DORI, the transitions of states including control and data flows can be modeled using *state transitions*. For instance, once the *Search* button is clicked in the initial state, the given route date is passed to a program logic of orchestration depicted in Figure 5. Eventually, all possible routes are displayed in the second state of interactivity.

In the DORI interactivity modeling approach, the state-based interactivity models can initially be designed for Web interactions, which uses Java Server Faces for the user interfaces and REST services for the program logic. Moreover, interaction models can also be designed and interpreted for the Android platform. The modeling language designed by the project group is based on UML state diagrams and IFML (Interaction Flow Modeling Language) diagrams.

5 Conclusion

This paper has demonstrated the application of the SENSEI approach to develop a sustainable and flexible mobility platform for the NEMo project. The clear separation of concerns i.e. services and components in SENSEI allows to specify application behavior on a non-technical level, close to the application domain. Service orchestrations are comparatively easy to adapt or extend, and the

corresponding software application can be re-generated, allowing for fast turnarounds, and resulting in a high degree of flexibility. The use of SENSEI reduces the effort required to develop and maintain the mobility platform, particularly when sustainability raises.

The only prerequisite is the basic functionality to be already available in the form of components provided by the ICT Platform. The component-based structure supported by SENSEI promotes building up the catalog of both services and components, so that over time existing functionality can be readily reused, adapted, extended or new ones can be added. Both aspects potentially increase productivity and serve as the basis for sustainable mobility platforms.

References

[1] J. Jelschen, C. Küpker, A. Winter, A. Sandau, B. Wagner vom Berg, and J. Marx Gómez, "Towards a Sustainable Software Architecture for the NEMo Mobility Platform," in *Environmental Informatics – Stability, Continuity, Innovation*, V. Wohlgemuth, F. Fuchs-Kittowski, and J. Wittmann, Eds. Herzogenrath: Shaker, 09.2016, pp. 41–48.

[2] M. Lehman, "Laws of software evolution revisited," in *European Workshop on Software Process Technology*. Springer, 1996, pp. 108–124.

[3] B. Combemale, B. Cheng, A. Moreira, J. M. Bruel, and J. Gray, "Modeling for Sustainability," in *Modeling in Software Engineering 2016 (MiSE'16),* ACM, 2016.

[4] V. Rajlich and K. Bennett, "A Staged Model for the Software Life Cycle," *Computer*, vol. 33, no. 7, 2000, pp. 66–71.

[5] R. Kateule and A. Winter, "Architectural Design of Sensor based Environmental Information Systems for Maintainability," in *Nachhaltige Betriebliche Umweltinformationssysteme*. Springer, 2018, pp. 87–96.

[6] J. Jelschen,"Service-oriented Tool-chains for Software Evolution," in *Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA), 2015 IEEE 9th International Symposium*. IEEE, 2015, pp. 51–58.

[7] A. Akyol, J. Halberstadt, K. Hebig, J. Jelschen, A. Winter, A. Sandau, and J. Marx Gómez, "Flexible Software Support for Mobility Services," *INFORMATIK 2017*, 2017.

[8] *B. Wagner vom Berg, F. Köster, J. Marx Gómez,* "Förderung der Elektromobilität durch innovative Infrastruktur- und Geschäftsmodelle," In: Automotive Services, M. Leimeister, H. Krcmar, H. Hoffmann, M. Schermann (Hrsg.), Books on Demand GmbH, Norderstedt, 2010.

[9] A. G. Kleppe, J. B. Warmer, and W. Bast, "MDA explained: the model driven architecture: practice and promise," *Addison-Wesley Professional,* 2003.