

Engineer BAINOMUGISHA

Abstract

We are on the verge of a post-PC revolution where computing power is shifting from laptops and desktops to smartphones, tablets and everyday objects such as eyeglasses and coffee machines. Those computing objects come equipped with a myriad of context sensors such as an accelerometer, a proximity sensor, a gyroscope, a GPS receiver and NFC technology. Context sensors enable the development of context-aware applications that continuously adapt their behaviour to match the context of use. In this dissertation, we refer to applications that are always prepared to promptly adapt their behavior in reaction to context changes as reactive context-aware applications.

Our research hypothesis is that developing reactive context-aware applications remains notoriously difficult because of the lack of suitable programming language abstractions coupled with the unpredictable nature of context changes. More concretely, current programming languages fall short of providing the appropriate support for developing context-aware applications that need to react promptly to a sudden context change – especially if such a context change occurs in the middle of an ongoing procedure execution.

Such a context change may require an ongoing procedure execution to be promptly interrupted in order to prevent its execution from happening in a wrong context. The vision of this dissertation is to investigate novel programming language abstractions to facilitate the development of reactive context-aware applications. We propose a new programming language model called interruptible context-dependent executions, where a procedure execution is always constrained to happen only under a developer specified context condition. In this model, the execution of a context-dependent procedure can be seamlessly interrupted or resumed depending on whether the specified context condition is satisfied or not. The language runtime ensures that the execution state of a context-dependent procedure is automatically preserved between interruptions.

The model features a new dispatching mechanism called reactive dispatching that continually takes into account new context changes to select the applicable procedures and to suspend or interrupt ongoing procedure executions.

We present a proof-of-concept programming language called Flute that incorporates the interruptible context-dependent executions model. Flute has been implemented as a meta-interpreter on top of iScheme – a technological research artefact that we developed to enable the experimentation with novel language abstractions and features to ease the development of reactive context-aware applications. iScheme runs on the iOS mobile platform and as such it fosters experiments on real mobile devices.

Samenvatting (Abstract in Dutch)

We bevinden ons aan het begin van een ware post-PC revolutie waarin rekenkracht verschuift van computers naar tablets, naar mobiele telefoons en zelfs naar alledaagse objecten zoals brillen en koffiemachines. Deze objecten zijn voorzien van verschillende sensoren voor context zoals versnellingsmeters, nabijheidssensoren, gyroscopen, ontvangers voor GPS-signalen en NFCtechnologie.

Deze sensoren maken het ontwikkelen van context-bewuste applicaties mogelijk die hun gedrag voortdurend afstemmen op de context waarin ze gebruikt worden. In dit proefschrift verwijzen we naar applicaties die op elk moment ten gevolge van contextveranderingen hun gedrag prompt kunnen aanpassen als reactieve context-bewuste applicaties. Onze onderzoekshypothese is dat het ontwikkelen van reactieve contextbewuste applicaties notoir moeilijk blijft door het gebrek aan geschikte programmeertaalabstracties en de onvoorspelbare aard van contextveranderingen.

Bestaande programmeertalen bieden geen geschikte ondersteuning voor het ontwikkelen van context-bewuste applicaties die prompt moeten reageren op een plotse contextverandering –vooral wanneer deze zich voordoet tijdens de uitvoering van een procedure. Bij zulk een contextverandering kan het nodig zijn een lopende uitvoering te staken teneinde te voorkomen dat een procedure uitgevoerd wordt in de verkeerde context.

De visie van dit proefschrift is het ontwikkelen van reactieve contextbewuste applicaties te ondersteunen via nieuwe programmeertaalabstracties. Hiertoe stellen we een nieuw programmeertaalmodel voor genaamd onderbreekbare context-afhankelijke uitvoeringen waarin een door de ontwikkelaar opgegeven contextvoorwaarde de uitvoering van een procedure beschermt.

Hierbij wordt de uitvoering van een context-afhankelijke procedure onderbroken wanneer de contextvoorwaarde niet langer voldaan is en weer naadloos hervat van zodra dit terug het geval is. De runtime van de programmeertaal zorgt ervoor dat de uitvoeringsstaat van een context-afhankelijke uitvoering automatisch bewaard blijft tussen de onderbreking en de hervatting. Kenmerkend voor dit programmeermodel is een nieuw dispatching mechanisme genaamd reactive dispatching dat contextveranderingen in acht neemt bij het selecteren van nieuwe uit te voeren procedures en bij het onderbreken of hervatten van lopende procedures.

Als bewijs van concept stellen we de programmeertaal Flute voor die het programmeermodel van onderbreekbare context-afhankelijke uitvoeringen incorporeert. Flute is gerealiseerd als een meta-interpreter in iScheme – een onderzoeksartefact dat we ontwikkeld hebben om het experimenteren met innovatieve programmeertaalabstracties voor reactieve context-bewuste applicaties te vergemakkelijken. iScheme bevordert experimenten op mobiele apparaten via diens ondersteuning voor het iOS platform.