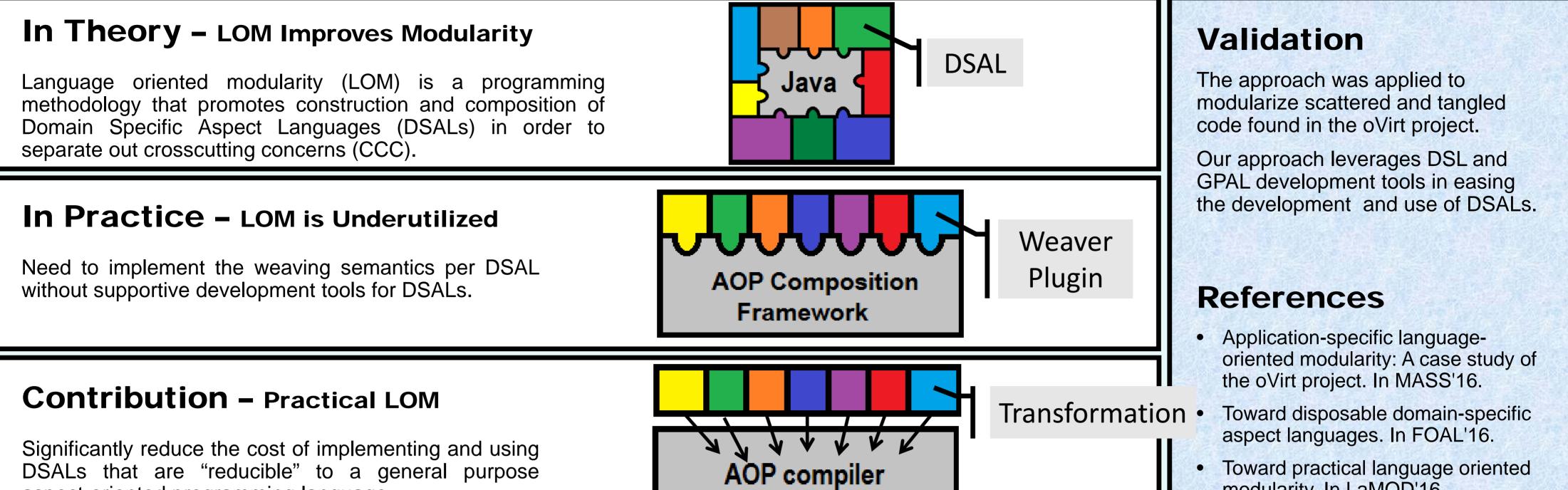
Language Oriented Modularity: From Theory to Practice Arik Hadas Open University of Israel

Abstract – Language Oriented Modularity (LOM) is a methodology that is based on Language Oriented Programming (LOP) but applied to Domain-Specific Aspect Languages (DSALs) rather than Domain-Specific Languages (DSLs). LOM involves the construction, composition, and use of DSALs on-demand during the software modularization process, in order to separate out crosscutting concerns (CCC). However, LOM is underutilized and often not used at all in modern projects. The goal of this research is to improve the applicability of LOM for resolving crosscutting concerns in real world projects.



Xcutting Problem LOM Solution Approach c class MigrateVmCommand < T extends MigrateVmParameters > ... { - - ★AuditLogGenerator.xtend ☆ private VDS destinationVds; 📄 ovirt.audit 🖾 **DSAL with easy-to-use** private EngineError migrationErrorCode vate Integer actualDowntime; 249 class AuditLogGenerator implements IGenerator { public MigrateVmCommand(T parameters) { ... } 10logs for org.ovirt.engine.core.bll.MigrateVmCommand (overrides): public MigrateVmCommand(T migrateVmParameters, CommandContext cmdContext) { ... } syntax for auditing private Resource resource; case success & isReturnValueUp rotected LockProperties applyLockProperties(LockProperties lockProperties) { ... } log VM MIGRATION DONE public final String getDestinationVdsName() { ... } public String getDueToMigrationError() { ... } override void doGenerate(Resource resource, IFileSystemAccess fsa) { △276 case success & internal 40 var path = 'org.ovirt.engine.core.bll.'.replaceAll('\\.', File.separator) + 'Logs.aj' 28 tected VDS getDestinationVds() { ... } fsa.generateFile(path, resource.compile) log VM_MIGRATION_START_SYSTEM_INITIATED, 29 00verride @Override protected void processVmOnDown() { ... } protected boolean initVdss() { ... } case success log VM_MIGRATION_START, private List<Guid> getDestinationHostList() { ... } **Synchronization Transforming DSAL code** def compile(Resource resource) { case failure & isHostInPrepareForMaintenance 89 protected void executeVmCommand() { ... } this.resource = resource private boolean perform() { ... } private boolean migrateVm() { ... } log VM_MIGRATION_FAILED_DURING_MOVE_TO_MAINTENANCE, 111 to a GPAL aspect case failure 100 package org.ovirt.engine.core.bll; private MigrateVDSCommandParameters createMigrateVDSCommandParameters() { 11 log VM MIGRATION FAILED public void runningSucceded() { ... } protected void getDowntime() { ... } 37 12 import org.aspectj.lang.annotation.BridgedSourceLocation; private void updateVmAfterMigratic)ifferentCluster() { ... } import org.ovirt.engine.core.common.AuditLogType; private Boolean getAutoConverge() { ... } private Boolean getMigrateCompressed() { ... } import org.ovirt.engine.core.bll.CommandActionState; \$14@logs for org.ovirt.engine.core.bll.storage.export.ExportVmTemplateCommand: private int getMaximumMigrationDowntime() { ... } private boolean isTunnelMigrationUsed() { ... } private String getMigrationNetworkIp() { ... } 150 case EXECUTE & success public privileged aspect Logs { log IMPORTEXPORT_STARTING_EXPORT_TEMPLATE, 16 «FOR command:resource.allContents.filter(typeof(Command)).toIterable» private String getMigrationNetworkAddress(Guid hostId, String migrationNetworkName) case EXECUTE & failure 170 «command.compile» **Implementing code transformation** tected boolean migrationInterfaceUp(VdsNetworkInterface nic, List< VdsNetworkInterface> nics) { ... } log IMPORTEXPORT_EXPORT_TEMPLATE_FAILED, «ENDEOR 18 198 case END SUCCESS & success using a Language Workbench public AuditLogType getAuditLogTypeValue() { ... } private AuditLogType getAuditLogForMigrationStarted() { ... } protected AuditLogType getAuditLogForMigrationFailure() { ... } log IMPORTEXPORT EXPORT TEMPLATE, 20 47 210 case END SUCCESS & failure protected Guid getDestinationVdsId() { ... } protected void setDestinationVdsId(Guid vdsId) { ...] log IMPORTEXPORT EXPORT TEMPLATE FAILED 22 def compile(Command command) ''' QOverride 23 ; «NodeModelUtils.getNode(command).toSourcePosition» protected boolean canDoAction() { ... } protected void setActionMessageParameters() { ... } AuditLogType around(«command.type.qualifiedName» command): execution(* getAuditLogTypeValue() Auditing «FOR acase:command.cases» 00verride 250 logs for org.ovirt.engine.core.bll.storage.disk.AddDiskCommand : public void rerun() { ... } «acase.compile» 268 case EXECUTE & success & internal & isDiskStorageTypeRequiresExecuteState **Generate IDE Support O**verride «ENDFOR» protected void reexecuteCommand() { ... } 27 log ADD DISK INTERNAL, return «IF command.^default != null»AuditLogType.«command.^default.simpleName»«ELSEIF comman protected void determineMigrationFailureForAuditLog() { ... } for programming with 28 case EXECUTE & success & Override tected Guid getCurrentVdsId() { ... } <u>8</u>29 ; public String getDuration() { ... } public String getTotalDuration() { ... } **Permissions** acquireLock the **DSAL** Annotations : EList<JvmAnnotationReference> - JvmAnnotationReference> - JvmAnnotationReferenc public String getActualDowntime() { acquireLockAsyncTask Fclass : Class<?> - Object.getClass() def compile(Case acase) ''' rotected String getLockMessage() { ... } rivate List<Guid> getVdsBlackList() { ... } rotected List<Guid> getVdsWhiteList() { ... } 609 if («acase.result.compile»«IF acase.actionState!=nu ocompile : CharSequence - AuditLogGenerator.com acquireLockInternal return AuditLogType.«acase.msg.simpleName»; ublic List<PermissionSubject> getPermissionCheckSubjects() { ... } constant : boolean - JvmField.isConstant() canRunActionOnNonManagedVm public void onPowerringUp() { ... } **DSAL with easy-to-use syntax Syntax Highlighting** def compile(Result clause) { 656 checkDiskUsedAsOvfStore switch clause { 66 ConstantValue : Object - JvmField.getConstantVal 67 case SUCCESS: 'command.getSucceeded()' for synchronization checkExceedingMaxBlockDiskSize 68 case FAILURE: '!command.getSucceeded()' ublic class AddDiskCommand <T extends AddDiskParameters> ... { ConstantValue = value : void - JvmField.setConsta rotected AddDiskCommand(Guid commandId) { ... } 69 ublic AddDiskCommand(T parameters) { ... } 70 ConstantValueAsBoolean : boolean - JvmField.get@ public AddDiskCommand(T parameters, Co Context commandContext) { ... } - -*ovirt.locks Jverride rotected boolean canDoAction() { ... } rotected boolean checkIfLunDiskCanBeAdded(DiskValidator diskValidator) { ... } ... DiskValidator diskValidator (...) ConstantValueAsByte : byte - JvmField.getConsta def compile(boolean internal) 728 "''«IF internal» && command.isInternalExecution()«EN protected boolean checkIfImageDiskCanBeAdded(VM vm, DiskValidator diskVa private boolean isShareableDiskOnGlusterDomain() { ... } 73 I locks for org.ovirt.engine.core.bll.MigrateVmCommand (async): **Generated aspect** Ctrl+Space to show shortest p exclusively (overrides) { colean canAddShareableDisk() { ... } def compile(JvmEnumerationLiteral actionState) 756 wivate boolean checkExceedingMaxBlockD for auditing group: VM instance: getVmId ''' && command.getActionState() == CommandActionState.«actionState.»''' private boolean isStoragePoolMatching(VM vm) { ... protected boolean checkImageConfiguration() { ... **3**76 **AJDT Markers** private double getRequestDiskSpace() { ... } def compile(JvmField field) message: ACTION_TYPE_FAILED_VM_IS_BEING_MIGRATED <"VmName", getVmName> **Override** '' && command.«field.simpleName»''' protected boolean isVmExist() { ... } private DiskImage getDiskImageInfo() { ... } private boolean isExceedMaxBlockDiskSize() { **Extend AspectJ to serve as** a kernel language for LOM rotected DiskLunMapDao getDiskLunMapDao() 🥇 ... 🖇 protected DiskImageDynamicDao getDiskImage 80 locks for org.ovirt.engine.core.bll.storage.disk.AddDiskCommand (sync): ao() { ...] public privileged aspect Logs { private Guid getDisksStorageDomainId() { ... } exclusively (overrides) { 99 **COverride** @BridgedSourceLocation(line=1, ublic Guid getStorageDomainId() { ... 10 group: VM_DISK_BOOT instance: getVmId if isBootableDisk file="/bll/src/main/java/org/ovirt/engine/core/bll/ovirt.audit", public List<PermissionSubject> getPermissionCheckSubjects() { ... } protected void setActionMessageParameters() { ... } inclusively (overrides) { module="ovirt.audit") Syntax-errors ©Override group: VM instance: getVmId AuditLogType around(org.ovirt.engine.core.bll.MigrateVmCommand command): protected void executeVmCommand() { ... } private void createDiskBasedOnLun() { ... } execution(* getAuditLogTypeValue()) && this(command) { d VmDevice addManagedDeviceFo isk(Guid diskId, Boolean isUsingScsiR if (command.getSucceeded() && command.isReturnValueUp()) VmDevice addManagedDeviceForDisk(Guid diskId) { ... } protected boolean shouldDiskBePlugged() { ... } private void createDiskBasedOnImage() { ... } return AuditLogType.VM_MIGRATION_DONE; locks for RunV 217 private void createDiskBasedOnCinder() { ... } if (command.getSucceeded() && command.isInternalExecution()) RuntimeValuePropertyInfo - com.sun.xml.internal.bind.v2.model.runtir

<pre>private VdcActionParametersBase buildAddCinderDiskParameters() { } private void setVmSnapshotIdForDisk(AddImageFromScratchParameters parameters) { } private void addDiskPermissions(Disk disk) { }</pre>	18
<pre>@Override public AuditLogType getAuditLogTypeValue() { } private boolean isDiskStorageTypeRequiresExecuteState() { } private AuditLogType getExecuteAuditLogTypeValue(boolean successful) { } protected AuditLogType getEndSuccessAuditLogTypeValue(boolean successful) { }</pre>	uto-Completion
<pre>@Override protected VdcActionType getChildActionType() { } @Override protected List<class<?>> getValidationGroups() { } @Override protected Map<string, pair<string,="" string="">> getSharedLocks() { } @Override protected Map<string, pair<string,="" string="">> getExclusiveLocks() { } @Override protected void setLoggingForCommand() { } private Guid getQuotaId() { } @Override protected void endSuccessfully() { } private void plugDiskToVmIfNeeded() { } @Override</string,></string,></class<?></pre>	
<pre>public List<quotaconsumptionparameter> getQuotaStorageConsumptionParameters() { } protected StorageDomainValidator createStorageDomainValidator() { } } public class ExportVmTemplateCommand<t extends="" moveorcopyparameters=""> extends { private String cachedTemplateIsBeingExportedMessage; public ExportVmTemplateCommand(T parameters) {} protected ExportVmTemplateCommand(Guid commanId) {} @Override protected LockProperties applyLockProperties(LockProperties lockProperties) {}</t></quotaconsumptionparameter></pre>	
<pre>@Override protected void moveOrCopyAllImageGroups(final Guid containerID, final Iterable<diskimage> disks) {} @Override protected Map<string, pair<string,="" string="">> getSharedLocks() {} @Override protected Map<string, pair<string,="" string="">> getExclusiveLocks() {} @Override protected void executeCommand() {} private String getTemplateIsBeingExportedMessage() {}</string,></string,></diskimage></pre>	IDE Langu
<pre>@Override protected boolean canDoAction() {} @Override public AuditLogType getAuditLogTypeValue() {} @Override protected void setActionMessageParameters() {} @Override protected void incrementDbGeneration() {}</pre>	IDE Lang
<pre>protected void endActionOnAllImageGroups() {} @Override public Map<string, string=""> getJobMessageProperties() {} }</string,></pre>	Aspect [



Advisor: David H. Lorenz This research was supp in part by the Israel Sc Foundation (ISF) under No. 1440/14.

	Language C	
	Aspect Dev. To	
z ported c <i>ience</i> grant	S T U D E RESEAR COMPETIT	

	LOP	LOM	Practical LOM	
Languages	DSLs	DSALs	DSALs reducible to a GPAL	
Compiler	Host compiler	Per-DSAL weaver	Kernel weaver	
IDE Tools for Language Dev.	Language Workbench	None for weaving semantics	Language Workbench	
IDE Tools for Language Use	Language Workbench	Manual	Language Workbench	
pect Dev. Tools	N/A	Per-DSAL	AJDT	

Q RuntimeValuePropertyInfoImpl - com.sun.xml.internal.bind.v2.model.it

G RuntimeVersion - com.sun.xml.internal.ws.util.RuntimeVersion

•	0 0
return	AuditLogType. <i>VM_MIGRATION_START_SYSTEM_INITIATED;</i>
if (comma	nd.getSucceeded())
return	AuditLogType. <i>VM_MIGRATION_START;</i>
•	and.getSucceeded() && command.isHostInPrepareForMaintenance()) AuditLogType. <i>VM_MIGRATION_FAILED_DURING_MOVE_TO_MAINTENANCE;</i>
return	and.getSucceeded()) AuditLogType. <i>VM_MIGRATION_FAILED;</i> ditLogType. <i>UNASSIGNED;</i>
}	
<pre>@BridgedSou</pre>	rceLocation(line=14,
C · · · · · · · ·	

file="/bll/src/main/java/org/ovirt/engine/core/bll/ovirt.audit", module="ovirt.audit") AuditLogType around(org.ovirt.engine.core.bll.storage.export.ExportVmTemplateCommand command): execution(* getAuditLogTypeValue()) && this(command) { if (command.getSucceeded() && command.getActionState() == CommandActionState.EXECUTE) return AuditLogType.IMPORTEXPORT STARTING EXPORT TEMPLATE; if (!command.getSucceeded() && command.getActionState() == CommandActionState.EXECUTE) return AuditLogType.IMPORTEXPORT_EXPORT_TEMPLATE_FAILED; if (command.getSucceeded() && command.getActionState() == CommandActionState.END_SUCCESS) return AuditLogType.IMPORTEXPORT_EXPORT_TEMPLATE; if (!command.getSucceeded()

&& command.getActionState() == CommandActionState.END_SUCCESS) return AuditLogType.IMPORTEXPORT EXPORT TEMPLATE FAILED;

return proceed(command);

@BridgedSourceLocation(line=25,

file="/bll/src/main/java/org/ovirt/engine/core/bll/ovirt.audit", module="ovirt.audit")

AuditLogType around(org.ovirt.engine.core.bll.storage.disk.AddDiskCommand command): execution(* getAuditLogTypeValue()) && this(command) {

... skipped ...