OpenDreamKit Work Package 6
Data/Knowledge/Software-Bases

Michael Kohlhase
FAU Erlangen-Nürnberg
http://kwarc.info/kohlhase

Second OpenDreamKit Review, Luxembourg, October 30. 2018
1 Introduction
Conclusion: What are we doing in WP6 in terms of a VRE

- SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.

- Overall pattern: design – prototype – scale
Conclusion: What are we doing in WP6 in terms of a VRE

- SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- SageMath/CoCalc is very successful, because integration is lightweight:
  - It makes no assumption on the meaning of math objects exchanged.
  - Restricts itself to master-slave integration of systems into SageMath.
- But there are safety, extensibility, and flexibility issues!

- MitM tries to take the high road (made possible by OpenDreamKit)
  - Safety: by semantic (i.e. context-aware) objects passed.
  - Extensibility: any open-API system (i.e. with API CDs) can play.
  - Flexibility: full peer-to-peer possibilities. (future: service discovery)
- But we have to develop a whole new framework!

- Review Period 2: State of WP6 (MitM) Integration
  - Developed mathematical use-cases (what do researchers want to do)
  - Extended middleware, grown MitM ontology, collected alignments
  - Jupyter integration into MathHub.info

- Plan for Review Period 3: Extend to external use-cases users (scale and deploy publicly)

- overall pattern: design – prototype – scale
Conclusion: What are we doing in WP6 in terms of a VRE

- SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
  - SageMath/CoCalc is very successful, because integration is lightweight:
    - It makes no assumption on the meaning of math objects exchanged.
    - Restricts itself to master-slave integration of systems into SageMath.
  - But there are safety, extensibility, and flexibility issues!

- MitM tries to take the high road (make possible by OpenDreamKit)
  - Safety: by semantic (i.e. context-aware) objects passed.
  - Extensibility: any open-API system (i.e. with API CDs) can play.
  - Flexibility: full peer-to-peer possibilities. (future: service discovery)

 But we have to develop a whole new framework!(Review 1 \(\leadsto\) Proof of Concept)

- overall pattern: design – prototype – scale
Conclusion: What are we doing in WP6 in terms of a VRE

- SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- SageMath/CoCalc is very successful, because integration is lightweight:
  - It makes no assumption on the meaning of math objects exchanged.
  - Restricts itself to master-slave integration of systems into SageMath.
- But there are safety, extensibility, and flexibility issues!
- MitM tries to take the high road (make possible by OpenDreamKit)
  - Safety: by semantic (i.e. context-aware) objects passed.
  - Extensibility: any open-API system (i.e. with API CDs) can play.
  - Flexibility: full peer-to-peer possibilities. (future: service discovery)
- But we have to develop a whole new framework!(Review 1 → Proof of Concept)

- Review Period 2: State of WP6 (MitM) Integration
  - Developed mathematical use-cases (what do researchers want to do)
  - Extended middleware, grown MitM ontology, collected alignments
  - Jupyter integration into MathHub.info

- overall pattern: design – prototype – scale
Conclusion: What are we doing in WP6 in terms of a VRE

- **SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.**
- **SageMath/CoCalc is very successful, because integration is lightweight:**
  - It makes no assumption on the meaning of math objects exchanged.
  - Restricts itself to master-slave integration of systems into SageMath.
- But there are safety, extensibility, and flexibility issues!
- **MitM tries to take the high road** (make possible by OpenDreamKit)
  - **Safety:** by semantic (i.e. context-aware) objects passed.
  - **Extensibility:** any open-API system (i.e. with API CDs) can play.
  - **Flexibility:** full peer-to-peer possibilities. (future: service discovery)
- But we have to develop a whole new framework!(Review 1 ~ Proof of Concept)

- **Review Period2: State of WP6 (MitM) Integration**
  - Developed mathematical use-cases (what do researchers want to do)
  - Extended middleware, grown MitM ontology, collected alignments
  - Jupyter integration into MathHub.info

- **Plan for Review Period 3: Extend to external use-cases users** (scale and deploy publicly)
- **overall pattern:** design – prototype – scale
Background: WP6 (Data/Knowledge/Software-Bases)

▶ From the Proposal:

Proposed Focus: Supply this data to VRE components in an integrated fashion programmatically

Kohlhase: ODK WP6 3 Second ODK Review, Oct. ’18
Background: WP6 (Data/Knowledge/Software-Bases)

From the Proposal:

Proposed Focus: Supply this data to VRE components in an integrated fashion programmatically
Results of the WP6 Workshops: Semantic Interoperability

The WP6 group had a series of workshops

- Kickoff in Paris (Sep '15): strategies for joint knowledge representation

- WS in St. Andrews (Feb '16): Math in the Middle Arch. for System Interop.
- WS in Bremen (June '16): GAP/SageMath API Content Dictionaries (CDs)
- WS in Berlin (Feb '17): Math-in-the-Middle Ontology

Kohlhase: ODK WP6 4 Second ODK Review, Oct. '18
Results of the WP6 Workshops: Semantic Interoperability

- The WP6 group had a series of workshops
  - Kickoff in Paris (Sep '15): strategies for joint knowledge representation
  - WS in St. Andrews (Feb '16): Math in the Middle Arch. for System Interop.
  - WS in Bremen (June '16): GAP/SageMath API Content Dictionaries (CDs)
  - WS in Berlin (Feb '17): Math-in-the-Middle Ontology
  - Kohlhase: ODK WP6 4 Second ODK Review, Oct. '18
Results of the WP6 Workshops: Semantic Interoperability

- The WP6 group had a series of workshops
  - Kickoff in Paris (Sep ’15): strategies for joint knowledge representation
  - WS in St. Andrews (Feb ’16): Math in the Middle Arch. for System Interop.

The WP6 group had a series of workshops

- Kickoff in Paris (Sep ’15): strategies for joint knowledge representation
- WS in St. Andrews (Feb ’16): Math in the Middle Arch. for System Interop.
- WS in Bremen (June ’16): GAP/SageMath API Content Dictionaries (CDs)
Results of the WP6 Workshops: Semantic Interoperability

- The WP6 group had a series of workshops
  - Kickoff in Paris (Sep ’15): strategies for joint knowledge representation
  - WS in St. Andrews (Feb ’16): Math in the Middle Arch. for System Interop.
  - WS in Bremen (June ’16): GAP/SageMath API Content Dictionaries (CDs)
  - WS in Berlin (Feb ’17): Math-in-the-Middle Ontology
2 Mathematical Use Cases
Running Example/Use Case: Jane’s Invariant Experiments

Jane wants to experiment with invariant theory of finite groups. 

- She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$. 
- **Goal**: construct an ideal $I$ in $R$ that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of $G$ to properties of $I$ and the quotient of $R$ by $I$. 
- **Idea**: pick some polynomial $p$ from $R$ and consider the ideal $I$ of $R$ that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$. 
- For effective further computation with $I$, she needs a Gröbner base of $I$. 

Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP’s orbit algorithm and Singular’s Gröbner base algorithm, which she knows to be very efficient. 

Problem: Jane has to learn the GAP and Singular languages and retype the results in them. (error-prone) 

For the sake of example, we will work with $n = 4$, $G = D_4$ (the dihedral group), and $p = 3 \cdot X_1 + 2 \cdot X_2$, but our results apply to arbitrary values. 

Caveat: $G$ is called “$D_4$” in SageMath but “$D_8$” in GAP due to differing conventions in different mathematical communities.
Jane wants to experiment with invariant theory of finite groups. She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$. The goal is to construct an ideal $I$ in $R$ that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of $G$ to properties of $I$ and the quotient of $R$ by $I$. The idea is to pick some polynomial $p$ from $R$ and consider the ideal $I$ of $R$ that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$. For effective further computation with $I$, she needs a Gröbner base of $I$. Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP’s orbit algorithm and Singular’s Gröbner base algorithm, which she knows to be very efficient.
Jane wants to experiment with invariant theory of finite groups. She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$. **Goal:** construct an ideal $I$ in $R$ that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of $G$ to properties of $I$ and the quotient of $R$ by $I$. **Idea:** pick some polynomial $p$ from $R$ and consider the ideal $I$ of $R$ that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$. For effective further computation with $I$, she needs a Gröbner base of $I$.

Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP’s orbit algorithm and Singular’s Gröbner base algorithm, which she knows to be very efficient.

**Problem:** Jane has to learn the GAP and Singular languages and retype the results in them. (error-prone)
Running Example/Use Case: Jane’s Invariant Experiments

Jane wants to experiment with invariant theory of finite groups.
- She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$.
- **Goal**: construct an ideal $I$ in $R$ that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of $G$ to properties of $I$ and the quotient of $R$ by $I$.
- **Idea**: pick some polynomial $p$ from $R$ and consider the ideal $I$ of $R$ that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$.
- For effective further computation with $I$, she needs a Gröbner base of $I$.
- Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP’s orbit algorithm and Singular’s Gröbner base algorithm, which she knows to be very efficient.

**Problem**: Jane has to learn the GAP and Singular languages and retype the results in them. (error-prone)
- For the sake of example, we will work with $n = 4$, $G = D_4$ (the dihedral group), and $p = 3 \cdot X_1 + 2 \cdot X_2$, but our results apply to arbitrary values.

**Caveat**: $G$ is called “$D_4$” in SageMath but “$D_8$” in GAP due to differing conventions in different mathematical communities.
John’s Use Case for LMFDB (slightly abridged)

- John wants to investigate the number fields which are generated by the coefficients of Hilbert modular forms (HMFs).
- LMFDB contains information about all HMFs over base fields $F$ of degree $N = 2, 3, 4, 5, 6$ (of parallel weight 2 and trivial character).
- Each HMF comes with a Hecke field $K$ which is stored via a defining polynomial (not canonical or minimal $\sim$ difficult to study).

**Example 2.1.** $K = \mathbb{Q}(\sqrt{2})$ may occur as $x^2 - 2$ and $x^2 - 2x - 1$.

- John would like to be able to
  1. extract these defining polynomials from the LMFDB,
  2. use them to define number fields in SageMath,
  3. find simpler polynomials defining the same fields, and
  4. study their arithmetic properties (e.g., their class numbers).
3 Realizing MitM Interoperability
– The Computational Group Theory Case Study –
Example 3.1. A theory of Groups

Declaration $\equiv$

name : type \[= \text{Def} \] [\# notation]

Axioms $\equiv$ Declaration with type $\vdash F$

ModelsOf makes a record type from a theory.

MitM Foundation: optimized for natural math formulation

- higher-order logic based on polymorphic $\lambda$-calculus
- judgements-as-types paradigm: $\vdash F \equiv$ type of proofs of $F$
- dependent types with predicate subtyping, e.g. \{n\}{‘a $\in$ mat(n, n)$|$symm(a)’}
- (dependent) record types for reflecting theories
MitM Computational Group Theory

Four levels of modeling

- **Abstract Level**: the group axioms, generating sets, homomorphisms, group actions, stabilisers, orbits, centralizers, normalizers.
- **Representation Level**: axiomatizations concrete objects suitable for computation — permutation groups, matrix groups, ..., also group actions, group homomorphism
- **Implementation Level**: permutation groups as subgroups of $S_{N+}$, concretely $S_{[1,\ldots,n]}$.
- **Concrete Level**: where actual computations happen.

Alignments between the MitM Ontology and the GAP API

<table>
<thead>
<tr>
<th>Level</th>
<th>MitM Ontology</th>
<th>GAP API</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract</td>
<td>Abstract GT</td>
<td>IsGroup</td>
</tr>
<tr>
<td>repn.</td>
<td>Permutation Groups, Matrix Groups, Finitely Presented Groups</td>
<td>IsPermGroup, IsMatrixGroup, IsFpGroup</td>
</tr>
<tr>
<td>impl.</td>
<td>$G \leq \text{Symmetric}([1..n])$</td>
<td>Group($(1,2,3)$)</td>
</tr>
<tr>
<td>concrete</td>
<td>Mathieu(11) $\leq \text{Symmetric}([1..11])$</td>
<td>. MathieuGroup(11)</td>
</tr>
</tbody>
</table>
The Knowledge Graph for MitM, SageMath, GAP, Singular
Definition 3.2. We call a pair of identifiers \((a_1, a_2)\) that describe the same mathematical concept an alignment. We call an alignment perfect, if it induces a total, truth-preserving translation.

(e.g. alignment up to argument order)

Intuition: Alignments don’t need to be perfect to be useful!

- Alignment up to Totality of Functions (e.g. division undefined on 0 and with \(\frac{x}{0} = 0\))
- Alignment for Certain Arguments (e.g. Addition on natural numbers and addition on real numbers)
- Alignment up to Associativity (e.g. binary addition and “sequential” addition)

They still allow for translating expressions between libraries. (under certain conditions)
In **SageMath** Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action $A$ of $G$ on $R$ that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$. 

**Jane’s Use case in the MitM System**
Jane’s Use case in the MitM System

In SageMath Jane has already built the ring \( R = \mathbb{Z}[X_1, X_2, X_3, X_4] \), the group \( G = D_4 \), the action \( A \) of \( G \) on \( R \) that permutes the variables, and
\[ p = 3 \cdot X_1 + 2 \cdot X_2. \]

She calls
\[
\begin{align*}
o &= \text{MitM.Gap.orbit}(G,A,p) \quad \# \text{the orbit} \\
i &= \text{MitM.Singular}(o).\text{Ideal}() \quad \# \text{the ideal} \\
g &= i.\text{Groebner}().\text{sage}() \quad \# \text{the Groebner basis}
\end{align*}
\]
Jane’s Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action $A$ of $G$ on $R$ that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.  
- She calls
  
  ```
  o = MitM.Gap.orbit(G,A,p)  # the orbit
  i = MitM.Singular(o).Ideal()  # the ideal
  g = i.Groebner().sage()  # the Groebner basis
  ```

- The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.
**Jane’s Use case in the MitM System**

- In SageMath Jane has already built the ring \( R = \mathbb{Z}[X_1, X_2, X_3, X_4] \), the group \( G = D_4 \), the action \( A \) of \( G \) on \( R \) that permutes the variables, and \( p = 3 \cdot X_1 + 2 \cdot X_2 \).

- She calls
  
  ```
  o = MitM.Gap.orbit(G,A,p) # the orbit
  i = MitM.Singular(o).Ideal() # the ideal
  g = i.Groebner().sage() # the Groebner basis
  ```

- The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.

- GAP returns the orbit: \( O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2] \).
Jane’s Use case in the MitM System

In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action $A$ of $G$ on $R$ that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.

She calls

\begin{align*}
o &= \text{MitM.Gap.orbit}(G,A,p) \ # \text{the orbit} \\
i &= \text{MitM.Singular}(o).\text{Ideal()} \ # \text{the ideal} \\
g &= i.\text{Groebner()}.\text{sage()} \ # \text{the Groebner basis}
\end{align*}

The MitM server translates MitM.Gap.orbit(G,A,p) to the GAP system dialect and sends it to GAP.

GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$

The MitM server translates MitM.Singular(O).Ideal().Groebner() to the Singular system dialect and sends it to Singular.
Jane’s Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action $A$ of $G$ on $R$ that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.

- She calls
  
  $o = \text{MitM.Gap.orbit}(G, A, p) \ # \text{the orbit}$
  $i = \text{MitM.Singular}(o).\text{Ideal}() \ # \text{the ideal}$
  $g = i.\text{Groebner}().\text{sage}() \ # \text{the Groebner basis}$

- The MitM server translates $\text{MitM.Gap.orbit}(G, A, p)$ to the GAP system dialect and sends it to GAP.

- GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$

- The MitM server translates $\text{MitM.Singular}(O).\text{Ideal}().\text{Groebner}()$ to the Singular system dialect and sends it to Singular.

- Singular returns the Gröbner base $B$. 
Jane’s Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action $A$ of $G$ on $R$ that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls
  
  ```
  o = MitM.Gap.orbit(G,A,p)  # the orbit
  i = MitM.Singular(o).Ideal()  # the ideal
  g = i.Groebner().sage()  # the Groebner basis
  ```

- The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.
- GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$.
- The MitM server translates `MitM.Singular(O).Ideal().Groebner()` to the Singular system dialect and sends it to Singular.
- Singular returns the Gröbner base $B$.
- The MitM server translates $B$ to the SageMath system dialect and sends it to SageMath, where the result is shown to Jane.

  $$B = [X_1 - X_4, X_2 - X_4, X_3 - X_4, 5 \cdot X_4].$$
Distributed Computational Group Theory

- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

  ![Diagram showing the integration of Sage, MMT Mediator, GAP, and Singular]

- Nucleus of the OpenDreamKit interoperability layer. Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

1: $\text{orbit}(G, A, p)@Sa$

Nucleus of the OpenDreamKit interoperability layer. Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

1: \( \text{orbit}(G, A, p)@Sa \)

2: \( \text{orbit}(G, A, p)@G \)

Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Distributed Computational Group Theory

- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

1: $\text{orbit}(G, A, p)@S_a$

2: $\text{orbit}(G, A, p)@G$

3: $O@G$

- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Distributed Computational Group Theory

- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

1: $\text{orbit}(G, A, p)@S_a$

Sage  \hspace{5cm} MMT Mediator  \hspace{5cm} GAP

2: $\text{orbit}(G, A, p)@G$

3: $O@G$

$\text{O.i.p}@Si$

- Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

1: \( \text{orbit}(G, A, p)@Sa \)
2: \( \text{orbit}(G, A, p)@G \)
3: \( O@G \)
4: \( O.i.p@Si \)
5: \( B@Si \)

Nucleus of the OpenDreamKit interoperability layer. Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Distributed Computational Group Theory

- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

1: \(\text{orbit}(G, A, p)@Sa\)
2: \(\text{orbit}(G, A, p)@G\)
3: \(O@G\)
4: \(O.i.p@Si\)
5: \(B@Si\)
6: \(B@Sa\)

- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments.
Future Use Case (Steve is Jane’s Colleague)

- Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial \( p = x^5 - 2 \).
- He discovers the GAP package radiroot (does not work for \( p \)).
Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial \( p = x^5 - 2 \).

He discovers the GAP package radiroot (does not work for \( p \)).

Jane suggests PARI/GP: he calls (once that is MitM-connected)
\[
G := \text{MitM}("\text{PARIGP}", "\text{GaloisGroup}", p)
\]
from PARI/GP which gives him the desired Galois group as a GAP permutation group.
Future Use Case (Steve is Jane’s Colleague)

▶ Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
▶ He discovers the GAP package radiroot (does not work for $p$)
▶ Jane suggests PARI/GP: he calls (once that is MitM-connected) $G := \text{MitM}("\text{PARIGP},"\text{GaloisGroup},p)$ from PARI/GP which gives him the desired Galois group as a GAP permutation group.
▶ Steve repeats Jane’s experiments on $G$, without leaving GAP.
Future Use Case (Steve is Jane’s Colleague)

- Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial \( p = x^5 - 2 \).
- He discovers the GAP package radiroot (does not work for \( p \)).
- Jane suggests PARI/GP: he calls (once that is MitM-connected) \( G := \text{MitM("PARIGP","GaloisGroup",p)} \) from PARI/GP which gives him the desired Galois group as a GAP permutation group.
- Steve repeats Jane’s experiments on \( G \), without leaving GAP.
- Finally, Steve installs a GAP method by calling

\[
\text{InstallMethod(GaloisGroup, "for a polynomial", [IsUnivariatePolynomial],}
\quad \text{p -> MitM("PARIGP", "GaloisGroup", p))}
\]

\( \sim \) extends \text{GaloisGroup} to rational polynomials in GAP.

- This replaces a significant part of the 1800-LoC radiroot package (by PARI/GP delegation)
MitM-based Integration centers around the MitM Ontology

If you are Really interested in the Graphs

interact with them in 3D
4 MitM InterOperability for Mathematical Databases
Mathematical Knowledge Bases (MKS)

State of the Art: mathematical object databases (GAP libraries, OEIS, LMFDB)

Introduction and more
- Introduction
- Features
- Universe
- Future Plans
- News

L-functions
- Degree: 1 2 3 4
- Zeros

Modular Forms
- Classical
- Maass
- Hilbert
- Bianchi

GL(2)
- Mass

GL(3)
- Slagel

Other
- Varieties
- Elliptic
- NumberFields
- Genus 2:

Curves
- /Q

Elliptic curves in class 11.a
- LMDB label
- Cremona label
- Weierstrass coefficients
- Torsion order
- Modular degree
- O

<table>
<thead>
<tr>
<th>Class</th>
<th>Label</th>
<th>Weierstrass Coefficients</th>
<th>Order</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.a1</td>
<td>11a1</td>
<td>[0, -1, 1, -7820, -263580]</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>11.a2</td>
<td>11a2</td>
<td>[0, -1, 1, -10, -20]</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>11.a3</td>
<td>11a3</td>
<td>[0, -1, 1, 0, 0]</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Rank
The elliptic curves in class 11.a have rank 0.

Modular form 11.2.1.a

q - 2q^2 - q^3 + 2q^4 + q^5 + 2q^6 - 2q^7 - 2q^8 + q^11 - 2q^{12} + 4q^{14} - q^{15} - 4q^{16} - 2q^{17} + ...

Isogeny matrix

\[
\begin{pmatrix}
1 & 5 & 23 \\
5 & 1 & 5 \\
23 & 5 & 1
\end{pmatrix}
\]

Isogeny graph

The ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES®

founded in 1964 by N. J. A. Sloane

Annual appeal: Please make a donation to keep the OEIS running! Over 6000 articles have referenced us, often saying "we discovered this result with the help of the OEIS".

(Greetings from The On-Line Encyclopedia of Integer Sequences®)

A000045 Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1.

<table>
<thead>
<tr>
<th>OEIS</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A000045</td>
<td>Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1.</td>
</tr>
</tbody>
</table>

OEIS: A000045

- Also sometimes called Lamé’s sequence.
- \( F(n+2) \) = number of binary sequences of length \( n \) that have no consecutive 0’s.
- \( F(n+2) \) = number of subsets of \( \{1, 2, \ldots, n\} \) that contain no consecutive integers.
- \( F(n+1) \) = number of tilings of a \( 2 \times n \) rectangle by \( 2 \times 1 \) dominoes.
- \( F(n+1) \) = number of matchings (i.e., Hosoya index) in a path graph on \( n \) vertices: \( F(n+1) \) = because the matchings of the path graph on the vertices A, B, C, D are the empty set, \{A\}, \{B\}, \{C\}, \{D\}, and \{A, B, C, D\}.

Kohlhase: ODK WP6 15 Second ODK Review, Oct. '18
Mathematical Knowledge Bases (MKS)

- **State of the Art**: mathematical object databases (GAP libraries, OEIS, LMFDB)
- **Problem**: human-oriented interface, very limited programmatic API, no computation

- **Idea**: can’t we use MitM Technologies here to integrate?
John’s Use Case for LMFDB (slightly abridged)

▶ John wants to investigate the number fields which are generated by the coefficients of Hilbert modular forms (HMFs).
▶ LMFDB contains information about all HMFs over base fields \( F \) of degree \( \mathcal{N} = 2, 3, 4, 5, 6 \) (of parallel weight 2 and trivial character).
▶ Each HMF comes with a Hecke field \( K \) which is stored via a defining polynomial (not canonical or minimal \( \sim \) difficult to study)

▶ Example 4.1. \( K = \mathbb{Q}(\sqrt{2}) \) may occur as \( x^2 - 2 \) and \( x^2 - 2x - 1 \).
▶ John would like to be able to
  1. extract these defining polynomials from the LMFDB,
  2. use them to define number fields in SageMath,
  3. find simpler polynomials defining the same fields, and
  4. study their arithmetic properties (e.g., their class numbers).
MitM-based Integration of Math Knowledge Bases

Requirements:
- a uniform programatic API to multiple MKB
- interacting with MKB at the “mathematics Level”.

Idea: use the Math-in-the-Middle Paradigm
- OMDoc/MMT-based API theories for the mathematical interface (MKB records as OM objects)
- alignments into MitM Ontology (for OM-dialect mediation)
- extend MMT’s built-in query language QMT to general Math query language

Problems:
- MKB tables become OMDoc/MMT theories (size problems)
- how to reconcile MKB records with OMDoc/MMT terms. (encoding/decoding)
- how to translate math-level queries to physical database queries
Example 4.2 (A transitive group represented in in LMFDB).

```json
{
  "ab": 1,
  "arith_equiv": 0,
  "auts": 1,
  "cyc": 1,
  "label": "1T1",
  "n": 1,
  ...
}
```

Legend: for understanding them

- the **cyc** field represents **being cyclic**
- the **n** field represents **degree**
- ...  

**Two Problems**: that have to be solved for MitM integration

- data base schema is not at the mathematical level (let alone interoperable)
- values are encoded for MongoDB convenience (what do they mean?)

(LMFDB improved documentation)
Definition 4.3 (Codec). A codec consists of two functions that translate between semantic types and realized types.

<table>
<thead>
<tr>
<th>Codecs</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>codec</td>
<td>type → type</td>
<td></td>
</tr>
<tr>
<td>StandardPos</td>
<td>codec (\mathbb{Z}^+)</td>
<td>JSON number if small enough, else JSON string of decimal expansion</td>
</tr>
<tr>
<td>StandardNat</td>
<td>codec (\mathbb{N})</td>
<td></td>
</tr>
<tr>
<td>StandardInt</td>
<td>codec (\mathbb{Z})</td>
<td></td>
</tr>
<tr>
<td>IntAsArray</td>
<td>codec (\mathbb{Z})</td>
<td>JSON List of Numbers</td>
</tr>
<tr>
<td>IntAsString</td>
<td>codec (\mathbb{Z})</td>
<td>JSON String of decimal expansion</td>
</tr>
<tr>
<td>StandardBool</td>
<td>codec (\mathbb{B})</td>
<td>JSON Booleans</td>
</tr>
<tr>
<td>BoolAsInt</td>
<td>codec (\mathbb{B})</td>
<td>JSON Numbers 0 or 1</td>
</tr>
<tr>
<td>StandardString</td>
<td>codec (\mathbb{S})</td>
<td>JSON Strings</td>
</tr>
</tbody>
</table>

StandardInt decodes 1 into the float 1, but \(2^{54}\) into the string "18014398509481984"
Matrix in the isogeny_matrix field

\[
\begin{bmatrix}
1 & 5 & 25 \\
5 & 1 & 5 \\
25 & 5 & 1
\end{bmatrix}
\]

represented as \([\{1,5,25\}, \{5,1,5\}, \{25,5,1\}]\)
Definition 4.4 (Codec Operator). A codec operator is a function which takes a codec, a set of parameters, and returns a codec.

<table>
<thead>
<tr>
<th>Codecs (continued)</th>
<th>_codec T → _codec List(T)</th>
<th>JSON list, recursively coding each element of the list</th>
</tr>
</thead>
<tbody>
<tr>
<td>StandardList</td>
<td>codec T → codec List(T)</td>
<td>JSON list, recursively coding each element of the list</td>
</tr>
<tr>
<td>StandardVector</td>
<td>codec T → codec Vector(n, T)</td>
<td>JSON list of fixed length n</td>
</tr>
<tr>
<td>StandardMatrix</td>
<td>codec T → codec Matrix(n, m, T)</td>
<td>JSON list of n lists of length m</td>
</tr>
</tbody>
</table>

StandardMatrix(StandardInt, 3, 3) generates the codec we used for the isogeny matrix
Our approach: Virtual Theories

Numbers

\[
\begin{align*}
Z^+ &: \text{type} \\
Z &: \text{type} \\
Z^+ \subset Z
\end{align*}
\]

Matrices

\[
\text{matrix} : \text{type} \rightarrow Z^+ \rightarrow Z^+ \rightarrow \text{type}
\]

Codecs

\[
\begin{align*}
\text{codec} &: \text{type} \rightarrow \text{type} \\
\text{standardInt} &: \text{codec} Z \\
\text{standardMatrix} &: \{T, n, m\} \text{codec} T \rightarrow \text{codec} \text{matrix}(n, m, T)
\end{align*}
\]

Elliptic Curve

\[
\begin{align*}
\text{ec} &: \text{type} \\
\text{from_record} &: \text{record} \rightarrow \text{ec} \\
\text{curveDegree} &: \text{ec} \rightarrow Z \\
\text{isogenyMatrix} &: \text{ec} \rightarrow \text{matrix}(3, 3, Z)
\end{align*}
\]

Elliptic Curve Schema Theory

\[
\begin{align*}
degree &: ?\text{implements} \text{curveDegree} \\
?\text{codec} &: \text{StandardInt} \\
\text{isogeny_matrix} &: ?\text{implements} \text{isogenyMatrix} \\
?\text{codec} &: \text{StandardMatrix}(3, 3, \text{StandardInt})
\end{align*}
\]

Elliptic Curve Database Theory

\[
\begin{align*}
11a1 &: \text{ec} = \ldots \\
11a2 &: \text{ec} = \ldots \\
\ldots
\end{align*}
\]

Imfdb Elliptic Curves

lazily loads from

\[\text{describes}\]

implements

Kohlhase: ODK WP6 22 Second ODK Review, Oct. '18
Example 4.5. Finding all cyclic transitive groups in LMFDB (recall from above)

\[
\begin{align*}
\text{x in (related to ( literal 'lmfdb:db/transitivegroups?group ) by (object declares))} \\
\text{| holds x (x cyclic x **=** true)}
\end{align*}
\]

This example does not rely on the internal structure of LMFDB

- can be translated into an LMFDB query using the just-defined codecs theory
- [http://www.lmfdb.org/api/transitivegroups/groups/?cyc=1](http://www.lmfdb.org/api/transitivegroups/groups/?cyc=1)
Solving John’s Hecke Fields Use Case

- **Remember**: John wanted to study number fields of HMFs via their Hecke field polynomials.

- John computes in SageMath and accesses LMFDB programmatically at the mathematical level (directly in the MitM dialect).

- Build a query for LMFDB

  ```python
  lmfdb = MitM.lmfdb
  algebra = MitM.smglom.algebra
  # a MitM expression that returns all hmf_forms with degree 2
  hmfs_query = lmfdb.hmf_forms.where(algebra.base_field_degree(2))
  # a MitM expression that additionally extracts the Hecke polynomial
  # from each hmf_form
  polys_query = hmfs_query.map(lambda x: lmfdb.hecke(x))
  # run the query via MitM and obtain the set of Sage polynomials
  polys = MitM.run(polys_query)
  # further processing in Sage
  fields = [NumberField(p) for p in polys]
  ```
Example 4.6. John’s use case in a Jupyter Notebook (with a SageMath kernel)

```
In [1]: # import all the relevant bits from MitM
    import MitM
    from MitM import lmfdb, algebra

In [6]: # put the query together
    query = lmfdb.hmf_hecke.where(
        algebra.HilbertNewforms.base_field_degree(int(2)),
        algebra.HilbertNewforms.dimension(int(2)),
    ).limit(until=int(10)).map(algebra.HeckeEigenvalues.heckePolynomial)

In [7]: # and run it
    MitM.run(query)

Out[7]: [x^2 + x + 7,
    x^2 + x + 4,
    x^2 + x + 7,
    x^2 + x + 2,
    x^2 + x + 4,
    x^2 + 12,
    x^2 + x + 1,
    x^2 + x + 4,
    x^2 + x + 2,
    x^2 + x + 1]```
...and the same in a Jupyter Notebook

Example 4.6. John’s use case in a Jupyter Notebook (with a SageMath kernel)

```python
In [1]: # import all the relevant bits from MitM
import MitM
from MitM import lmfdb, algebra

In [6]: # put the query together
query = lmfdb.hmf_hecke.where(
    algebra.HilbertNewforms.base_field_degree(int(2)),
    algebra.HilbertNewforms.dimension(int(2)),
).limit(10).map(algebra.HeckeEigenvalues.heckePolynomial)

In [7]: # and run it
MitM.run(query)

Out[7]:

[x^2 + x + 7,
 x^2 + x + 4,
 x^2 + x + 7,
 x^2 + x + 2,
 x^2 + x + 4,
 x^2 + 12,
 x^2 + x + 1,
 x^2 + x + 4,
 x^2 + x + 2,
 x^2 + x + 1]

In [ ]:
```

Upshot: We have a programmatic, math-level API for LMFDB

- embed into any MitM-connected system (syntax adapted to host system)
- no DB-level JSON encodings, but concepts like HilbertNewForms.dimension.
5  Jupyter Integration into MathHub
MathHub: A Portal and Archive of Flexiformal Maths

- **Idea**: learn from the open source community, offer a code repository with management support that acts as a hub for publication/development projects.
- **MathHub**: a collaborative development/hosting/publishing system of open-source, formal/informal math. (See http://mathhub.info)
MathHub: A Portal and Archive of Flexiformal Maths

- **Idea:** learn from the open source community, offer a code repository with management support that acts as a hub for publication/development projects.

- **MathHub:** a collaborative development/hosting/publishing system of open-source, formal/informal math.  
  *(See http://mathhub.info)*

- **MathHub Architecture:** Three core components  
  - **Representation:** OMDoc/MMT mechanized by the MMT system.
  - **Repositories:** GitLab  
    (git-based public/private repositories)  
    (all content served by MMT)
  - **Front-End:** React.JS

![Diagram of MathHub architecture](image)
An OpenDreamKit Risk come True

- **Drupal Apocalypse:**
  - The MathHub front-end was based onDrupal (very popular, mature)
  - our Drupal server was repeatedly hacked and compromised $\sim$ large maintenance overhead

- **Decision in April 2018:** Completely re-develop MathHub front-end using a web framework only.
  This was planned anyway (Drupal too heavyweight), but cost us months developer time until now.

- The new architecture (Docker compose + JSON + React.JS) helped integrate with Jupyter.
KPIs and Deliverables for WP6

- MitM-connected Systems: four (GAP, Sage, LMFDB, Singular) (See D6.5)
- Formal MitM Ontology: 55 files, 2600 LoF, 360 commits (See D6.8)
- Informal MitM Ontology: 815 theories, 1700 concepts in English, German, (Romanian, Chinese)
- MitM System API Theories (GAP, Sage, LMFDB, Singular): 1.000+ Theories, 22,000 Concepts.
- Multi-Site involvement of Researchers (Mobility of Researchers)
  - PD. Dr. Florian Rabe (Joint appointment UPSud/FAU)
  - Felix Schmoll Summer Internship (From JacU to St.Andrews)
  - Prof. Nathan Carter (Bentley Univ.) in St. Andrews (Sabbatical)
- Heavy interest by the theorem proving community about MitM Ontology
- Logipedia (http://logipedia.science) adopts the MitM principle of integrating (logical) systems by aligning concepts.
- First ODK-external MitM “user” for the next months: Andrea Thevis, Saarbrücken
Conclusion: What are we doing in WP6 in terms of a VRE

- SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- SageMath/CoCalc is very successful, because integration is lightweight:
  - It makes no assumption on the meaning of math objects exchanged.
  - Restricts itself to master-slave integration of systems into SageMath.
  - But there are safety, extensibility, and flexibility issues!
- MitM tries to take the high road (make possible by OpenDreamKit)
  - Safety: by semantic (i.e. context-aware) objects passed.
  - Extensibility: any open-API system (i.e. with API CDs) can play.
  - Flexibility: full peer-to-peer possibilities. (future: service discovery)
  - But we have to develop a whole new framework!(Review 1 ~ Proof of Concept)

- Review Period 2: State of WP6 (MitM) Integration
  - Developed mathematical use-cases (what do researchers want to do)
  - Extended middleware, grown MitM ontology, collected alignments
  - Jupyter integration into MathHub.info

- Plan for Review Period 3: Extend to external use-cases users (scale and deploy publicly)
- overall pattern: design – prototype – scale